



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Plataforma web de
trabajo colaborativo**



Presentado por Álvaro Díez Sáenz en
Universidad de Burgos — 8 de julio de 2024
Tutor: Rubén Ruiz González

Índice general

Índice general	3
Índice de figuras	5
Índice de tablas	7
Plan de Proyecto Software	9
A.1. Introducción	9
A.2. Planificación temporal	10
A.3. Estudio de viabilidad	12
Especificación de Requisitos	17
B.1. Introducción	17
B.2. Objetivos generales	18
B.3. Catálogo de requisitos	19
B.3. Especificación de requisitos	20
Especificación de diseño	29
C.1. Introducción	29
C.2. Diseño de datos	29
C.3. Diseño procedimental	33
C.4. Diseño arquitectónico	39
Documentación técnica de programación	41
D.1. Introducción	41
D.2. Estructura de directorios	41
D.3. Manual del programador	43
D.4. Compilación, instalación y ejecución del proyecto	51
D.5. Pruebas del sistema	53
Documentación de usuario	55
E.1. Introducción	55
E.2. Requisitos de usuarios	55
E.3. Instalación	56
E.4. Manual del usuario	56
Identificación de usuario	56
F.1. Introducción	71

F.2.	Implicación ética	71
F.3.	Implicación sociocultural	73
F.4.	Implicación medioambiental	73
F.5.	Conclusión	74
Bibliografía		77

Índice de figuras

Ilustración A.1 Diagrama de Gantt	10
Ilustración B.1 Casos de uso	20
Ilustración C.1 Diagrama Entidad Relación	32
Ilustración C.2 Diagrama secuencia registro usuario	33
Ilustración C.3 Diagrama secuencia inicio de sesión	34
Ilustración C.4 Diagrama secuencial subida de archivos	35
Ilustración C.5 Diagrama secuencial eliminado archivos	35
Ilustración C.6 Diagrama secuencial creación tarea	36
Ilustración C.7 Diagrama secuencial eliminación tareas	36
Ilustración C.8 Diagrama secuencia crear grupo	37
Ilustración C.9 Diagrama secuencia invitar usuario	37
Ilustración C.10 Diagrama secuencia compartir archivo	38
Ilustración C.11 Modelo-Vista-Controlador [8]	39
Ilustración D.1 Pagina descarga Python [9]	43
Ilustración D.2 Path del sistema	44
Ilustración D.3 Configuración path del sistema	44
Ilustración D.4 Variables del sistema	45
Ilustración D.5 Descarga Visual Studio Code	46
Ilustración D.6 Ventana Visual Studio Code	46
Ilustración D.7 Descarga Git [10]	47
Ilustración D.8 Instalación Git	47
Ilustración D.9 git --version	48
Ilustración D.10 Configuración git	48
Ilustración D.11 Ejemplo ruta app.py	49
Ilustración D.12 Ejemplo modelo en models.py	50
Ilustración D.13 Ejemplo HTML /templates	50
Ilustración D.14 Repositorio git del proyecto	51
Ilustración D.15 Ejecución Flask	52
Ilustración E.1 Login usuarios	56
Ilustración E.2 Registro usuarios	57
Ilustración E.3 Home aplicación	58

Ilustración E.4 Vista archivos usuario	59
Ilustración E.5 Sección subir archivos	60
Ilustración E.6 Sección archivo	60
Ilustración E.7 Sección compartir archivo	61
Ilustración E.8 Vista tareas usuario	61
Ilustración E.9 Sección crear tarea	62
Ilustración E.10 Sección tarea usuario	62
Ilustración E.11 Sección modificar tarea	63
Ilustración E.12 Sección compartir tarea	63
Ilustración E.13 Vista grupos usuario	64
Ilustración E.14 Sección crear grupo	64
Ilustración E.15 Vista principal grupos usuario	65
Ilustración E.16 Vista de un grupo	66
Ilustración E.17 Vista administrador de un grupo	67
Ilustración E.18 Sección editar grupo	67
Ilustración E.19 Administración grupo	68
Ilustración E.20 Sección invitar usuario	68
Ilustración E.21 Sección eliminar usuario	68
Ilustración E.22 Mensaje error	69
Ilustración E.23 Mensaje de éxito	69
Ilustración E.24 Mensaje advertencia	69

Índice de tablas

Tabla A 1 Listado de materiales y amortización	12
Tabla A.2 Amortización de materiales	12
Tabla A.3 Gastos variables y alquiler	13
Tabla A.4 Coste del trabajador para la empresa	14
Tabla A.5 Licencias	15
Tabla B.1 CU-1 Registrar usuario	21
Tabla B.2 CU-2 Autenticas usuarios	22
Tabla B.3 CU-3 Subir archivos	23
Tabla B.4 CU-4 Crear tarea	24
Tabla B.5 CU-5 Crear grupos	25
Tabla B.6 CU-6 Compartir archivos	26
Tabla B.7 CU-7 Descargar archivos	27

Apéndice A

Plan de Proyecto Software

A.1.Introducción

En el siguiente apartado se va a pensar tanto la planificación temporal como un estudio de viabilidad.

En la planificación temporal se detalla el ciclo de vida del proyecto con un enfoque iterativo e incremental, donde se describirán las fases y tareas llevadas a cabo durante el proyecto, así como un diagrama de Gantt para ilustrar las distintas etapas desde el inicio hasta el final.

Por otra parte, un estudio de viabilidad, donde se abordará tanto la viabilidad económica como la viabilidad legal. En la parte económica se van a analizar los costes generales del proyecto, dando una visión global de los recursos invertidos. En la viabilidad legal veremos las distintas licencias de software utilizado y como sus requisitos legales en cuanto a distribución y uso.

A.2. Planificación temporal

En la siguiente sección se va a mostrar la metodología utilizada para el desarrollo de este trabajo, se va a ver tanto el ciclo de vida utilizado como un diagrama de Gantt sobre todo el desarrollo del proyecto

Ciclo de vida iterativo e incremental: Este es un enfoque en el todo el proceso se divide en distintas interacciones, en estas se realizan repeticiones de ciclo de desarrollo es decir distintos ciclos cortos donde se revisa el trabajo y también se agrega funcionalidad de forma gradual [1]. Por esto este ha sido el ciclo de vida utilizado en el proyecto ya que desde un principio iba a ser complicado definir unas “metes” debido a los tiempos disponibles de desarrollo. Con este enfoque se consigue mayor flexibilidad y adaptación que sea necesaria al ser un proyecto de una única persona.

Se han planificado y llevado a cabo las diversas tareas cada una con su ciclo corto de desarrollo interno y sus objetivos específicos. A continuación, se muestra el diagrama de Gantt.

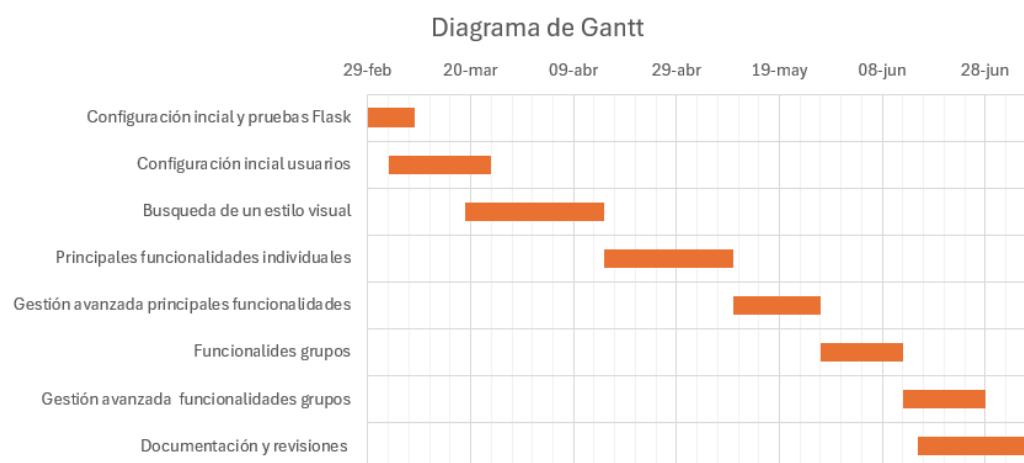


Ilustración 0.1 Diagrama de Gantt

Configuración inicial y pruebas con Flask: Esta etapa se realizó la configuración inicial del entorno de desarrollo utilizando Flask. Se realizaron pruebas para asegurar que la base del proyecto estaba correctamente configurada

Configuración inicial de usuarios: Se implementa el primero sistema de gestión de usuarios, permitiendo el registro, inicio de sesión y autenticación de los usuarios. También se implementan las primeras medidas básicas de seguridad.

Búsqueda de un estilo visual: En esta fase se pensaron diversas opciones de diseño para la interfaz de usuario. Se termino con un estilo visual coherente y sencillo.

Principales funcionalidades usuario: Se implementan las funcionalidades principales de los usuarios de la aplicación, incluyendo la gestión de archivos y tareas. Esta fase se centra en las características esenciales que permiten a los usuarios interactuar con la plataforma.

Gestión avanzada principales funcionalidades: Una vez correctas las funcionalidades básicas, se mejoró y optimizo estas características, tanto mejoran su estructura y de código y agregando más funcionalidad más complejas y seguras como también mejorando su aspecto visual.

Funcionalidades de grupos: En esta fase se centra en desarrollar las funcionalidades específicas para la gestión de grupos. Aquí incluimos la capacidad de crear y gestionar grupos de usuarios además de sus interfaces, pruebas etc...

Gestión avanzada de funcionalidades de grupos: Fase muy similar a la gestión avanzada de las principales funcionalidades, esta fase se centró en mejorar y expandir las capacidades relacionadas con los grupos. Se añadieron características adicionales y sobre todo se realizan mejoras en todos los aspectos relacionados.

Documentación y revisiones: La última fase del proyecto se dedicó a la creación de la documentación necesaria para el uso y mantenimiento de la aplicación. También se realizaron revisiones finales con un punto de vista más global del proyecto, para asegurar que todos los componentes del proyecto estuvieran funcionando correctamente y cumplieran con los requisitos establecidos, además de solucionar errores finales y modificaciones que se aprecian con una visión más global.

A.3. Estudio de viabilidad

Viabilidad económica

Este apartado se destina para visualizar una revisión de los beneficios potenciales en comparación con los costes que implica un desarrollo de estas características, todo esto se verá desde una perspectiva empresarial y se valoraran tanto los costes personales como los costes del material utilizado.

En primer lugar, van a recopilarse los costes materiales, en estos se evalúan todos los elementos que son necesarios para el desarrollo continuo del proyecto, en estos se excluyen todos los softwares libres, ya que estos permiten su uso de una manera gratuita.

También se recopilan los gastos variables como son los servicios utilizados por la persona para el desarrollo de la vida diaria y sus funciones y función del consumo que se realiza, dentro de estos están los gastos que pueden modificarse en función de su uso, como su agua, luz, internet, entre otros.... Así como también el costo de un estudio durante los 5 meses de proyecto.

Material	Valor (€)	Amortización
Ordenador y periféricos	1200	5 años
Licencia Windows 11	129	5 años

Tabla A A.3-1 Listado de materiales y amortización [2]

Amortización de Ordenador y periféricos

Costo anual = $1200\text{€}/5 \text{ años} = 240\text{€}/\text{año}$

Costo para 5 meses = $240\text{€}/\text{año} \times 5\text{meses}/12\text{meses} = 100\text{€}$

Amortización de licencia Windows 11

Costo anual = $129\text{€}/5 \text{ años} = 28,80\text{€}/\text{año}$

Costo para 5 meses = $28,80\text{€}/\text{año} \times 5\text{meses}/12\text{meses} = 10,75\text{€}$

Material	Gasto amortizado
Ordenador y periféricos	100€
Licencia Windows 11	10,75€

Tabla A.A.3-2 Amortización de materiales

Adicionalmente, también se incluyen los gastos indirectos como son, luz agua, internet, etc.... además del coste de alquiler de un estudio a 400€/mes durante los 5 meses del proyecto.

Concepto	Gasto mensual	Gasto total 5 meses
Gastos variables	40 €0	200 €
Alquiler estudio	400 €	2000 €

Tabla A.A.3-3 Gastos variables y alquiler

Para continuar se debe tener en cuenta el coste del trabajador, este proyecto se considera que ha sido llevado a cabo únicamente por un solo ingeniero informático, para esto vamos a establecer un salario bruto aproximado de 25.000€ ya que contamos que estamos antes un ingeniero informático Junior. También vamos a tener en cuenta que durante estos 5 meses de trabajo, se han invertido una media de 300 horas, distribuida a lo largo de las semanas de cada mes.

Teniendo en cuenta ese salario bruto de 25.000€ al año debemos calcular también cual sería el salario bruto hora.

$$\text{Horas anuales} = 40\text{h/semana} \times 52\text{semanas/año} = 2080\text{h/año}$$

Como el salario bruto es 25.000€ ahora podemos calcular el salario bruto/h

$$\text{Salario bruto por hora} = 25.000\text{€}/2080\text{h} = 12.02\text{€/h}$$

Sabiendo el salario bruto hora del trabajo, y las horas trabajadas podemos obtener cual ha sido el salario bruto mensual del trabajador.

$$\text{Horas trabajadas por semana} = 300\text{h} / (4\text{semanas/mes} \times 5\text{meses}) = 15\text{h/semana}$$

$$\text{Salario Bruto mensual} = 12.02\text{€/h} \times 15\text{h/semana} \times 4\text{semanas/mes} = \mathbf{721\text{€/mes}}$$

Se deben considerar también los costes que este trabajador genera para la empresa.

1. Seguridad Social: 23,60% contingencias comunes, 5,50% desempleo, 1,25% enfermedad profesional, 0,60% formación profesional, 0,20% FOGASA) = **31,15 % del salario neto**

2. IRPF: 14,09 % del salario neto

$$\text{Coste total para la empresa} = 721\text{€/mes} / (1 - 0.1409 - 0.3115) = \mathbf{1316.49\text{€/mes}}$$

Concepto	Coste (€)
Salario anual bruto	25.000
Seguridad Social	3.522,5
IRPF	7.787,50
Total (5 meses)	6.582,45

Tabla A.A.3-4 Coste del trabajador para la empresa

Coste total del proyecto

Coste total = 110,75€ + 200€ + 2000€ + 6582,45€ = 8893,20€

Por lo tanto, el coste total de proyecto asciende a **8893,20€**

Viabilidad legal

En el siguiente apartado se va a abordar el desarrollo del proyecto dentro de un marco legal, para tener en cuenta su correcta distribución, modificación y uso. Veremos que principalmente se han utilizado licencias con software libre. También existe alguna excepción, como es el caso de Micros Visual Studio Code, puesto que este mientras se utilice como una persona de trabajo libre si se aplica como una licencia de uso libre, en cambio en caso de que esta se utilizase realmente en un entorno empresarial debería adquirirse una licencia empresarial para el uso de este. [3]

A continuación, veremos las licencias utilizadas las cuales son las siguientes:

- **BSD-3:** Esta es una licencia permisiva, la cual tiene pocas limitaciones en su reutilización, permite tanto copiar, usar, modificar y distribuir el software, así como su uso en productos comerciales. Sus condiciones son que debe incluir en todas sus copias el aviso de copyright, y no puede utilizarse el nombre de la organización para promocionar productos derivados sin permiso [4]
- **MIT:** Está es una de las licencias más utilizadas en el software libre, MIT es una licencia permisiva, por lo que tiene pocas limitaciones en cuanto a su reutilización, permite tanto el uso, como la copia, modificación y distribución del software, así como también su uso en productos comerciales. La única condición es que debe incluirse el aviso de copyright en todas las copias del software [5]
- **Dominio público:** Esto quiere decir que no está cubierto por ninguna licencia por lo que no requiere atribución [6]

Herramienta	Licencia
Flask	BSD-3
Flask-Login	MIT
SQLAlchemy	MIT
Werkzeug	BSD-3
Bootstrap	MIT
SQLite	Dominio publico

Tabla A.A.3-5 Licencias

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este apartado, se van a especificar los distintos requisitos del trabajo, los cuales son necesarios para cumplir con las funcionalidad y objetivos del proyecto.

