**Informe Documentación de la aplicación**

**Unidad 2: Desarrollo de la aplicación**

Asignatura: Taller de Desarrollo de Aplicaciones

Sección: TIHI31/D-B50-N3-P12-C3/D

Nombre del docente: Carolina Ehrmantraut Caballero

Nombre de los integrantes del grupo:

Juan Alonso Astudillo Murua, Bruno Robinson Morales, Sebastián Maulen Santander

Fecha de entrega: Martes 11 de Julio de 2023

# Para la demostración de la aplicación.

## Presentación del proyecto.

Este proyecto es una aplicación de gestor de tareas, la cual permite al usuario administrar tareas y subdividirlas por categorías como también incluye la opción de administrar un grupo e ingresar miembros para realizar tareas en conjunto como grupo, como última cosa la aplicación está hecha a partir del framework Django y el lenguaje Python.

## 1. Módulos de la aplicación, a partir de los procesos principales del negocio.

En este ítem deben demostrar con la aplicación en ejecución los módulos desarrollados para los procesos principales del CRUD propuesto en su proyecto. Además, mostrar la navegación a través de la aplicación para lograr la ejecución de un proceso. Por ejemplo, módulo de ingreso de datos, módulo de consultas e informes, etc.

Módulo “DjangoCrud”

Este módulo configura a través del módulo urls.py las direcciones url de la página, estas direcciones url funcionan con las vistas o funciones que están definidas en otro módulo que es “tasks.views”, también este módulo tiene el settings.py que permite vincular la app con la base de datos y definir el lenguaje de la aplicación, el nombre de la aplicación y los directorios que trabaja el proyecto.

Módulo de “tasks” -> “tasks.views”

Este contiene las siguiente vistas o funciones:

Este módulo de Django, proporciona vistas y funciones relacionadas con la autenticación de usuarios, la creación y gestión de tareas, la organización de tareas en categorías y la gestión de grupos de usuarios y tareas asociadas. Aquí está una descripción de lo que hace cada una de las partes del código:

- Las primeras líneas del código importan varias funciones, clases y modelos de Django necesarios para el funcionamiento del sistema.

- La función `home(request)` renderiza la plantilla 'home.html'.

- La función `signup(request)` maneja la creación de nuevos usuarios. Si el método de solicitud es GET, se muestra un formulario de registro. Si el método de solicitud es POST, se verifica la información enviada por el usuario, se crea un nuevo usuario y se inicia sesión.

- La función `loguserout(request)` maneja el cierre de sesión del usuario actual y redirige a la página de inicio.

- La función `loguserin(request)` maneja el inicio de sesión de los usuarios. Si el método de solicitud es GET, se muestra un formulario de inicio de sesión. Si el método de solicitud es POST, se verifica la información de inicio de sesión y se inicia sesión.

- La función `createTask(request)` maneja la creación de nuevas tareas por parte de los usuarios. Si el método de solicitud es GET, se muestra un formulario para crear una tarea. Si el método de solicitud es POST, se guarda la nueva tarea en la base de datos.

- La función `task(request)` muestra todas las tareas activas del usuario actual.

- La función `taskend(request)` muestra todas las tareas completadas por el usuario actual.

- La función `task\_detail(request, task\_id)` muestra los detalles de una tarea específica y permite al usuario actualizarla.

- La función `taskComplete(request, task\_id)` marca una tarea como completada.

- La función `taskDelete(request, task\_id)` elimina una tarea.

- La función `createCategory(request)` maneja la creación de nuevas categorías para organizar tareas. Si el método de solicitud es GET, se muestra un formulario para crear una categoría. Si el método de solicitud es POST, se guarda la nueva categoría en la base de datos.

- La función `category(request)` muestra todas las categorías y las tareas asociadas a cada categoría del usuario actual.

- La función `deleteCategory(request, category\_id)` elimina una categoría y todas las tareas asociadas a ella.

- La función `updateCategory(request, category)` renderiza un formulario para actualizar una categoría existente.

- Las funciones restantes (`showGroup`, `deleteGroup`, `modifyGroup`, `createGroup`, `createGroupTask`, `task\_group\_detail`) manejan la creación, modificación y eliminación de grupos de usuarios y tareas asociadas a esos grupos.

Módulo de “tasks” -> “tasks.forms”

Este módulo provee un conjunto o grupo de clases para construir y manejar formularios HTML. Este simplifica la validación de formularios, el envío o traspaso de datos, y el renderizar los campos de los formularios.