Se van a detallar las especificaciones de los distintos requisitos, asegurando de este modo que se cumplan las expectativas y necesidades del proyecto.

Para esto los requisitos se dividen en dos tipos.

- **Requisitos funcionales:** Definen las funciones y objetivos que debe tener el proyecto.
- **Requisitos no funcionales:** Definen la calidad y restricciones del sistema, así como su usabilidad y eficiencia.

B.2. Objetivos generales

Los objetivos generales del proyecto para cumplir con el objetivo y un correcto funcionamiento son los siguientes.

1. Desarrollar una aplicación colaborativa que permita a los usuarios gestionar archivos y tareas de manera personal, como poder compartir archivos, y establecer tareas mediante el uso de grupos.
2. Implementar un sistema de gestión de usuarios, grupos y permisos en los mismo.
3. Permitir a los usuarios crear, invitar y unirse a distintos grupos colaborativos.
4. Facilitar la colaboración y el intercambio de archivos entre los miembros de los equipos.

B.3. Catálogo de requisitos

A continuación, en la siguiente sección se van a listar y especificar los distintos requisitos que se han extraído a partir de los objetivos generales del trabajo.

Requisitos funcionales:

- **RF-1 Registrar Usuarios:** El usuario debe poder Registrarse en la aplicación.
- **RF-2 Autenticar Usuarios:** El usuario con la cuenta registrada debe poder iniciar sesión en la aplicación.
- **RF-3 Subir archivos:** La aplicación debe permitir subir y almacenar archivos de manera individual.
- **RF-4 Crear notas:** La aplicación debe permitir crear y gestionar tareas de manera individual.
- **RF-5 Crear grupos:** La aplicación debe permitir la creación y gestión de grupos.
- **RF-6 Compartir archivos:** La aplicación debe permitir compartir archivos en los grupos del usuario.
- **RF-7. Descarga de archivos:** La aplicación debe permitir a los usuarios la recuperación y descarga de los archivos almacenados y/o compartidos.

Requisitos no funcionales:

- **RNF-1. Usabilidad:** La aplicación debe ser intuitiva y fácil de usar, para todos los usuarios.
- **RNF-2. Escalable:** La aplicación debe ser escalable para poder conllevar líneas futuras de trabajo o la implantación de nuevas funcionalidades
- **RNF-3. Seguridad:** La aplicación debe garantizar la seguridad de los datos de los usuarios, así como las sesiones de los mismo.
- **RNF-4. Mantenible:** La aplicación de tener un código sencilla y documentado con una buena estructura para poder mantener y actualizar el proyecto.

B.3. Especificación de requisitos

A continuación, se va a visualizar un diagrama resumen sobre los distintos casos de uso que se definen en el trabajo, buscados a partir de los requisitos funcionales.

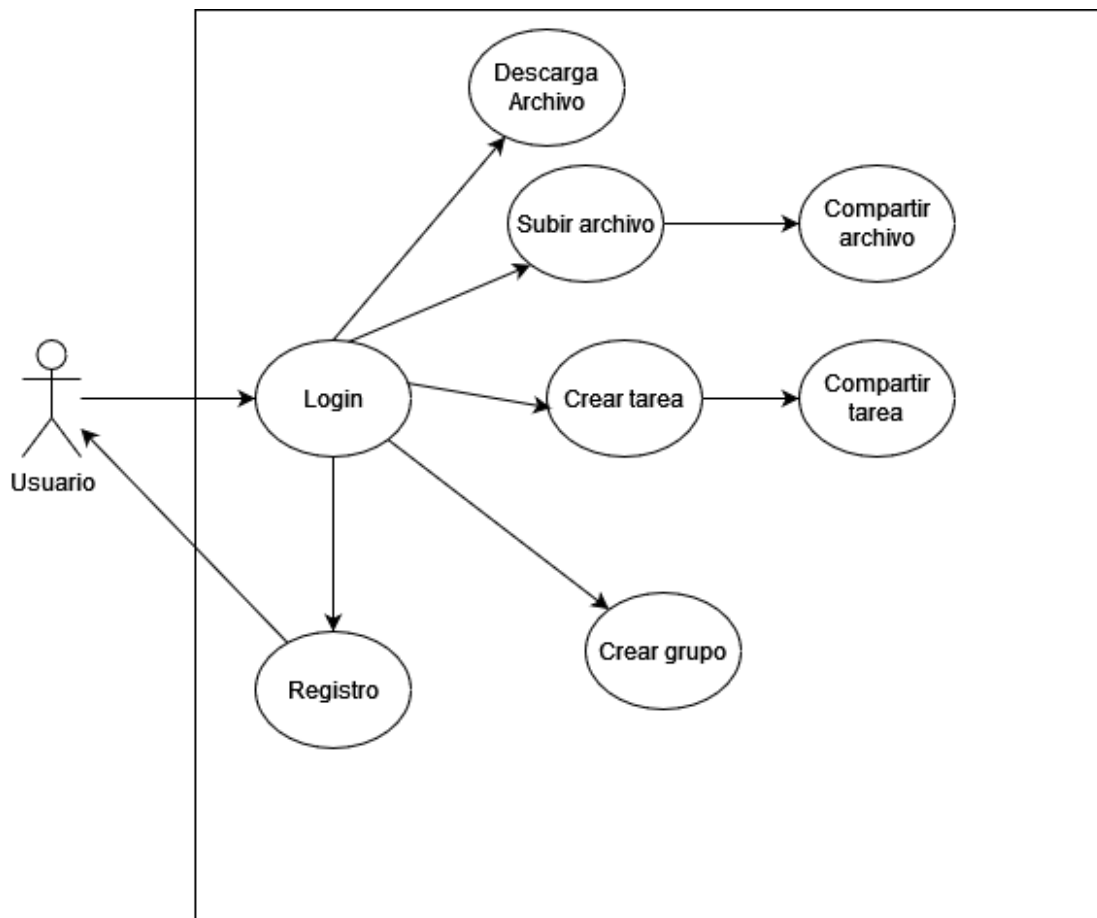


Ilustración 0.1 Casos de uso

CU-1	Registrar usuario
Versión	1.0
Requisitos asociados	RF-1
Autor	Álvaro Diez Sáenz
Descripción	Permite al usuario el registro en la aplicación para su posterior acceso
Precondición	Se debe disponer un email y acceder a /registro
Acciones	<ol style="list-style-type: none"> 1. Usuario debe abrir el navegador y acceder a /registro 2. Introducir su email, nombre y credenciales. 3. Si el usuario no existe se registra el usuario.
Postcondición	Usuario registrado
Excepciones	Email ya existente
Importancia	Alta

Tabla B.3B.3-1 CU-1 Registrar usuario

CU-2	Autenticar usuarios
Versión	1.0
Requisitos asociados	RF-2
Autor	Álvaro Diez Sáenz
Descripción	Permite al usuario el acceso a la aplicación mediante un email y contraseña y su posterior cierre de sesión.
Precondición	La cuenta debe estar dada de alta y acceder a /login.
Acciones	<ol style="list-style-type: none"> 1. Usuario debe abrir el navegador y acceder a /login 2. Introducir su email y credenciales correctas 3. Si es correcto accede al home del usuario
Postcondición	Acceder a la vista del home de usuario
Excepciones	Usuario / Contraseña inválida. El usuario no tiene una cuenta creada
Importancia	Alta

Tabla B.3B.3-2 CU-2 Autenticas usuarios

CU-3	Subir Archivos
Versión	1.0
Requisitos asociados	RF-3
Autor	Álvaro Díez Sáenz
Descripción	Permite al usuario subir archivos para su posterior almacenamiento.
Precondición	El Usuario debe estar logueado y debe subir un tipo de archivo válido
Acciones	<ol style="list-style-type: none"> 1. Usuario debe acceder a la página de archivos y tareas y presionar “subir archivo” 2. Seleccionar un archivo válido de su sistema 3. Se almacena archivo en la ruta correspondiente 4. Se almacena la operación con la ubicación del archivo en la base de datos.
Postcondición	Estar logueado en el sistema
Excepciones	Tipo de archivo no válido Usuario no logueado
Importancia	Alta

Tabla B.3-3 CU-3 Subir archivos

CU-4	Crear tarea
Versión	1.0
Requisitos asociados	RF-4
Autor	Álvaro Díez Sáenz
Descripción	Permite al usuario crear tareas en su espacio de trabajo personal.
Precondición	El Usuario debe estar logueado
Acciones	<p>1. Usuario debe acceder a la página de archivos y tareas y presionar “crear tarea”</p> <p>2. Introducir nombre y descripción de la tarea.</p> <p>4. Se almacena la tarea en la base de datos</p>
Postcondición	Estar logueado en el sistema
Excepciones	Usuario no logueado
Importancia	Alta

Tabla B.3B.3-4 CU-4 Crear tarea

CU-5	Crear grupos
Versión	1.0
Requisitos asociados	RF-5
Autor	Álvaro Díez Sáenz
Descripción	Permite al usuario crear grupos, donde invitar a otros usuarios y compartir archivos.
Precondición	El Usuario debe estar logueado
Acciones	<ol style="list-style-type: none"> 1. Usuario debe acceder a la página de mis grupos y presionar “Crear grupo” 2. Introducir nombre y descripción del grupo. 4. Se almacena el grupo creado en la base de datos y crea la relación del usuario con el grupo.
Postcondición	Estar logueado en el sistema
Excepciones	Usuario no logueado
Importancia	Alta

Tabla B.3B.3-5 CU-5 Crear grupos

CU-6	Compartir archivos
Versión	1.0
Requisitos asociados	RF-6
Autor	Álvaro Díez Sáenz
Descripción	Permite al usuario compartir archivos de su almacenamiento personal con los usuarios que formen parte de un grupo.
Precondición	El Usuario debe estar logueado
Acciones	<ol style="list-style-type: none"> 1. Usuario debe acceder a la página de mis grupos y presionar “Crear grupo” 2. Introducir nombre y descripción del grupo. 4. Se almacena el grupo creado en la base de datos y crea la relación del usuario con el grupo.
Postcondición	Estar logueado en el sistema
Excepciones	Usuario no logueado
Importancia	Alta

Tabla B.3B.3-6 CU-6 Compartir archivos

CU-7	Descargar archivos
Versión	1.0
Requisitos asociados	RF-7
Autor	Álvaro Díez Sáenz
Descripción	Permite al usuario la descargar o recuperación de los archivos almacenados o compartidos
Precondición	El Usuario debe estar logueado, y tener permisos sobre el archivo en cuestión
Acciones	<ol style="list-style-type: none"> 1. Usuario debe acceder a la ubicación del archivo, ya sea en su espacio personal o dentro de un grupo 2. Presionar sobre el botón de descarga en el archivo correspondiente 4. Comienza la descarga de los archivos en el equipo.
Postcondición	Estar logueado en el sistema
Excepciones	Usuario no logueado
Importancia	Alta

Tabla B.37 CU-7 Descargar archivos

Apéndice C

Especificación de diseño

C.1. Introducción

En la siguiente sección se van a detallar las distintas especificaciones del trabajo, el cual como se ha dicho previamente ha sido desarrollado con tecnologías como el framework Flask, SQLite o HTML y Bootstrap para el frontend, entre otras.

C.2. Diseño de datos

El diseño de datos para este proyecto se centra en el uso de la base de datos que brinda SQLite. Para estructurar y gestionar los datos se ha utilizado esta base de datos, juntos a la biblioteca SQLAlchemy compatible con Flask, brindando así la posibilidad del uso Object-Relational Mapping (ORM) [7] lo que agiliza y brinda una forma mucho más cómoda de gestionar los datos desde el código.

1. **Usuario:** Tabla que define los usuarios de la plataforma

- **Id** (Integer): Identificador único del usuario
- **usuario** (String): Nombre del usuario
- **email** (String): Dirección de correo electrónico único.
- **contraseña** (String): Contraseña del usuario

2. **Grupo:** Tabla que define los distintos grupos de la plataforma
 - **Id** (Integer): Identificador único del grupo
 - **nombre** (String): Nombre del usuario
 - **email** (Text): Descripción del grupo
 - **Contraseña** (String): Contraseña del usuario
3. **GrupoUsuario:** Tabla que define la relación muchos a muchos entre usuarios y grupo.
 - **Id** (Integer): Identificador único de la relación
 - **usuario_id** (Integer): Identificador del usuario
 - **grupo_id** (Integer): Identificador del grupo
4. **Archivo:** Tabla que representa los distintos archivos subidos por los usuarios.
 - **Id** (Integer): Identificador único del archivo
 - **nombre** (String): Nombre del usuario
 - **ruta** (String): Ruta del archivo en el sistema de archivos
 - **propietario_id** (String): Identificador del usuario que subió el archivo
 - **fecha_subida** (DateTime): Fecha y hora a la que se subió el archivo
5. **ArchivoGrupo:** Tabla que define la relación muchos a muchos entre grupos y archivos.
 - **Id** (Integer): Identificador único de la relación.
 - **archivo_id** (Integer): Identificador único del archivo
 - **grupo_id** (Integer): Identificador único del grupo

6. **Tarea:** Tabla que representa las distintas tareas creadas por los usuarios.

- **Id** (Integer): Identificador único de la tarea
- **titulo** (String): Título de la tarea
- **descripción** (String): Descripción de la tarea
- **propietario_id** (String): Identificador del usuario que creó la tarea
- **fecha_subida** (DateTime): Fecha y hora a la que se creó la tarea.

7. **TareaGrupo:** Tabla que define la relación muchos a muchos entre grupos y tareas.

- **Id** (Integer): Identificador único de la relación.
- **tarea_id** (Integer): Identificador único de la tarea
- **grupo_id** (Integer): Identificador único del grupo

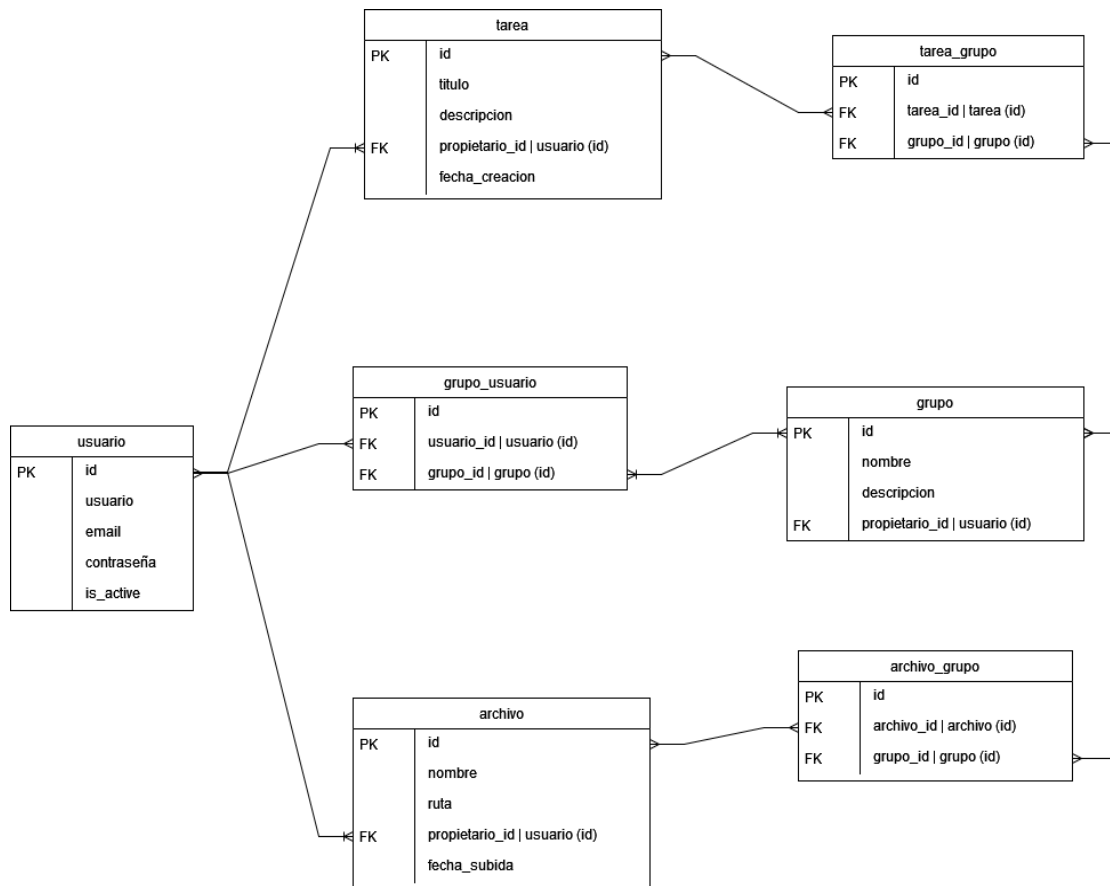


Ilustración 0.1 Diagrama Entidad Relación

C.3. Diseño procedimental

El diseño procedimental se enfoca en definir los distintos procesos y como se gestionan estas peticiones desde la llamada del cliente hasta ser procesada por el frontend y posteriormente devuelta. Para esto van a describir las distintas interacciones clave y a representar sus acciones mediante los correspondientes diagramas de secuencia.

1. Registro de usuario

- Usuario acceder a página de registro, introduce su nombre, email y contraseña
- Si los datos son válidos, la cuenta es creado y el usuario es redirigido a la vista de inicio de sesión
- En caso de ser invalidad se pide introducir de nuevo unos datos de registro mediante un mensaje de error.

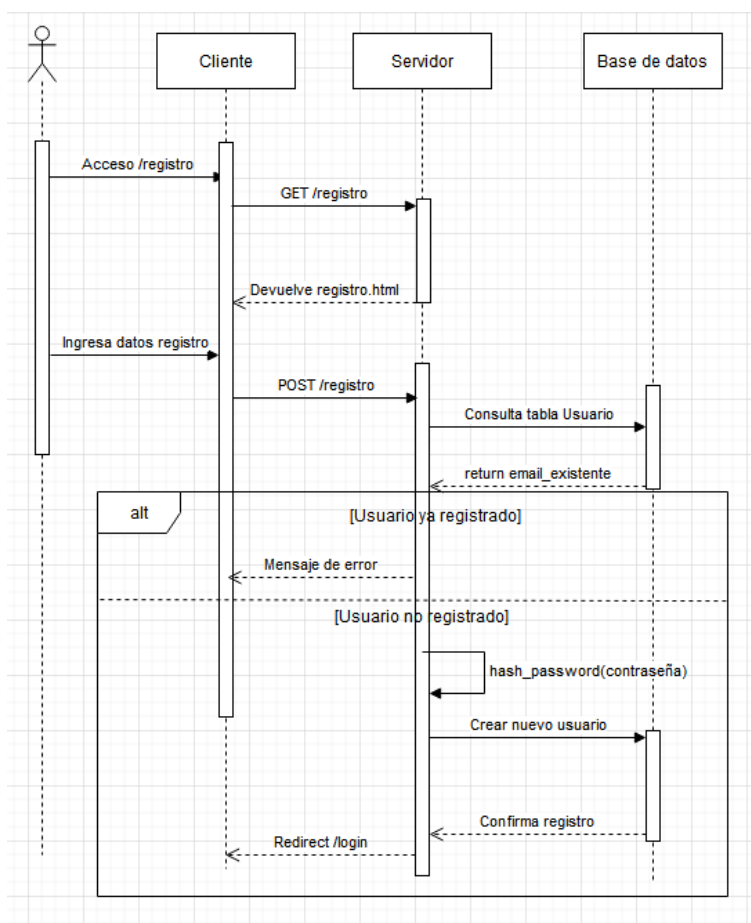


Ilustración 0.2 Diagrama secuencia registro usuario

2. Inicio de sesión

- Usuario introduce correo electrónico y contraseña en el formulario de inicio de sesión
- Si los datos son válidos, el usuario inicia sesión y es redirigido a la página de inicio.
- En caso de que el login no sea correcto se muestra un mensaje de error.

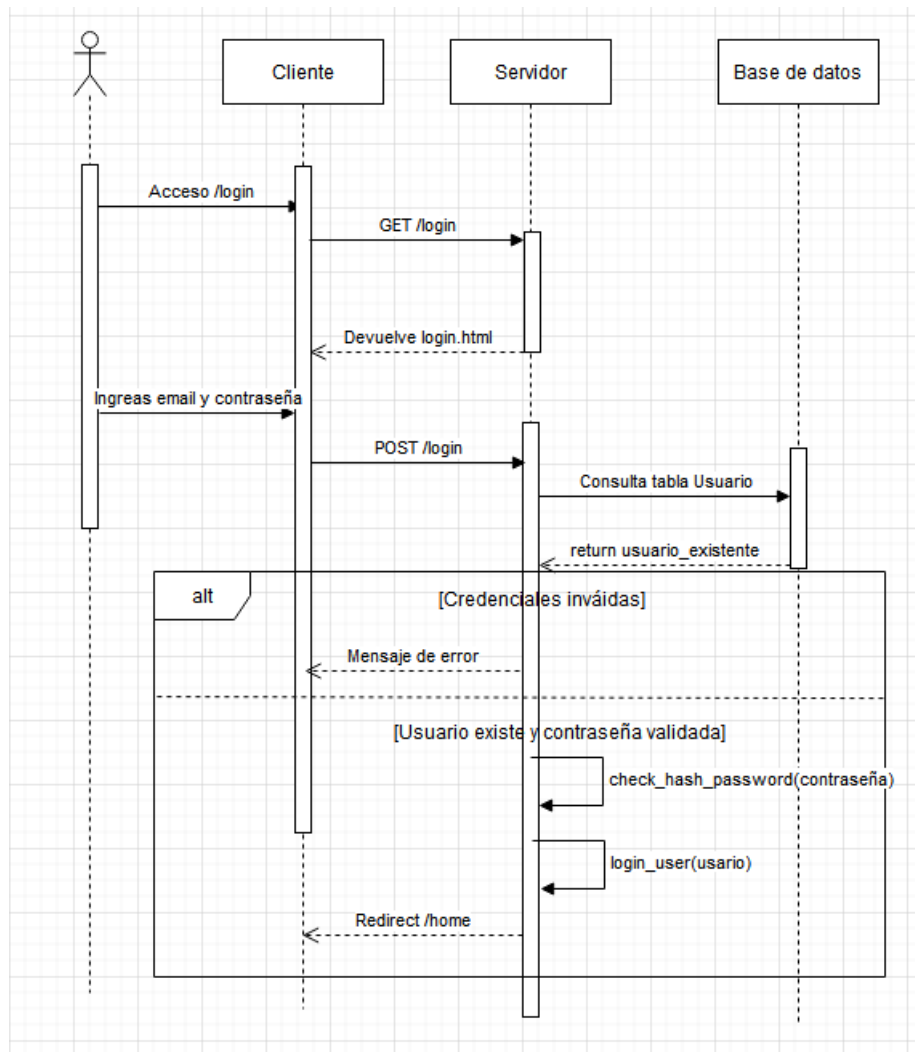


Ilustración 0.3 Diagrama secuencia inicio de sesión

3. Gestión de archivos

- El usuario puede subir archivos a su espacio de trabajo individual a través de un formulario de carga de archivos.
- Los archivos subidos son almacenados en un path específico y registrados en la base de datos.
- Los usuarios pueden eliminar sus archivos, por lo que son eliminados del sistema de archivos y borrado su registro de la base de datos.

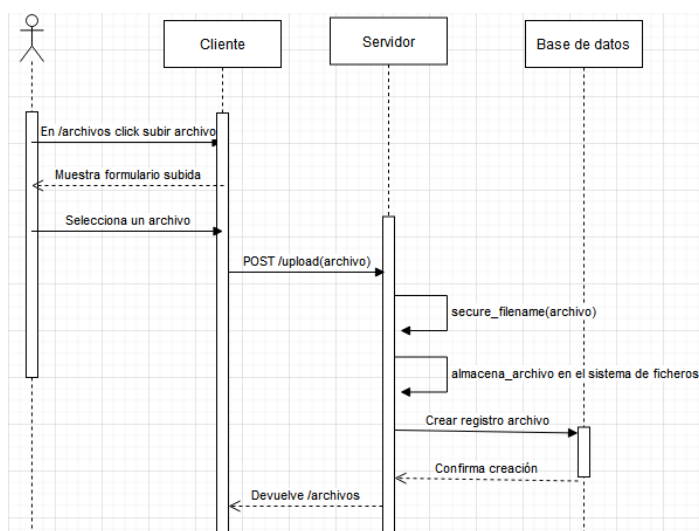


Ilustración 0.4 Diagrama secuencial subida de archivos

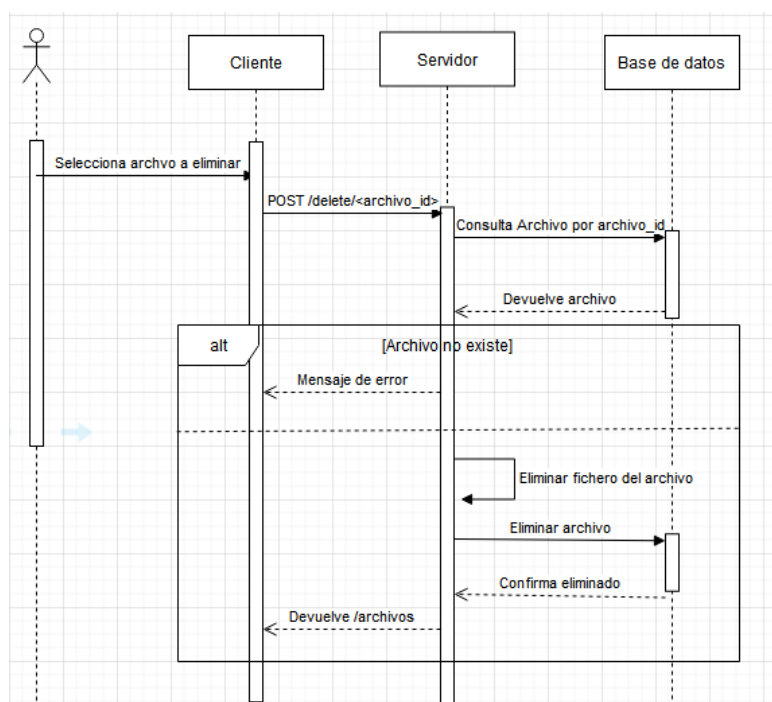


Ilustración 0.5 Diagrama secuencial eliminado archivos

4. Gestión de tareas

- El usuario puede crear sus propias tareas, en su espacio de trabajo individual a través de un formulario donde rellenar nombre y descripción de la misma.
- Las tareas son asignadas al usuario y almacenadas
- El usuario puede visualizar, modificar y eliminar la tarea desde su espacio personal

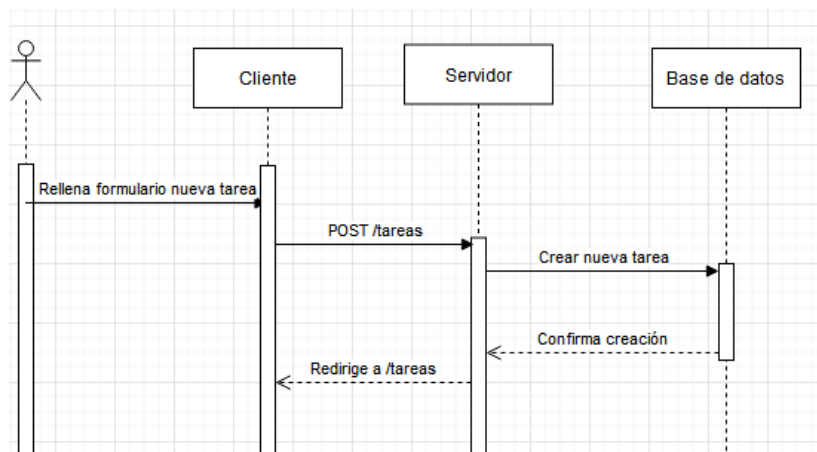


Ilustración 0.6 Diagrama secuencial creación tarea

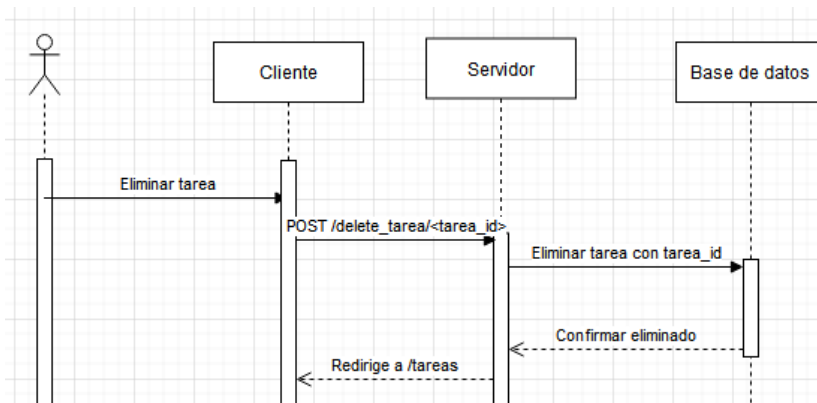


Ilustración 0.7 Diagrama secuencial eliminación tareas

5. Gestión de grupos

- Los usuarios pueden crear nuevos grupos introduciendo un nombre y una descripción para estos.
- Pueden unirse nuevos usuarios a estos grupos, siendo invitados por los administradores de los grupos.
- Los usuarios pueden visualizar los distintos archivo y tareas asociados a ese grupo