Módulo de “tasks” -> “tasks.templates”

Este módulo es responsable para el motor de plantillas en Django. Este te permite definir las plantillas HTML pueden ser dinámicamente renderizadas con la información que vienen de las vistas. El motor de plantillas soporta variables, filtros, etiquetas y herencia de plantillas.

## 2. Modelo usado en la creación de la aplicación.

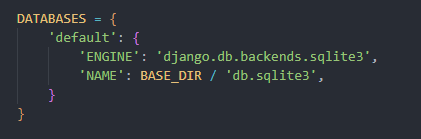
Entre los diversos patrones que hemos utilizado debido al uso del framework Django, se encuentran los siguientes:

1. **MVC**: Al utilizar el modelo-vista-controlador, ya que de esta forma representamos los datos y la lógica empresarial. Entre sus ventajas tenemos: *Separación clara de la lógica empresarial y la presentación, lo que facilita el mantenimiento y la evolución del código*.
2. **ORM**: En vista de mapear objetos Python a tablas de base de datos y realizar operaciones CRUD en la base de datos. Entre sus ventajas tenemos: *Abstracción y portabilidad de la base de datos.*
3. **URL Dispatcher**: Es indispensable el uso del sistema de enrutamiento basado en patrones de URL para asociar a las respectivas vistas correspondientes de nuestra app web. Entre sus ventajas tenemos: *Separación clara de la lógica de enrutamiento y las vistas, además de flexibilidad en el enrutamiento.*
4. **Patrón Middleware**: Su uso es elemental para proporcionar una capa de procesamiento intermedia entre las solicitudes entrantes y las respuestas. Entre sus ventajas tenemos: *Permitir agregar funcionalidad adicional a nivel de solicitud y respuesta y facilitar la gestión de aspectos comunes de las solicitudes y respuestas.*
5. **Patrón Formulario**: La captura de datos por medio de formulario es esencial para extraer funcionalidades. En base a esto se incluye la validación de datos, generación automática de formularios de HTML y manipulación de datos enviados por usuarios. *Entre sus ventajas tenemos: Validación de datos y generación automática de formularios HTML*.

## 3. Conexión a la base datos y efectuar las operaciones CRUD.

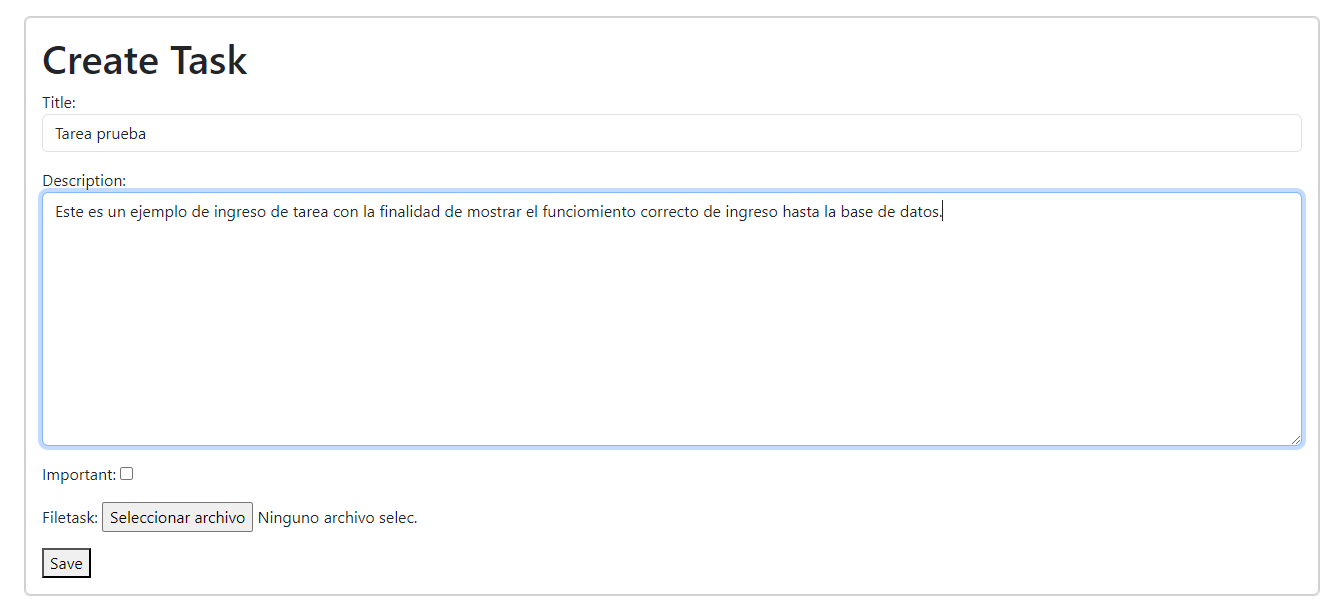
Para iniciar la demostración de la aplicación web en el funcionamiento correcto de la base de datos, por medio de operaciones CRUD; se tiene lo siguiente:

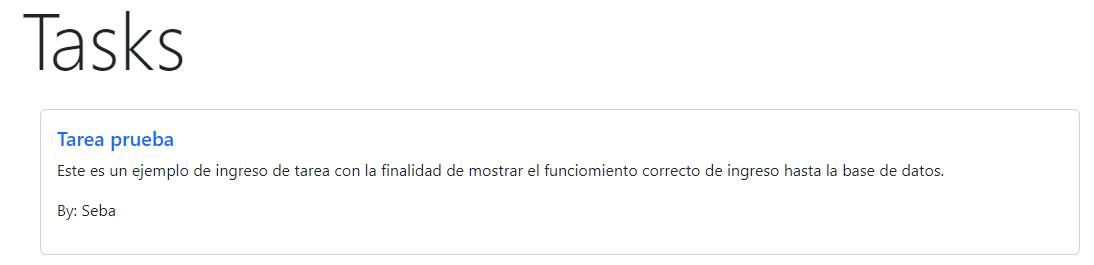
*i. Conexión a la base, por medio del archivo de nuestro proyecto setting.py:*



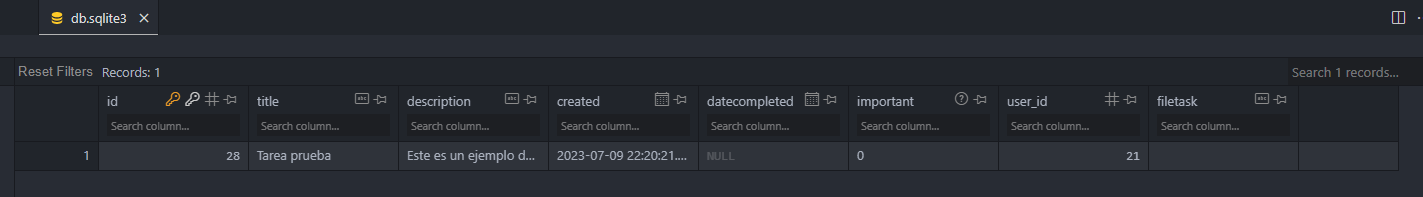
Django utiliza SQLite3 como base de datos predeterminada por varias razones: Ligereza y simplicidad, fácil configuración, desarrollo rápido y buen rendimiento.

*ii. Creando tarea desde interfaz de usuario:*





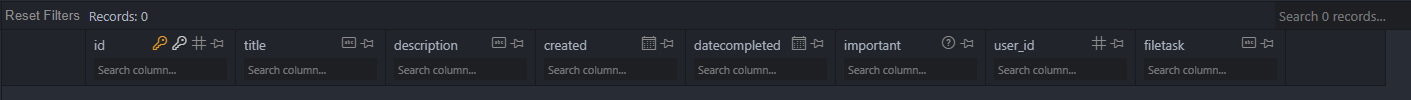
Tarea ingresada correctamente.

*iii. Verificando en la base de datos el correcto ingreso de la nueva tarea:*

# *iv. Eliminando desde interfaz de usuario la tarea ingresada anteriormente:*



*v. Verificando registro en base de datos se tiene:*



En base a los pasos antes señalados, podemos comprobar la manipulación de los datos en lo que respecta a un CRUD.