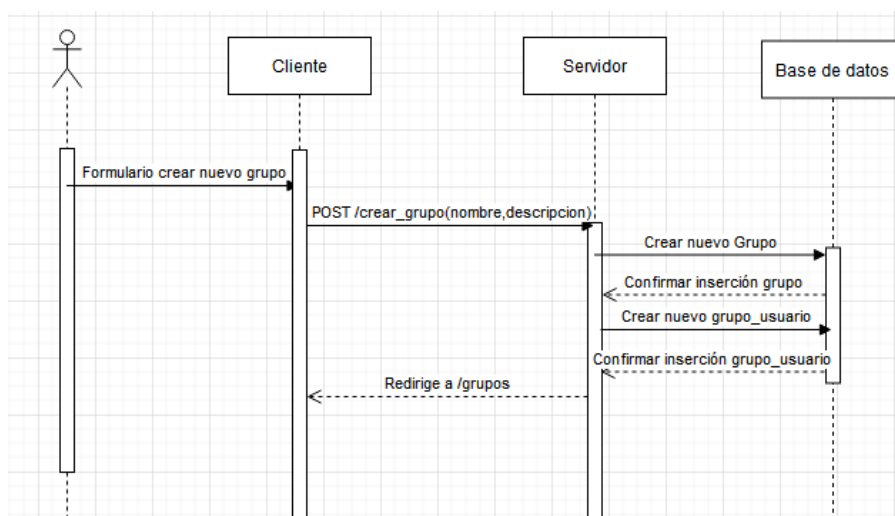


Ilustración 0.8 Diagrama secuencia crear grupo

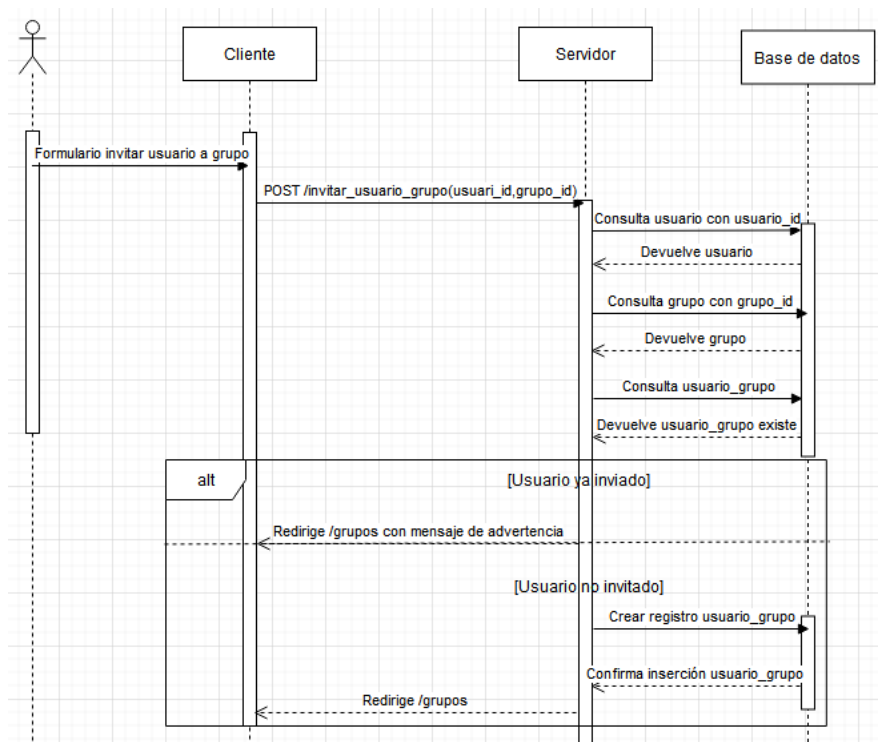


Ilustración 0.9 Diagrama secuencia invitar usuario

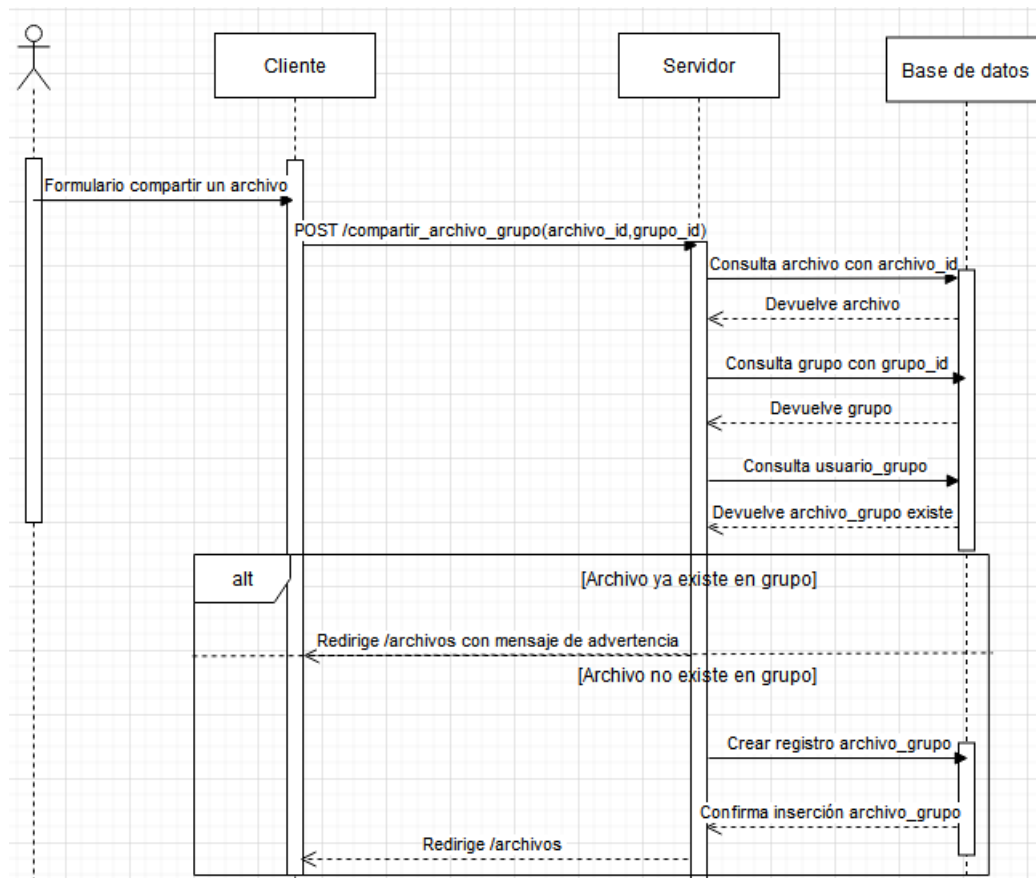


Ilustración 0.10 Diagrama secuencia compartir archivo

C.4. Diseño arquitectónico

El diseño arquitectónico del trabajo se basa en el patrón Modelo-Vista-Controlador, como ya hemos dicho previamente se ha utilizado el framework Flask, el cual brinda de una manera sencilla y cómoda, la posibilidad de utilizar esta arquitectura.

Modelo-Vista-Controlador

La arquitectura del sistema se basa en el patrón Modelo-Vista-Controlador o MVC, esto nos permite tener un diseño modular y escalable que permite de una forma cómoda separar la lógica y gestión de datos, de la interfaz de usuario.

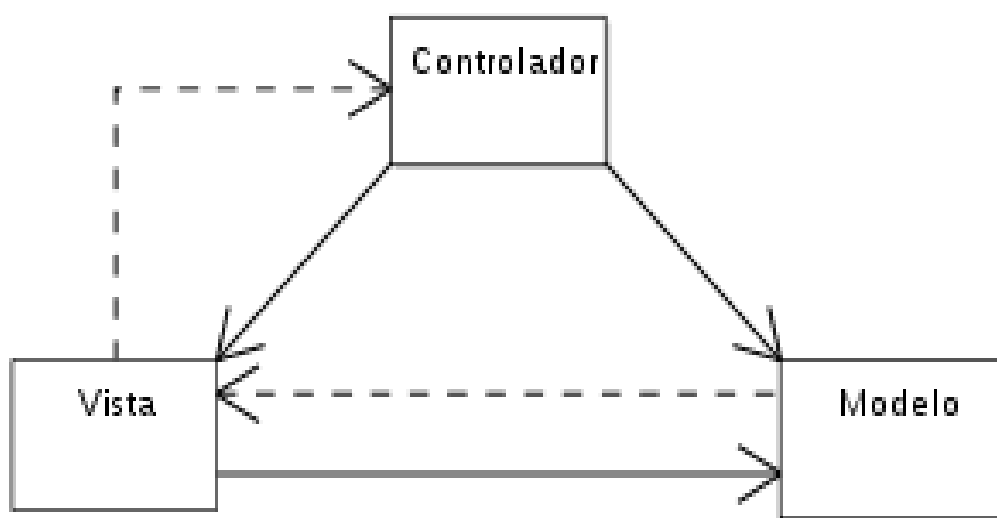


Ilustración 0.11 Modelo-Vista-Controlador [8]

A continuación, se van a definir las 3 entidad de la arquitectura Modelo-Vista-Controlador y como se utilizan estas en las distintas partes del trabajo.

- **Modelo:** El modelo gestiona, tanto la estructura de datos como la lógica de trabajo de la aplicación. En este trabajo, los modelos son definidos mediante SQLAlchemy, la cual brinda la posibilidad de mapear estas tablas a objetos para Python, los modelos de los que disponemos son “Usuario”, “Grupo”, “Archivo” y “Tarea” aquí se gestiona la información y las relaciones de las distintas entidades

Vista: La vista es la parte visual de la aplicación se encarga de la presentación de la interfaz del usuario. Utiliza plantillas HTML junto a CCS y otras herramientas de frontend como Bootstrap, estas proporcionan la interfaz visual con la que interactúa el Usuario. En este trabajo Flask renderiza las plantillas y las combina con los datos que proporciona el modelo para de esta forma renderizar y mostrar la interfaz final.

■ **Controlador:** El controlador es el responsable de actuar como intermediario entre el modelo y la vista de la aplicación, este gestiona todas las solicitudes entrantes y llama a la lógica correspondiente de la aplicación para que esta realice las operaciones correspondientes con los datos, después devuelve las vistas correspondientes generadas. Con el uso de Flask, las rutas que están definidas en “app.py” son las que actúan como controladores y manejan la lógica de la aplicación.

Como podemos ver con este patrón MVC se proporciona una separación de responsabilidad en las distintas partes del proyecto por lo que facilita el desarrollo y el mantenimiento del mismo. Eso también permite una mayor flexibilidad para los futuros cambios y evolución del trabajo.

Apéndice *D*

Documentación técnica deprogramación

D.1. Introducción

En este apartado se va a documentar una guía detallada, donde se van a cubrir aspectos esenciales como la estructura de los directorios, manual del programado, instrucciones para la compilación, instalación y ejecución del trabajo. Todo esto con el objetivo de facilitar el mantenimiento y comprensión del código y ser guía práctica para poder continuar mediante nuevas funciones o futuras líneas de trabajo.

D.2. Estructura de directorios

La estructura de directorios de proyecto está organizada de tal forma que facilite la gestión y acceso a los distintos componentes del sistema. A continuación, se van a detallar los distintos directorios, que pueden visualizarse en el directorio de GitHub del trabajo

- **/:** Directorio raíz del trabajo que contiene el todo el código fuente
- **/app.py:** Archivo principal del proyecto Flask, en él se configuran las rutas, se inicializa y define su comportamiento
- **/models.py:** Archivo donde se definen los modelos de datos utilizando SQLAlchemy

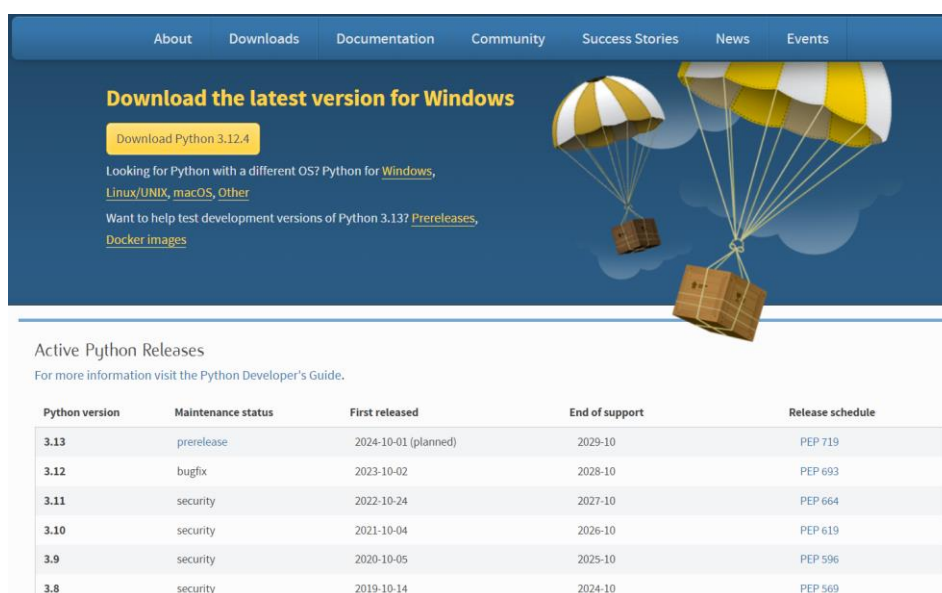
- **/files/:** Directorio donde se almacenan los archivos subidos por los usuarios.
- **/instance/:** Directorio utilizado por Flask, donde almacena las configuraciones específicas de la instancia de la aplicación, como por ejemplo el archivo de base de datos local.
- **/static/:** Directorio que contiene los archivos estáticos como hojas de estilo CSS
 - **/static/style.css:** Hoja de estilo general para el trabajo.
 - **/static/home.css:** Hoja de estilo específica para algunas partes del trabajo
- **/templates/:** Directorio que contiene las plantillas HTML utilizadas
 - **/templates/archivos.html:** Plantilla para la página principal de archivos del usuario
 - **/templates/base.html:** Plantilla base que sirve como base para otras plantillas.
 - **/templates/grupo.html:** Plantilla para la página de un grupo específico.
 - **/templates/grupos.html:** Plantilla para la visualización de todos los grupos.
 - **/templates/home.html:** Plantilla para la página de inicio.
 - **/templates/login.html:** Plantilla para la vista de inicio de sesión
 - **/templates/registro.html:** Plantilla para la vista de registro
 - **/templates/tareas.html:** Plantilla para la vista de gestión de tareas.
 - **/templates/uploadPop.html:** Plantilla para la ventana emergente de subida de archivos
- **/.gitignore:** Archivo que define que archivos o directorios debe ignorar Git, como los generador automáticamente o configuraciones locales
- **/requirements.txt:** Archivo que lista todas las dependencias del proyecto.
- **/README.md** Fichero utilizado para desplegar en la nube nuestra aplicación web.

D.3. Manual del programador

En el siguiente apartado Manual del programador se van a proporcionar instrucciones sobre los softwares necesarios para el correcto funcionamiento del trabajo.

Software

Python es el lenguaje principal del trabajo, por lo que para poder utilizar y hacer funcionar el proyecto es necesario tener previamente instalado y configurado este lenguaje, para la descarga de Python podemos utilizar el siguiente enlace <https://www.python.org/downloads/>



Download the latest version for Windows

[Download Python 3.12.4](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python 3.13? [Prereleases](#), [Docker images](#)

Active Python Releases
For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.13	prerelease	2024-10-01 (planned)	2029-10	PEP 719
3.12	bugfix	2023-10-02	2028-10	PEP 693
3.11	security	2022-10-24	2027-10	PEP 664
3.10	security	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569

Ilustración D.0.1 Pagina descarga Python [9]

Una vez instalado Python debe configurarse de forma correcta en el Path del sistema. Para que Python funcione correctamente este debe configurarse como una variable más del sistema, por lo que a continuación vamos a ver los distintos pasos necesarios para su correcta configuración.

1º En el equipo acceda a “Configuración avanzada del sistema” puede encontrarse buscando directamente el nombre en la barra de búsqueda de Windows.

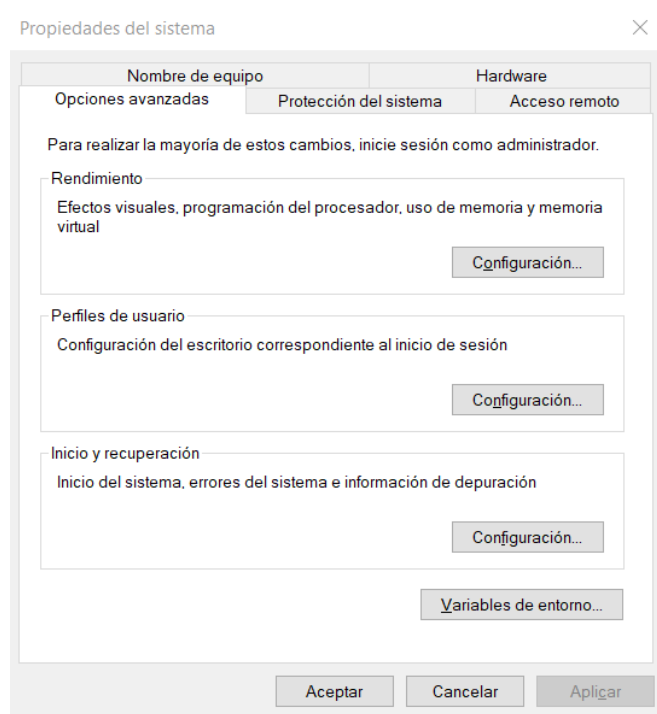


Ilustración 0.2 Path del sistema

2º Una vez aquí se hace click sobre el botón “Variables de entorno”

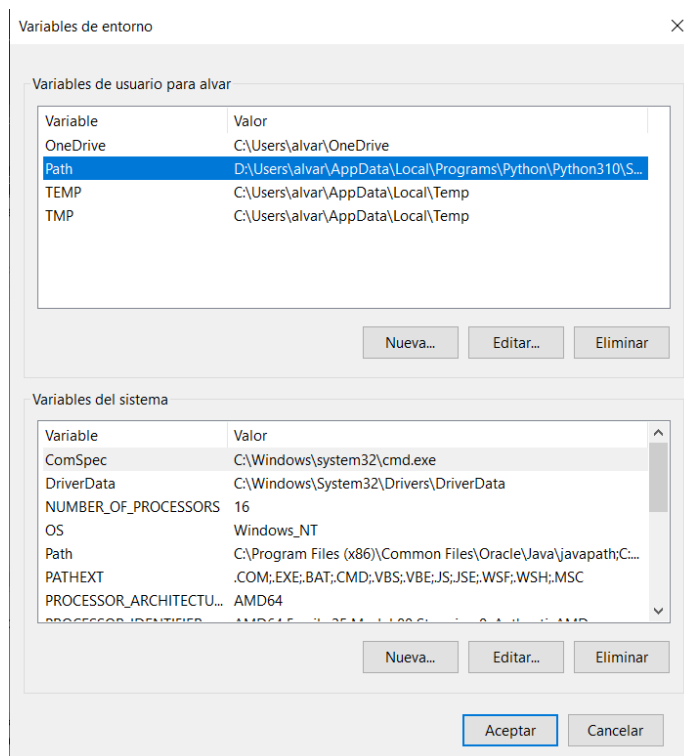


Ilustración 0.3 Configuración path del sistema

3º Ahora se debe localizar la variable “Path” y presionar sobre editar para especificar la ruta específica Python.

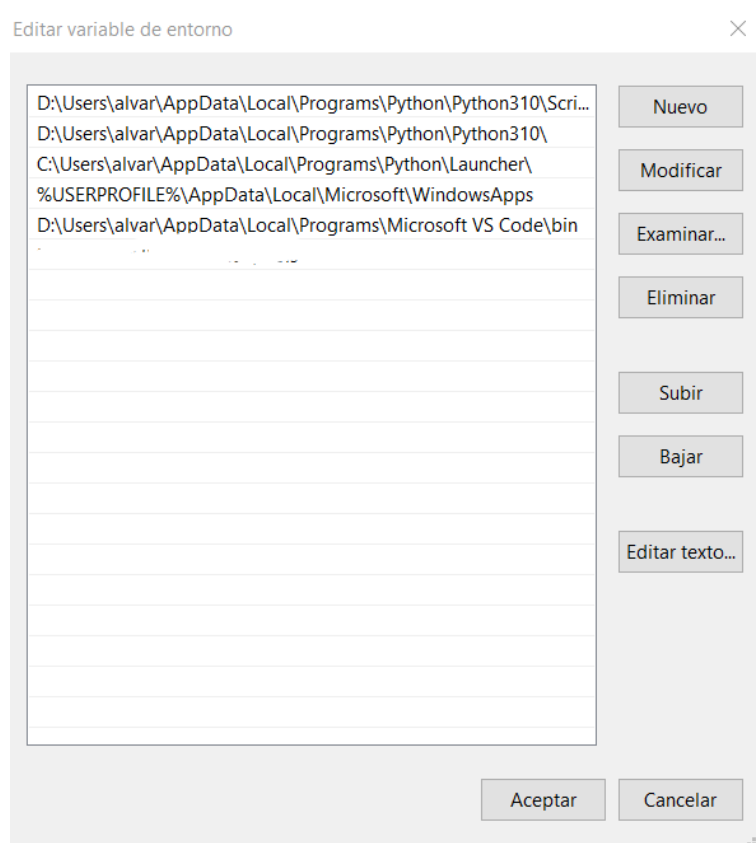


Ilustración 0.4 Variables del sistema

4º Como vemos en la imagen se deben crear las rutas de Python tanto “Python310” como “Python310/Scripts”. Una vez configuras se presiona sobre “Aceptar” y ya estará configurado Python como variable en nuestro sistema.

Sumado al propio lenguaje de programación, para facilitar el desarrollo y el control de versiones del trabajo, se recomienda el uso de Visual Studio Code, ya que ha sido el editor de código utilizado durante el trabajo, y el uso de Git para seguir manteniendo el control de versiones.

Visual Studio Code es un entorno de desarrollo o editor de código gratuito y multiplataforma que ha sido desarrollado por

Microsoft. Ofrece una gran cantidad de características que facilitan el desarrollo de código dentro del entorno, también incluye soporte para depuración, resaltado de sintaxis y además cuenta con una gran cantidad de extensiones disponibles.

Para descargar Visual Studio debemos acceder a su página principal de descarga y comenzar la descarga para Windows o tu sistema operativo de preferencia. Puedes descargar Visual Studio Codes desde este link <https://code.visualstudio.com/>

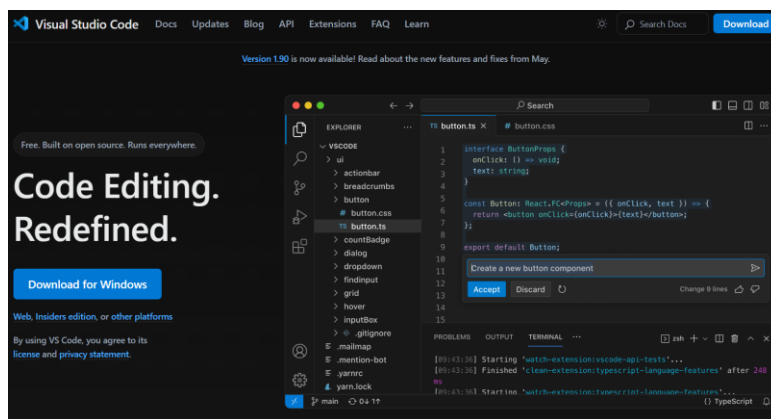


Ilustración 0.5 Descarga Visual Studio Code

Una vez descargado ejecuta el archivo .exe descargado y sigue las instrucciones de instalación que se muestran en Visual Studio Code.

Una vez correctamente instalado cuando abras Visual Studio Code deberás encontrar una ventana como esta, donde vemos las novedades y actualizaciones del editor.

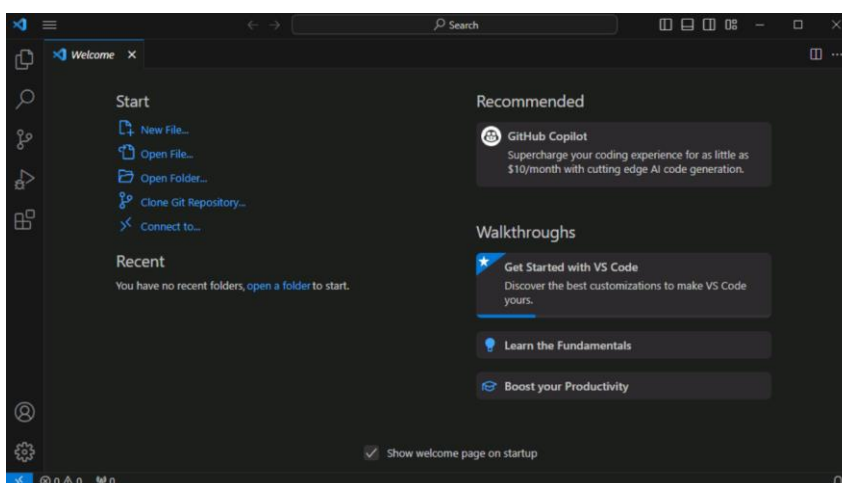


Ilustración 0.6 Ventana Visual Studio Code

Instalación de Git para el control de versiones, como ya hemos dicho es altamente recomendable contar con Git instalado en el sistema para de este modo seguir manteniendo

un flujo fluido de control de versiones.

Para descargar Git, lo primero es acceder a la web oficial y descargar la versión correspondiente a tu sistema operativo. Puedes descargar Git desde el siguiente link <https://git-scm.com/downloads>

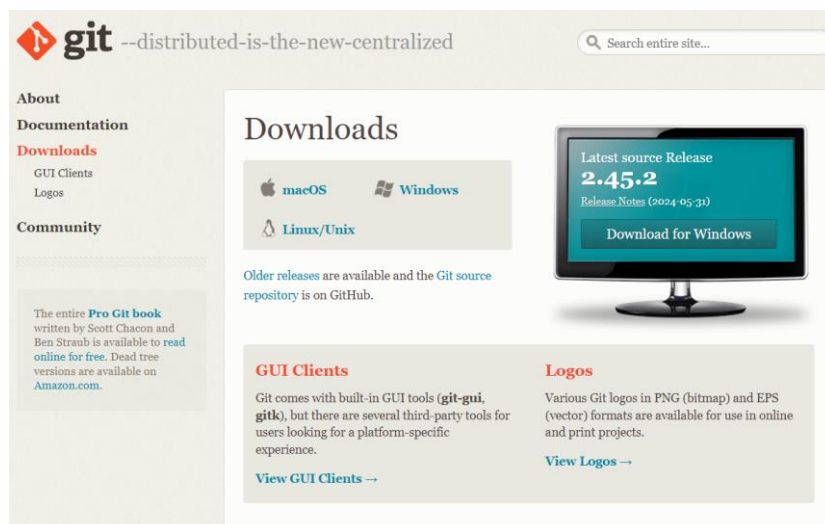


Ilustración 0.7 Descarga Git [10]

Una vez descargado ejecuta el instalador y sigue las instrucciones del instalador, algunos de los pasos más importantes son la instalación de componentes de Git, en esta ventana mantener la instalación por defecto a menos que se quiera agregar algún componente extra a Git

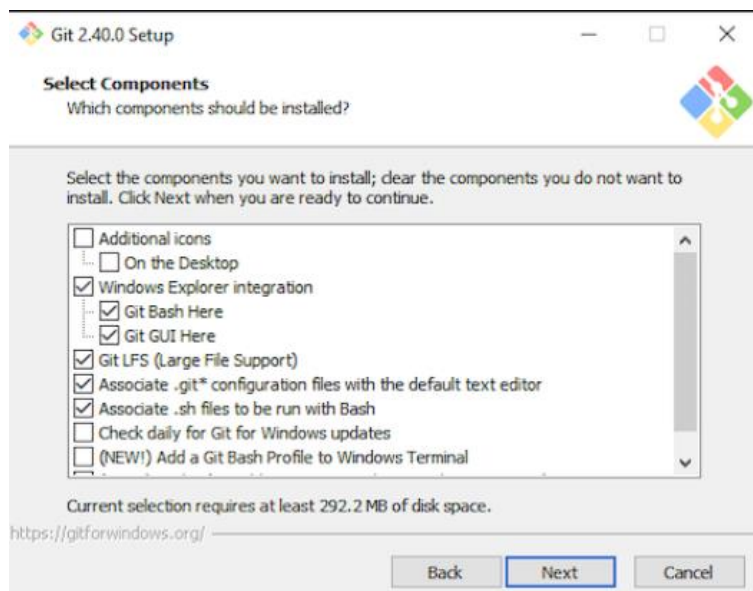


Ilustración 0.8 Instalación Git

Una vez completadas todas las instrucciones del instalador Git debería estar funcionando correctamente en el sistema. Para verificar esto puede utilizarse el siguiente comando

- `git --versión`

Y se debería visualizar una respuesta similar a la siguiente.

```
C:\Users\alvar>git --version
git version 2.45.0.windows.1
```

Ilustración 0.9 git --versión

Una vez se ha comprobado que Git esta instalado correctamente debemos configurar nuestro nombre de usuario y nuestra dirección de correo, esto es importante ya que es la información que Git va a utilizar para los “commits” y será introducida en cada “commit” realizado desde el sistema.