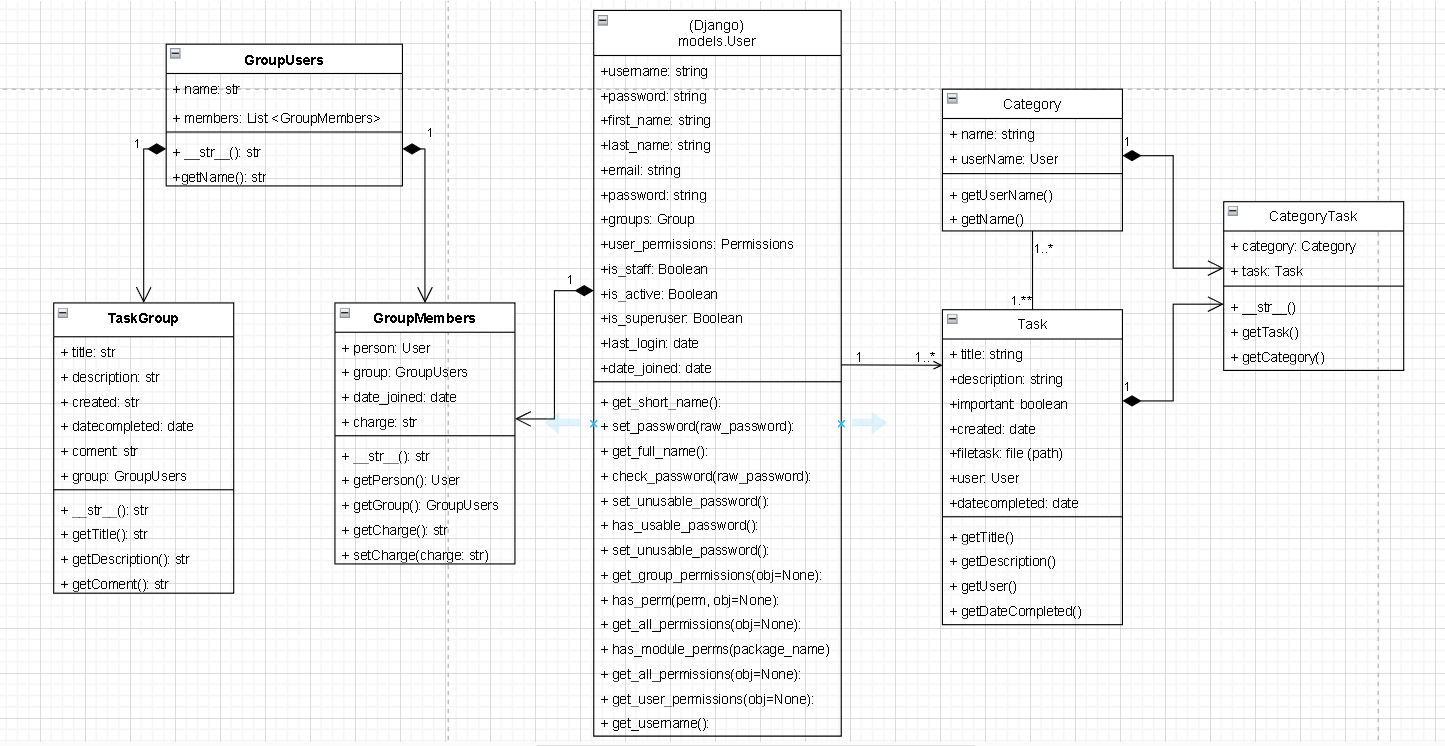
# Para la documentación del proyecto.

## 1. Documentación de las clases del proyecto.

Diagrama de clases.

Explicacion: este es el modulo de models.py que utilizamos en la app de tareas (Tasks), en general django provee un models.user por defecto y fue el que utilizamos pero por supuesto no supe cómo relacionarlo, ya que una parte de las clases de nuestro models.py depende o requiere del models.user de Django. Así que las clases de este módulo:

GroupUsers, GroupMembers, TaskGroup,CategoryTest ,CategoryTestTask => dependen de la clase User de django.models.User.



## 2. Documentación de pruebas.

Llenar la siguiente plantilla para **dos casos de prueba** que sean relevantes.

| **<Clasificar tareas en categoría>** | **<Código del Caso Prueba>** |
| --- | --- |
| **Descripción:**  El objetivo es crear una categoría e ingresar a la categoría creada una tarea. | |
| **Prerrequisitos**  El usuario debe tener una tarea creada previamente para crear una categoría y clasificarla. | |
| **Pasos:**  Paso 1: Tener una tarea creada.    Paso 2: Dirigirnos al apartado    Paso 3: Creando una nueva categoría.    Paso 4: Una vez creada la categoría, seleccionamos la categoría creada y la tarea que deseamos añadir a dicha categoría y la agregamos.    Paso 5: Verificamos si la categoría se ha creado correctamente en la base de datos. | |
| **Resultado esperado:**  Se espera que la categoría pueda ser creada y que las tareas puedan ser ingresadas correctamente en dicha categoría. | |
| **Resultado obtenido:**  Al momento de seleccionar la categoría creada previamente, esta no aparece para ser utilizada al momento de añadir una tarea. | |
| **Acciones correctivas:**  No se presentaron problemas previos debido a una correcta estructura del código. | |