Para esto deben utilizarse los siguientes comandos

- `git config --global user.name " Nombre"`
- `git config --global user.email “email@ejemplo.com”`

Con esto configurado podemos utilizar el siguiente comando en nuestro equipo, donde podremos la configuración de Git en nuestro sistema

```
C:\Users\alvar>git config --list
diff.astextplain.textconv=astextplain
http.sslbackend=openssl
http.sslcainfo=D:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
init.defaultbranch=master
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Álvaro Díez Sáenz
user.email=ads1010@alu.ubu.es
pull.rebase=true
core.editor="D:\Users\alvar\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
```

Ilustración 0.10 Configuración git

Estructura y nuevas rutas

En este manual también vamos a revisar el correcto entendimiento del código junto a la revisión de las directrices para agregar nuevas funcionalidades, mejorar las ya existentes o trabajar en líneas futuro de trabajo.

Lo primero debemos conocer cuales los archivos o directorios donde se encuentran las distintas rutas, modelos o plantillas.

- **Modelos:** Los modelos están definidos en el archivo “models.py”, como ya sabemos estos representan las tablas que se crean en la base de datos.
- **Rutas (endpoint):** Las rutas o controladores se definen en el archivo “app.py”, cada una de estas rutas maneja los distintos endpoint y la lógica de la aplicación. Aquí se crean y modifican las rutas de las solicitudes web.
- **Templates:** Las plantillas HTML están en el directorio “templates/”, Aquí se crean los archivos .html que después se procesan por las rutas para ser devueltos con el contenido dinámico.

Agregar una nueva ruta

Para agregar una nueva ruta que pueda ser accesible desde la aplicación es necesario agregar un nuevo endpoint. La estructura de nuevo endpoint sería la siguiente:

```
@app.route('/nueva_ruta',methods=['GET'])
@login_required
def nuevo_endpoint():
    # Acciones a realizar
    return render_template('nueva_plantilla.html', nombre_usuario=current_user.usuario)
```

Ilustración 0.11 Ejemplo ruta app.py

Como vemos lo primero debemos definir “@app.route” y sus parámetros como son el nombre de la ruta y el tipo de petición que se espera GET o POST, a continuación colocamos el decorador “@login_required” y finalmente definimos una función, con el nombre que queramos, dentro de esta definimos las acciones a realizar y finalmente es importante devolver la plantilla que será visualizada por el navegador, como se ve en la imagen también estamos incluyendo una variable al renderizar la plantilla, esto es para posteriormente poder utilizar esa variable dentro del archivo .html y dinamizar nuestras plantillas.

Agregar nuevos modelos

Para agregar un nuevo modelo, es decir crear una tabla en la base de datos, necesitamos definirla en el archivo `models.py`. Esto es tan sencillo como definir la clase correspondiente en el archivo, como el ejemplo que se muestra a continuación:

```
class NuevoModelo(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    nombre = db.Column(db.String(80), nullable=False)
    descripcion = db.Column(db.String(200), nullable=True)
```

Ilustración 0.12 Ejemplo modelo en `models.py`

Como vemos, únicamente tenemos que definir una clase con el nombre que queramos para la tabla, y a continuación definir sus columnas como atributos de la clase.

Es importante decir que la creación de tablas o modificación de estas puede generar conflictos si el archivo base de datos SQLite se ya había sido creado, una solución para esto es el borrado completo del archivo, o mas interesante usar bibliotecas de migración que el propio Flask incluye, como pueden ser Flask-migrate, esto nos permite de una forma sencilla actualizar correctamente el archivo SQLite.

Agregar nuevas plantillas

Al agregar nuevas plantillas vamos a crear un nuevo archivo `.html` convencional, pero para dar ese dinamismo a las plantillas podemos utilizar la llamada desde del archivo `app.py` para introducir esas variables en ella. En nuestro caso también vamos a extender “`base.html`” ya que está es la base de las plantillas de nuestra aplicación donde se incluye los encabezados y el header y footer.

Un ejemplo de una plantilla muy básica seria la siguiente.

```
{% extends "base.html" %}
{% block content %}
<h1>Hola {{ nombre_usuario }}</h1>
<!-- Resto de estructura HTML -->
{% endblock %}
```

Ilustración 0.13 Ejemplo HTML /templates

Como vemos lo primero que se encuentra en la extensión del “`base.html`” este incluye todos los encabezados como las librerías por ejemplo Bootstrap, y a continuación se introduce todo el contenido HTML necesario dentro de “`{% block content %}`”, como vemos dentro de esta estructura HTML podemos utilizar “`{{ variable }}`” para insertar la variable que envíe la ruta del `app.py`

D.4. Compilación, instalación y ejecución del proyecto

El trabajo se basa en una aplicación web en Python, como sabemos este lenguaje no es compilado si no interpretado, por lo que no se dispone de un ejecutable, por lo que ejecutaremos el programa mediante su interprete en nuestro IDE de preferencia, como puede ser Visual Studio Code, entre otros.

Descarga del repositorio

Para descargar el código se debe acceder a repositorio de GitHub el cual esta disponible en el siguiente enlace: <https://github.com/ads1010/TFG>

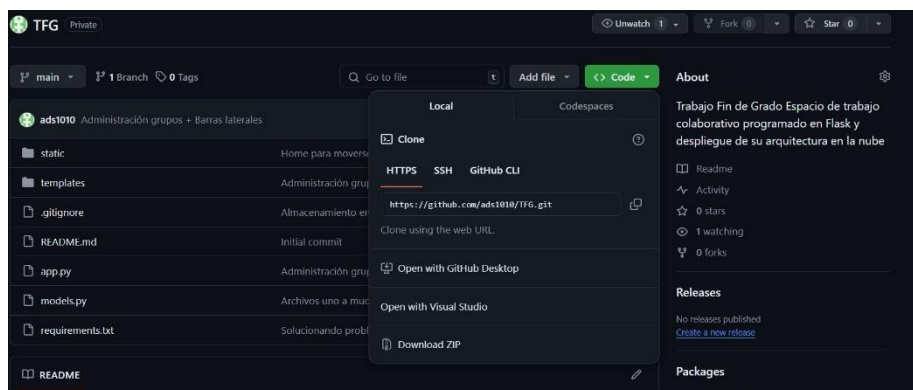


Ilustración 0.14 Repositorio git del proyecto

Como vemos es posible tanto la descarga de un archivo .zip donde dispondremos de todo el código fuente, como también es posible la descarga mediante el uso de Git el cual hemos visto previamente su instalación, para ello tendremos que usar el siguiente comando, ubicados en la dirección donde se desee realizar la descarga.

- “git clone https://github.com/ads1010/TFG”

Una vez descargado y abierto el repositorio, será necesario instalar todas las librerías necesarias para el correcto funcionamiento del código. Las librerías necesarias están recogidas en el fichero requirements.txt, para descargar todas estas de una manera cómoda, desde la consola o la terminal de nuestro entorno de desarrollo, introducimos el siguiente comando:

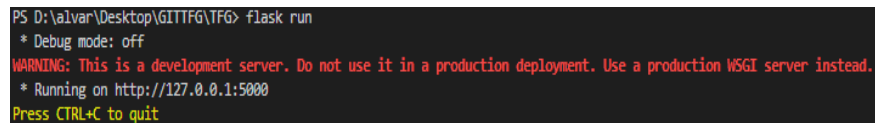
- **Pip install -r requirements.txt**

Esto comenzará automáticamente la descarga de todas las librerías.

Una vez la ejecución haya finalizado, tendremos la aplicación lista para lanzar en nuestro entorno local. Para lanzar la aplicación de nuevo debemos estar ubicados en el directorio del donde se encuentre el código fuente e introducir el siguiente comando en la terminal de nuestro entorno de desarrollo.

- **flask run**

Y veremos en la terminal como nos mostrará la dirección URL donde se ha desplegado nuestro proyecto en local, abrimos el navegador y acceder a la siguiente dirección, donde finalmente podremos visualizar la web



```
PS D:\alvar\Desktop\GITTFG\TFG> flask run
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Ilustración 0.15 Ejecución Flask

D.5. Pruebas del sistema

En el siguiente apartado vamos a revisar las pruebas que se han realizado en el sistema, estas son una parte imprescindible para comprobar que las nuevas funcionalidades implementadas se adaptan y funcionan correctamente en todo el sistema. Se han llevado a cabo las distintas pruebas durante los distintos ciclos del desarrollo iterativo e incremental, de esta forma se identifican y solucionan los problemas encontrados en cada ciclo de desarrollo y con cada modificación o incorporación de funcionalidades. A continuación, se van a detallar los distintos tipos de pruebas realizadas.

Pruebas unitarias:

Estas pruebas se realizaron durante el desarrollo de cada nueva funcionalidad, para poder comprobar que las nuevas funcionalidades individuales funcionaban de forma correcta. Estas pruebas se centran en los propios métodos en sí y la función del código. Algunos ejemplos de las pruebas unitarias son los siguientes:

Autenticación de Usuarios: Se realizaron las pruebas unitarias sobre la autenticación de los usuarios, cuando se desarrollaban o modificaban nuevas funcionalidades sobre estos. Aquí se verifican tanto los procesos de registro, como inicio de sesión, así como pruebas en cuanto al correcto funcionamiento de las credenciales y gestión de las sesiones activas o cerradas como corresponda.

Gestión de Archivos: Tanto cuando se implementa la primera subida y gestión de los archivos, como cuando se han realizado modificaciones o ampliaciones de esta funcionalidad se han realizado pruebas unitarias. En estas se comprueba desde la verificación del correcto almacenaje y gestión de los archivos en la base de datos y en los directorios como sobre la protección de estos ficheros y que no pudieran ser accedidos por otros usuarios.

Gestión de Tareas: Al igual que con la gestión de los archivos, cuando se realizan cambios sobre las funcionalidades que afectaban a la gestión de tareas, se realizaban las pruebas unitarias para comprobar que todo funcionara correctamente. Se verificaba tanto la correcta creación de esta como sus correctas relaciones en la base de datos.

Pruebas de Integración

Durante el proceso también se han realizado pruebas de integración, estas complementan a las pruebas unitarias, ya que comprueban la integración entre las diferentes funcionalidades o componentes que se van agregando o modificando. En estas pruebas se verifica el correcto flujo de trabajo completo, donde se interactúa con funcionalidades distintas que trabajan juntas en la aplicación: Algunos ejemplos de estas pruebas son los siguientes:

Flujo completo de un usuario: En estas pruebas se simula un flujo completo de uso de un usuario, y cómo interactúa este con los distintos elementos de la aplicación, por ejemplo, un flujo completo del usuario se compone de un registro, login del usuario, subida de archivos, creación de tareas, y gestión de estas, comprobando que todo funcione correctamente.

Gestión de grupo: También se han realizado verificación sobre el flujo completo de la gestión de grupos, verificando todas las funcionalidades que actúan sobre estos, como es la creación del grupo, asignación de usuarios, gestión de archivos, y tareas etc. Estas pruebas se centran en asegurar correctamente el acceso a los grupos por parte de los miembros.

Pruebas de seguridad

Puesto que en la aplicación se gestionan algunos datos sensibles, como los archivos personales y tareas de cada usuario, se llevan a cabo distintas pruebas que aseguran la integridad de estos datos. Estos son algunos ejemplos de las pruebas de seguridad.

Autenticación y Autorización: Verificando que únicamente los usuarios autenticados puedan acceder a los datos almacenados, es decir que se limitan correctamente los accesos a las distintas funcionalidades, únicamente para el usuario correspondiente.

Apéndice *E*

Documentación de usuario

E.1. Introducción

En el siguiente apartado se va a explorar el uso completo de la aplicación desde la vista de un usuario de esta, s

Apartado destinado a tratar lo referente al ámbito del usuario. Se tratarán principalmente temas relativos a los requisitos la instalación del proyecto y el manual de uso enfocado al usuario.

E.2. Requisitos de usuarios

Para ejecutar la aplicación de manera local, será necesario seguir los pasos de instalación previamente mencionados, estos casos están planteados para su uso en Windows, para la descarga del repositorio e instalación de las librerías necesarias una vez hecho esto arrancamos como hemos visto previamente y podemos acceder a la aplicación en el puerto local generado por Flask **<http://127.0.0.1:5000/>**.

También existe un despliegue el cual se ha creado y configurado una cuenta en “Gitpod”, pero en la cual sin pago se restringe a un uso de 50h, por lo que en caso de estar disponible el link se encontrará en el “readme.md” del repositorio.

También se ha intentado un despliegue en un VPS personal, corriendo en los servidores de clouding.io

<http://185.166.215.219/>

E.3.Instalación

Para la instalación los pasos están indicados en el apartado anterior, al ser una instalación limpia la base de datos se creará desde cero y sin ningún registro,

Siguiendo todos los pasos de el apéndice anterior se instalarán todas las dependencias y librerías necesarias, para de esto modo poder iniciar el proyecto flask y comenzar a utilizar la aplicación

E.4. Manual del usuario

A continuación, se va a comenzar a detallar de forma sencilla y para cualquier usuario, las distintas funcionalidades que brinda la aplicación web.

Identificación de usuario

Una vez tengamos la aplicación disponible ya sea de manera local o si ha sido posible su acceso de manera remota. Nos encontraremos con que la aplicación nos requiera un inicio de sesión para acceder a cualquier página.

A login form with a light gray background. At the top, the word "Login" is centered in a bold, black, serif font. Below it are two rounded rectangular input fields: the first is labeled "Email" and the second is labeled "Contraseña" (Password), both in a small, gray, sans-serif font. Below these fields is a dark gray, rounded rectangular button with the text "Iniciar sesión" (Log in) in white, bold, sans-serif font. At the bottom, there is a blue, underlined link that says "Crear una cuenta" (Create an account).

Ilustración 0.1 Login usuarios

Como vemos lo primero que encontraremos será la vista de Login, si dispones de una cuenta para iniciar sesión en esta únicamente debemos 2 pasos.

- Introducir Email y contraseña correspondiente

- Presionar sobre “Iniciar Sesión”

Si los datos son correctos se iniciará la sesión del usuario y accederemos al home del usuario.

Registro de Usuario

En caso de no disponer de una cuenta de usuarios, se deberá presionar sobre “Crear una cuenta”, para ser redirigido a la página de registro.

El formulario de registro está contenido dentro de un recuadro con esquinas redondeadas y un fondo gris claro. En la parte superior, el título "Registro" está en un color marrón oscuro. A continuación, hay tres campos de entrada de texto con un fondo gris muy claro y un borde sutil: "Nombre de usuario", "email" y "Contraseña". Debajo de estos campos, hay un botón rectangular con esquinas redondeadas de color negro con el texto "Registrarse" en blanco. En la parte inferior del formulario, hay un enlace hipervinculado "Iniciar sesión" en un color azul.

Ilustración 0.2 Registro usuarios

Ahora al igual que para el inicio de sesión, si lo que queremos es el registro se deberán seguir de nuevo 2 sencillos pasos.

- Introducir nombre, email y contraseña.
- Presionar sobre “Registrarse”

Como condiciones se debe tener en cuenta que deben rellenarse todos los campos y además que el email no debe estar registrado previamente, y en caso de que ya exista algún usuario con este email la cuenta no podrá ser creada y se recibirá un aviso.

Ahora una vez que el usuario se haya registrado en el sistema, será redirigido de nuevo a la vista de Inicio de Sesión,

donde como ya hemos visto previamente este deberá introducir su email y contraseñas para acceder al sistema.

En el siguiente apartado se va a explorar el uso completo de la aplicación desde la vista de un usuario de esta, s

Apartado destinado a tratar lo referente al ámbito del usuario. Se tratarán principalmente temas relativos a los requisitos la instalación del proyecto y el manual de uso enfocado al usuario.

Home principal del usuario

Una vez que nos hayamos iniciado sesión lo primero que veremos será el home del usuario, este es el lugar central donde podremos de un vistazo todas nuestros archivos y los grupos a los que estamos vinculados.

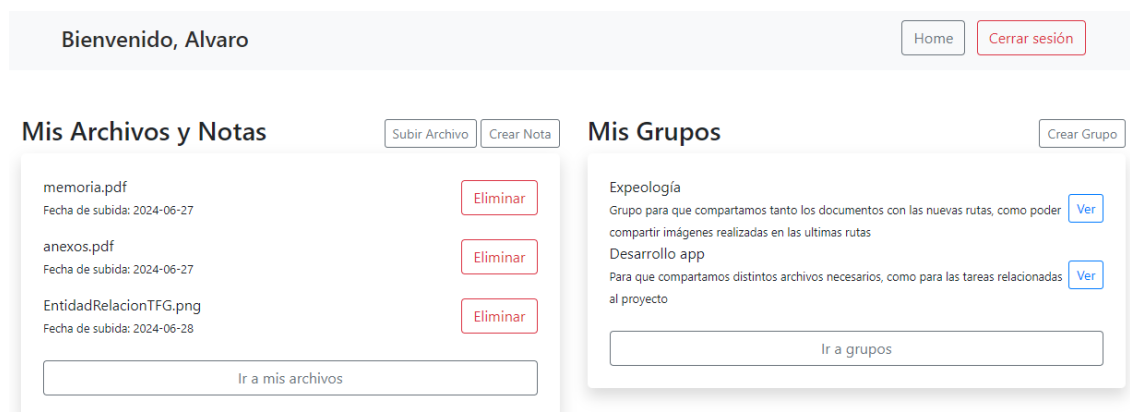


Ilustración 0.3 Home aplicación

Como vemos en esta vista encontramos bastante información, lo primero con lo que nos encontramos es la barra de navegación, la cual a partir de ahora será común para todas las páginas, desde esta siempre podremos volver al Home con el botón “Home” y también el botón “Cerrar sesión” con el cual cerraremos la sesión del usuario. A su vez también vemos el footer el cual será común para todas las páginas

A continuación, nos encontramos con la vista principal del Home, en el cual distinguimos 2 grandes tarjetas.

Tarjeta Izquierda, esta hace referencia a todos los archivos del usuario, en ella podremos ver una lista de nuestros archivos subidos y a lado de cada uno de ellos encontramos un botón con “Eliminar” mediante el cual podremos eliminar estos archivos de nuestro espacio de almacenamiento personal. También en esta tarjeta encontramos un botón “Ir a mis archivos” para poder acceder a la vista general de los archivos del usuario.

Tarjeta derecha, esta hace referencia a todos los grupos a los que pertenece el usuario, aquí podremos tanto ver el nombre del grupo como una pequeña descripción del mismo, también a la izquierda de cada uno de estos grupos, encontramos un botón para acceder directamente a la vista de estos. Podemos ver también como en la parte inferior existe el botón “Ir a grupos” el cual si pulsamos sobre el se nos redireccionara a la vista general de los grupos del usuario.

Vista de archivos del usuario

Como hemos dicho previamente desde el Home podemos acceder a la vista completa de nuestro archivo haciendo clic en “Mis archivos”. Esto nos dirigirá directamente a nuestra vista general de archivos.

Como vemos en esta se mantiene la misma barra superior, con la cual podremos volver al home o cerrar sesión en cualquier momento.

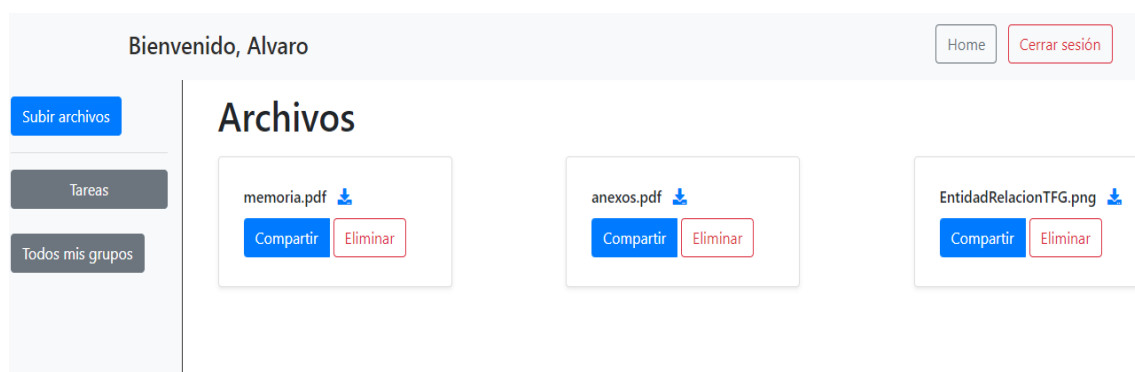


Ilustración 0.4 Vista archivos usuario

En nuestra vista archivos podemos apreciar que esta separa en 2 zonas claras, la barra lateral y el contenido principal.

En la **barra lateral**, encontramos principalmente 3 opciones.

Subir Archivo: Este botón despliega una pequeña zona donde se habilita un botón para la subida de archivos.

1. Pulsamos sobre el botón “Subir archivos” esto desplegara la zona que habilita la subida de archivos
2. Pulsamos sobre “Elegir archivo” y se selecciona un archivo de nuestro equipo

- Finalmente, con el archivo seleccionado pulsamos sobre “Subir”.

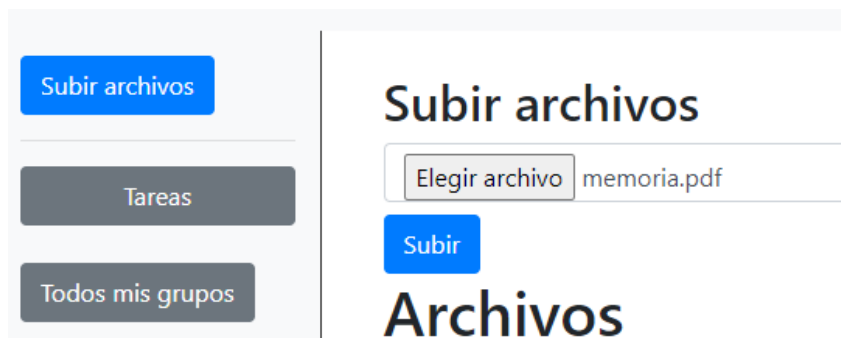


Ilustración 0.5 Sección subir archivos

Tareas: Este botón nos redirige a la vista de tareas del usuario.

Todos mis grupos: Este botón nos redirige a la vista de todos los grupos a los que pertenece el usuario.

A continuación, en el contenido principal de la vista, encontramos el listado mediante un estilo de tarjetas de los archivos de los cuales dispone el usuario en su almacenamiento personal.

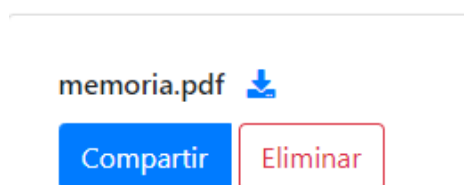


Ilustración 0.6 Sección archivo

Como vemos en la imagen, estas tarjetas que simbolizan los archivos, disponen de 3 botones.

Icono descarga: Este es el icono que se encuentra a la derecha del nombre de archivo, permite la descarga y recuperación de los archivos subidos.

Botón eliminar: Este botón al pulsarse elimina completamente el registro de ese archivo en el sistema. Eliminando tanto su referencia en la BD como el archivo en su ubicación dentro del sistema de ficheros.

Botón compartir: Este botón, permite al usuario compartir el archivo seleccionado con alguno de los grupos a los que pertenece. Cuando se pulsa este botón se abre un PopUp con un menú desplegable el cual permite seleccionar el grupo con el que se comparte el archivo.

Asignar grupo

Seleccionar Grupo

Expeología

Cancelar Guardar

Ilustración 0.7 Sección compartir archivo

1. Para compartir el archivo de forma correcta, seleccionamos el grupo con el que se quiere compartir.
2. Pulsamos sobre el botón “Guardar”

Vista de tareas del usuario

Como hemos visto anteriormente, desde la vista de archivos en la barra lateral podemos pulsar sobre “Tareas” para ser redirigidos directamente a esta vista de las tareas personales del usuario.

Como vemos en esta vista también se mantiene la misma barra superior, con la cual podremos volver al home o cerrar sesión en cualquier momento.

Bienvenido, Alvaro

Home Cerrar sesión

Crear Nueva Tarea

Archivos

Todos mis grupos

Lista de Tareas

Tarea 1

Subir las imágenes de los bocetos de diseño del 25/06 al grupo la App

Eliminar Compartir

Recordatorio

Descargar las rutas para el próximo fin de semana, del grupo de espeología

Eliminar Compartir

Ilustración 0.8 Vista tareas usuario

Lo primero que podemos observar es que es una vista muy similar a la vista de los archivos, ya que se intenta mantener la coherencia visual en las distintas listas. De nuevo vemos que la vista se encuentra separada en dos partes, la barra lateral y el contenido principal.

En la barra lateral encontramos tres opciones:

Crear nueva tarea: Este botón nos despliega un menú donde podemos rellenar la información para crear una nueva tarea.

Ilustración 0.9 Sección crear tarea

1. Pulsamos sobre “Crear nueva Tarea” para desplegar el menú de creación.
2. Rellenamos el título y descripción de la tarea
3. Finalmente pulsamos sobre “Agregar Tarea” y esta quedara registrada y creada en nuestro entorno de trabajo.

Archivos: Este botón nos redirige de nuevo la vista de los archivos del usuario.

Todos mis grupos: Este botón nos redirige a la vista de todos los grupos a los que pertenece el usuario.

A continuación, en el contenido principal de la vista, encontramos el listado de las distintas tareas, que pertenecen al usuario.

Tarea 1

Subir las imágenes de los bocetos de diseño del 25/06 al grupo la App



Eliminar

Compartir

Ilustración 0.10 Sección tarea usuario

Al igual que en los archivos, como vemos en la imagen, en las tareas, encontramos 3 botones.

Icono edición: Este icono con forma de lápiz, nos permite modificar tanto el nombre como la descripción de la tarea, al presionar sobre él, se despliegan las secciones para el editor de los campos.

Tarea 1

Subir las imágenes de los bocetos de diseño del 25/06 al grupo la App

**Eliminar****Compartir**

Tarea 1

Subir las imágenes de los bocetos de diseño del 25/06 al grupo la App

Guardar*Ilustración 0.11 Sección modificar tarea*

Botón eliminar: Al pulsar este botón se elimina totalmente el registro de la tarea en la base de datos.

Botón compartir: Al igual que con los archivos, este botón permite al usuario compartir las tareas con el grupo seleccionado. Una vez pulsamos sobre este nos mostrara un PopUp donde podemos seleccionar el grupo para compartir.

Ilustración 0.12 Sección compartir tarea

Vista general de grupos del usuario.

Como ya hemos visto previamente, tanto desde el “home” como desde las vistas de archivos y tareas podemos pulsar sobre el botón “Todos mis grupos” para ser redirigidos a la vista general de todos los grupos a los que pertenece el usuario.

De nuevo, como podemos observar en esta vista también se mantiene la misma barra superior, con la cual podremos volver al home o cerrar sesión en cualquier momento.

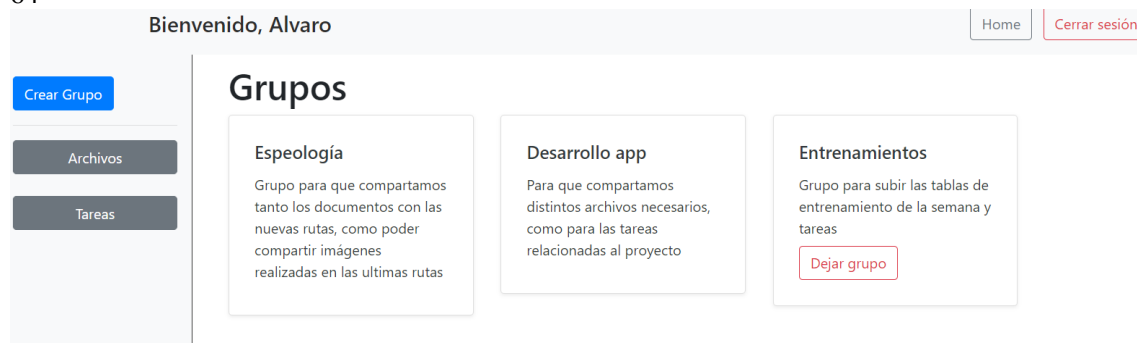


Ilustración 0.13 Vista grupos usuario

Lo primero que podemos observar es que es una vista muy tanto a la vista de archivos como de tareas, ya que se intenta mantener la coherencia visual en las distintas páginas de la aplicación

En la barra lateral encontramos tres opciones:

Crear nueva Grupo: Este botón nos despliega un menú donde podemos rellenar la información para crear un nuevo grupo.



Ilustración 0.14 Sección crear grupo

1. Pulsamos sobre “Crear nueva Grupo” para desplegar el menú de creación.
2. Rellenamos el nombre y descripción del grupo
3. Finalmente pulsamos sobre “Crear”.

Archivos: Este botón nos redirige de nuevo la vista de los archivos del usuario.

Tareas: Este botón nos redirige a la vista de tareas del usuario.

A continuación, en el contenido principal de la página encontramos los distintos grupos, a los que pertenece el usuario.

Grupos



Ilustración 0.15 Vista principal grupos usuario

Como podemos ver en los grupos los cuales el usuario no es propietario encontramos un botón “Dejar grupo” el cual permite al usuario abandonar los grupos en los cuales sea un invitado.

Finalmente, el usuario puede acceder a cualquiera de los grupos, pinchando sobre la tarjeta correspondiente, y será redirigido a la página del grupo del grupo correspondiente.

Vista principal de un grupo.

A la hora de acceder a un grupo, puedo que seamos el propietario de este, en caso de que el grupo haya sido creado por nosotros, o que seamos invitados es decir el grupo pertenezca a otros usuarios y nos haya invitado previamente.

Primero vamos a ver la vista del usuario como invitado.

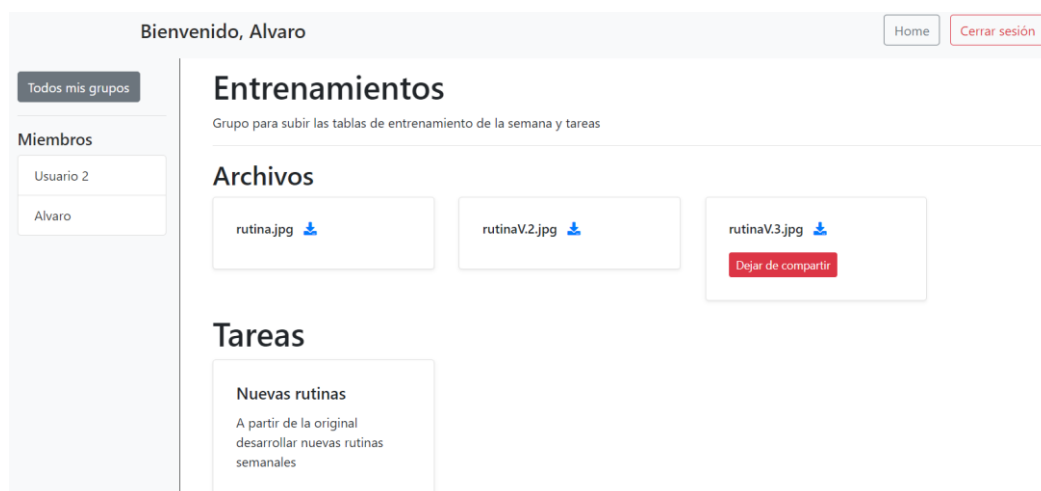


Ilustración 0.16 Vista de un grupo

Como podemos observar, de nuevo en esta vista mantenemos la coherencia visual, observando una barra lateral y un contenido principal.

En la barra lateral, encontramos la opción de “Todos mis grupos” un botón que de nuevo nos redirigirá a la página con todos los grupos a los que pertenecemos. A continuación, observamos una lista con los miembros actuales del grupo.

En el contenido principal, encontramos primeramente el Nombre y descripción del grupo, seguidamente vemos todos los archivos que han sido compartidos en el grupo, y al igual que en nuestro espacio personal, podemos pulsar sobre el icono de descarga para descargar estos archivos en nuestro equipo, y finalmente veremos las tareas del grupo.

Tanto en los archivos como en tareas de los que seamos propietarios, podemos observar el botón “Dejar de compartir”, el cual podremos utilizar para dejar de compartir un archivo con el grupo en el que nos encontremos.

Ahora vamos a observar cómo se ve un grupo del cual somos propietarios.

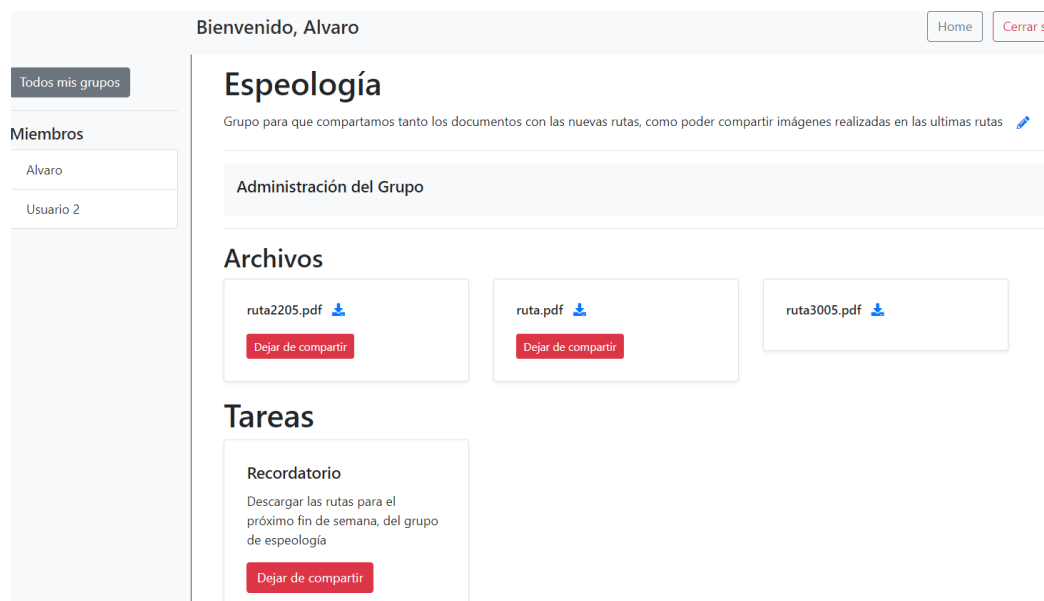



Ilustración 0.17 Vista administrador de un grupo

Como podemos observar la vista es muy similar a la de los grupos en los cuales somos invitados, pero principalmente podemos observar unos cambios para los administradores.

Si observamos el nombre y descripción veremos un icono de edición sobre el que podemos pulsar para modificar los datos del grupo.

Espeología

Grupo para que compartamos tanto los documentos con las nuevas rutas, como poder compartir imágenes realizadas en las ultimas rutas 

Espeología

Grupo para que compartamos tanto los documentos con las nuevas rutas, como poder compartir imágenes realizadas en las ultimas rutas

Guardar

Ilustración 0.18 Sección editar grupo

1. Pulsamos sobre el icono de edición para desplegar el menú.
2. Modificamos los datos necesarios.
3. Finalmente pulsamos sobre “Guardar”.

Posteriormente a diferencia de los usuarios invitados, también podemos la sección “Administración del grupo” sobre

la cual podemos pinchar para desplegar las distintas

opciones de administración.

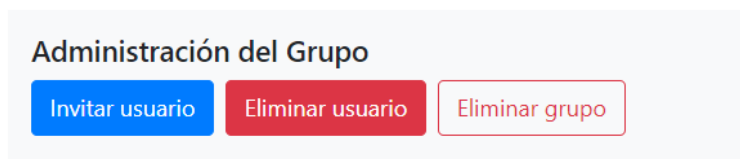


Ilustración 0.19 Administración grupo

Al desplegar esta sección podemos observar tres botones, los cuales cada uno de estos nos permite realizar una acción.

Invitar usuario: Si pulsamos sobre este botón desplegaremos un pequeño menú en el cual debemos introducir el correo del usuario que deseamos invitar.

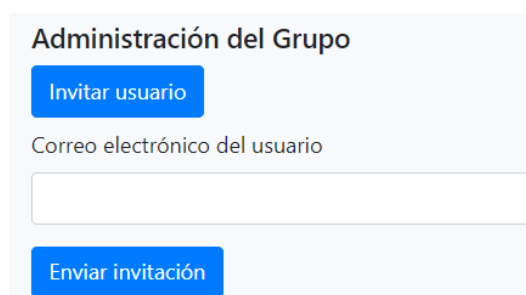


Ilustración 0.20 Sección invitar usuario

Si el usuario existe, este será agregado al grupo y podrá acceder a todos los archivos y tareas del grupo, así como comenzar a compartir los suyos.

Eliminar usuario: Si pulsamos sobre este botón desplegaremos un pequeño menú desplegable en el cual podremos seleccionar entre los miembros del grupo, al usuario que deseamos eliminar.

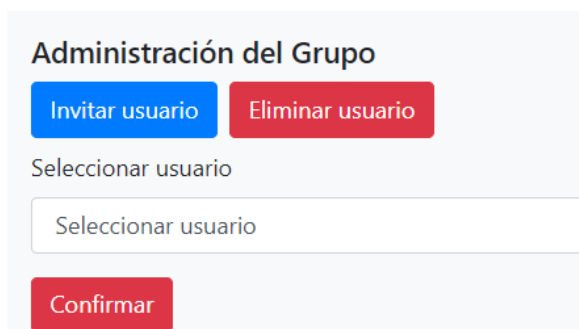


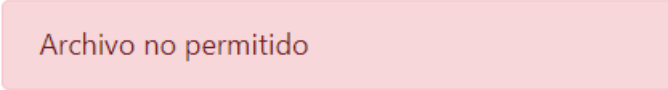
Ilustración 0.21 Sección eliminar usuario

Eliminar grupo: Pulsando este botón eliminaremos completamente el grupo de la base de datos.

Mensaje de estado

Durante la navegación cuando realicemos distintas acciones, ya sea compartir un archivo, invitar un usuario, recibir un error etc.... Vamos a recibir distintos mensajes de control, si la operación ha sido exitosa, si ha ocurrido un error conocido, o un error inesperado, esto no ocurre en todas las operaciones, puesto que en operación exitosas que sean claramente visibles, se considera irrelevante.

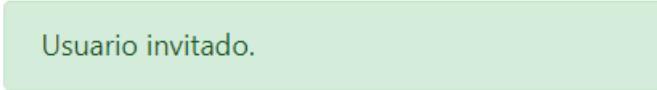
Vamos a ver a continuación unos ejemplos.

A horizontal rectangular box with a light red background and rounded corners. The text "Archivo no permitido" is centered in a dark red font.

Archivo no permitido

Ilustración 0.22 Mensaje error

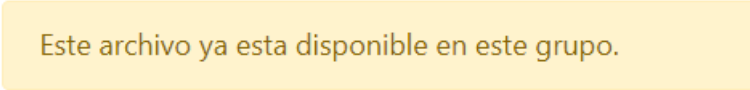
Como vemos un mensaje de color rojo, simboliza que ha ocurrido un error en este caso que el tipo de archivo no está permitido.

A horizontal rectangular box with a light green background and rounded corners. The text "Usuario invitado." is centered in a dark green font.

Usuario invitado.

Ilustración 0.23 Mensaje de éxito

Ahora observamos un mensaje de color verde, lo que simboliza que la operación se ha realizado con éxito en este caso la invitación de un usuario.

A horizontal rectangular box with a light yellow background and rounded corners. The text "Este archivo ya esta disponible en este grupo." is centered in a dark yellow font.

Este archivo ya esta disponible en este grupo.

Ilustración 0.24 Mensaje advertencia

Finalmente observamos un mensaje de color amarillo, el cual quiere mostrar un aviso, pero no realmente un error, por ejemplo en este caso nos indica que el archivo ya ha sido compartido con el grupo, por lo cual no ha sido compartido de nuevo.

Apéndice F

Anexo de sostenibilización curricular

F.1. Introducción

En este apartado se va a abordar la reflexión sobre como el desarrollo de este trabajo contribuye a la sostenibilidad en línea con los Objetivos de Desarrollo Sostenible (ODS) [11] establecidos por la ONU. Se va a ver tanto su implicación ética, impacto medioambiental y los aspectos socioculturales del trabajo.

Este proyecto se centra principalmente para su uso en el ámbito educativo o laboral. La aplicación tiene como objetivo la gestión de archivos y tareas para grupos de trabajos, promoviendo un entorno más organizado y eficiente, buscando además ser una herramienta accesible y gratuita. El objetivo es demostrar como esta aplicación de trabajo colaborativo puede ser una herramienta valiosa tanto a nivel técnico como a con una perspectiva sostenible.

F.2. Implicación ética

El desarrollo de esta aplicación web tiene fuertes implicaciones éticas, ya que facilita la colaboración e intenta fomentar los grupos de trabajo. Al buscar una herramienta gratuita y accesible para la gestión de archivos y tareas en grupos, se enfoca en promover la igualdad de oportunidad en los ámbitos laborales o educativos. El uso de software libre garantiza que independientemente sus recursos, cualquier

persona pueda acceder y disfrutar la aplicación.

Además, en el trabajo se intenta respetar la privacidad y la seguridad de los datos, siguiendo con el ODS 4: “Educación de calidad”, que busca una educación inclusiva, equitativa y de calidad, para promover el aprendizaje para todos.

F.3. Implicación sociocultural

Desde un punto de vista sociocultural, el desarrollo de este proyecto fomenta la colaboración y trabajo en equipo, y permite tanto en el ámbito laboral como escolar compartir recursos, colaborar y trabajar juntos independientemente de la ubicación geográfica. Además, con la idea de ser una aplicación de código abierto y gratuito, garantiza que todas las personas independientemente de la situación económica puedan tener acceso y beneficiarse del uso de esta.

Esto busca estar en línea con el ODS 10: “Reducción de las desigualdades”, el cual busca reducir las desigualdades entre los distintos países. Además, en el proyecto se busca tener un código libre y como se ha dicho busca ser escalable y de fácil mantenimiento, para que así pueda ser mejorado por la comunidad, promoviendo también de este modo la inclusión, y participación de diferentes personas en el desarrollo de este.

F.4. Implicación medioambiental

En el trabajo se busca un impacto medioambiental mínimo, debido a la naturaleza digital de la aplicación desarrollada. Como se promueve el uso de archivos digitales en lugar de documentos físicos en papel, se está contribuyendo a la reducción del uso de este, y por tanto a la conservación de los recursos naturales, además el uso de activos digitales también reduce la necesidad de transporte y almacenamiento, por lo que así también se reducen las emisiones de carbono.

Vemos, así como se promueve el uso eficiente de los recursos y minimiza el impacto medioambiental lo cual está en línea con el ODS 12: “Producción y consumo responsable”, el cual busca una gestión sostenible y uso eficiente de los recursos naturales.

En líneas futuras, podría abordarse el uso de energías renovables o la optimización del consumo energético en el caso de uso de servidores centrales para maximizar aún más el impacto

medioambiental del proyecto.

F.5. Conclusión

En conclusión, se puede apreciar que el proyecto no aporta solo beneficios técnicos, sino también se encuentra alineado con varios ODS, ya que prueba la sostenibilidad en las distintas perspectivas, tanto en la implicación ética, como con un bajo impacto medioambiental y en los aspectos socioculturales.

Al integrar prácticas sostenibles y responsables en el desarrollo de los proyectos, se está contribuyendo a un futuro más respetuoso y equitativo. En este apéndice se ha intentado demostrar como sostenibilidad y la tecnología pueden ir de la mano además de crear un impacto positivo en la sociedad.

Bibliografía

- [1] Proyectos Ágiles. "Desarrollo iterativo incremental". Disponible en: <https://proyectosagiles.org/desarrollo-iterativo-incremental/>
- [2] Holded. "Amortización de equipos informáticos". Disponible en: <https://www.holded.com/es/blog/amortizacion-de-equipos-informaticos>
- [3] Visual Studio. "Pricing". Disponible en: <https://visualstudio.microsoft.com/es/vs/pricing/?tab=Empresa>
- [4] Wikipedia. "Licencia BSD". Disponible en: https://es.wikipedia.org/wiki/Licencia_BSD
- [5] Wikipedia. "Licencia MIT". Disponible en: https://es.wikipedia.org/wiki/Licencia_MIT
- [6] Wikipedia. "Dominio público". Disponible en: https://es.wikipedia.org/wiki/Dominio_p%C3%ABlico
- [7] Dev.to. "Object-Relational Mapping (ORM): Explicación completa, definición, uso, estructura y práctica". Disponible en: <https://dev.to/ingeniela/object-relational-mapping-orm-explicacion-completa-definicion-uso-estructura-y-practica-4j56>
- [8] Wikipedia. "Modelo–vista–controlador". Disponible en: <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>
- [9] Python. "Downloads". Disponible en: <https://www.python.org/downloads/>
- [10] Git. "Downloads". Disponible en: <https://git-scm.com/downloads>
- [11] Pacto Mundial "ODS". Disponible en: <https://www.pactomundial.org/que-puedes-hacer-tu/ods/>