| **Inserción de datos** | **InsD01** |
| --- | --- |
| **Descripción:**  Se desea probar el sistema de creación de tareas, verificar si todos los datos se insertan correctamente y revisar si en las secciones que muestran datos de tareas los datos se mantienen coherentes. | |
| **Prerrequisitos**  El usuario debería iniciar sesión para crear una tarea. | |
| **Paso 1:**  Abrir la aplicación e iniciar sesión  .  **Paso 2:**  Hacer click en “create task”.  **Paso 3:**  Introducir el nombre, descripción e importancia de la tarea, para finalmente hacer click en “Save”.    **Paso 4:**  Revisar en la página principal si las tareas tienen datos coherentes.    **Paso 5:**  Revisar si los datos de las tareas son coherentes en la sección categorías.    **Paso 6:**  **(verificar tareas en grupo)(Debe pertenecer a un grupo o ser líder de este)**  Al crear un tarea en grupo, revisar si los datos ingresados en tareas grupales son correctamente introducidos.    **Paso 7:**  Revisar si los datos de tareas grupales son coherentes en los detalles del grupo  **Paso 8:**  Revisar si el sistema permite ingresar datos nulos    **Paso 9:**  Revisar si los datos se ingresaron correctamente a la Base de datos.    (Los datos null son la fecha de completación). | |
| **Resultado esperado:**  Se esperaba que los datos fueran correctamente introducidos en la aplicación, y que mantuvieran coherencia al verlos en otras secciones. | |
| **Resultado obtenido:**  El resultado final fue el esperado, el sistema ingresa y muestra los datos correctamente, además las validaciones del navegador no permite dejar campos vacíos.. | |
| **Acciones correctivas:**  No fue necesario tomar acciones, ya que el sistema no presenta fallos. | |

## 3. Manual de instalación (si corresponde).

Pasos a seguir para poder usar la aplicación:

1.Instalar python en su web oficial

2.Instalar Django

2.1 Para instalar django debes ingresar a la cmd y escribir pip install django

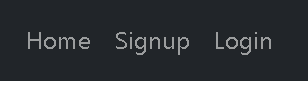
3.Descargar y descomprimir “my1stapp-main”

4.Para abrir el programa, hacer click en gestor 1.0

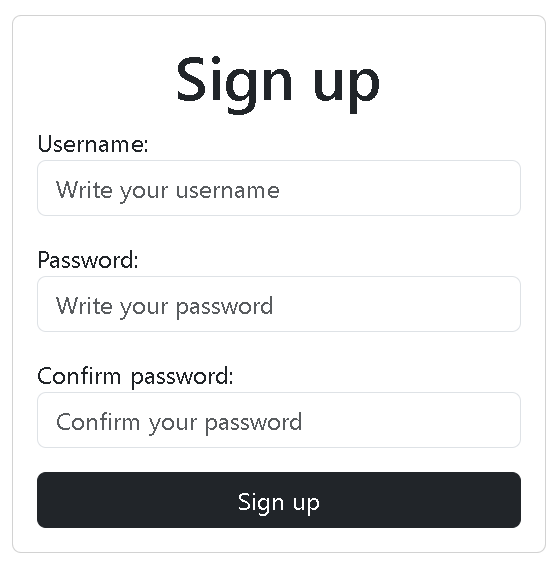
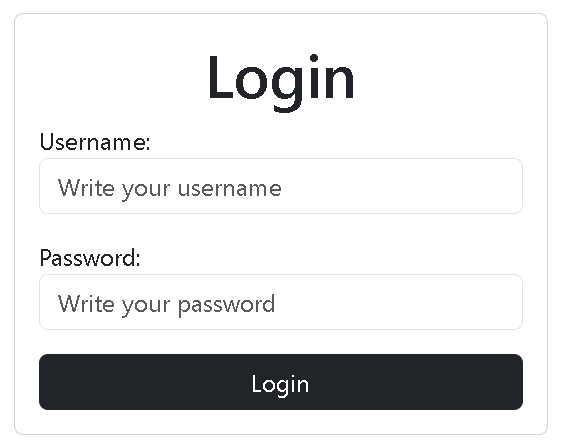
## 4. Manual de usuario.

1. Ejecutar Gestor 1.0

2. Login es para iniciar sesión en caso de tener una cuenta en la aplicación, Register es para crear una cuenta nueva.

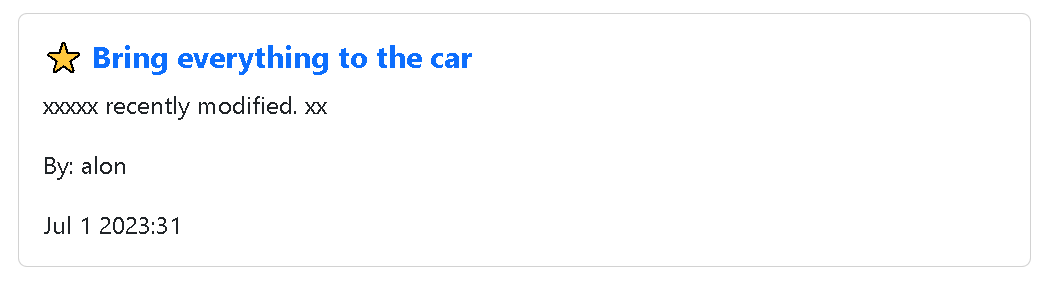


Vista del login y del register:

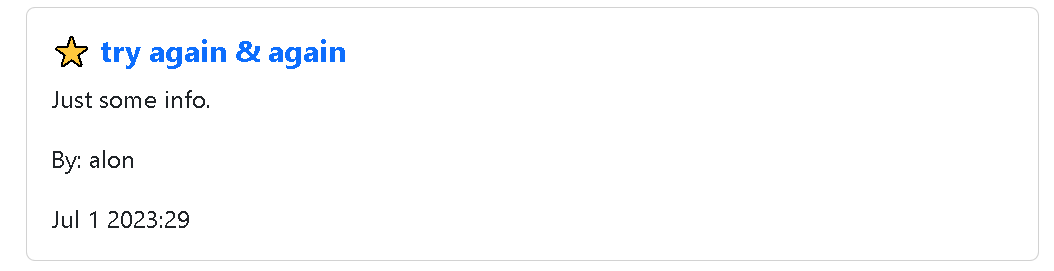


3.Tasks:

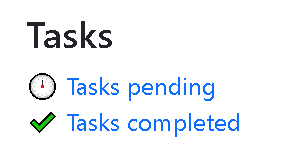
⭐ => Si una tarea contiene en su título este icono, significa que la tarea es importante.



Jul 1 2023:29 => Si la tarea indica una fecha es porque esta ya se completo.

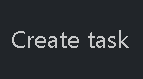


✔️Tasks completed / 🕛 Tasks pending => Estos enlaces indican al usuario para seleccionar entre las tareas ya completadas o las tareas pendientes.



📝 Create Task => Este enlace permite al usuario crear una nueva tarea.

Opción 1



Opción 2



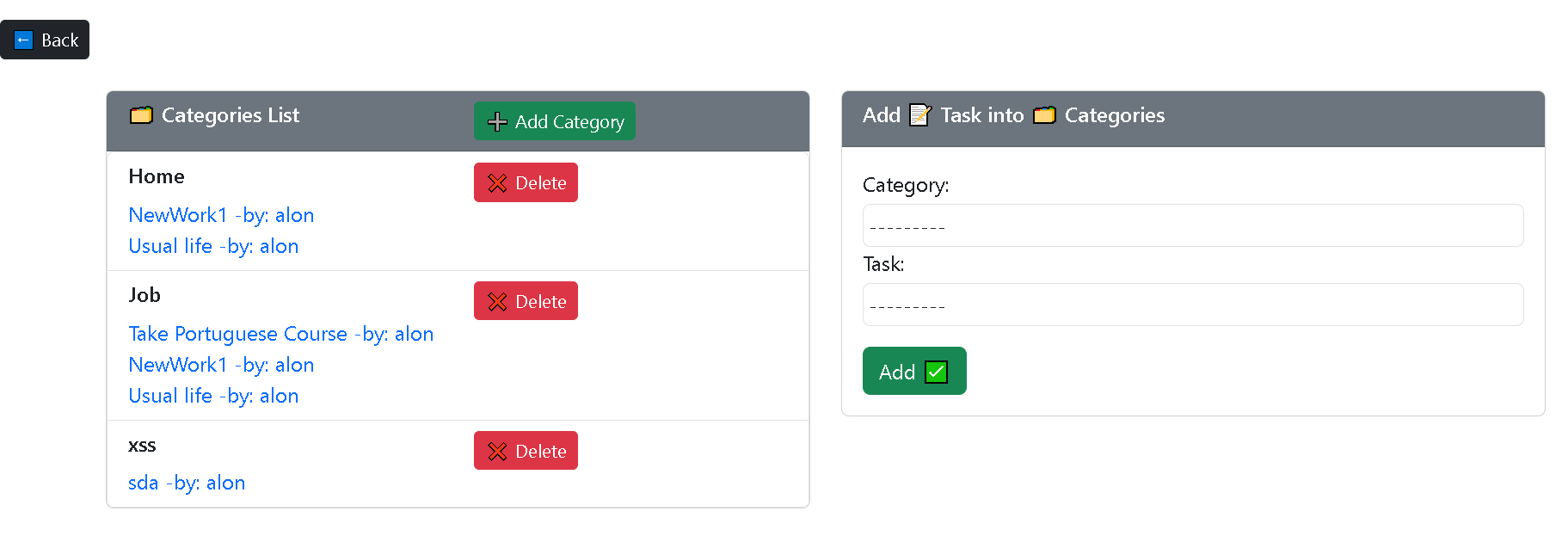
4.-Categories:

🗂️ Your Categories=> Enlace que te lleva a las categorías.

Enlace:



Página:



🗂️Categories List => Muestra las categorías del usuario y la tareas que contiene.

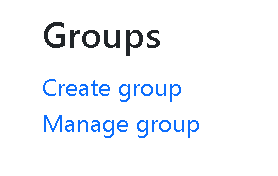
Posee la opciones de Add Category y Delete:

➕Add Category => Es una opción que permite al usuario añadir una categoría.

❌Delete => Es una opción que permite al usuario eliminar una categoría.

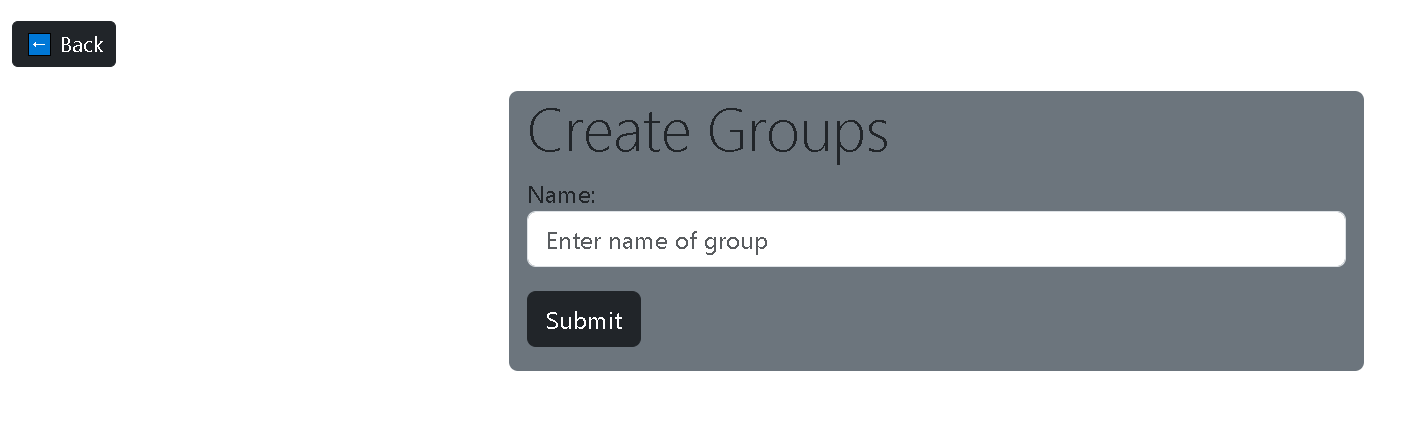
Add 📝 Task into 🗂️ Categories => Esta sección le permite al permite al usuario seleccionar una categoría y una tarea, esto hará que la tarea seleccionada sea guardada en la categoría seleccionada (Por lo cual debe haber creado una tarea y una categoría para poder usar esta sección).

5.Groups:

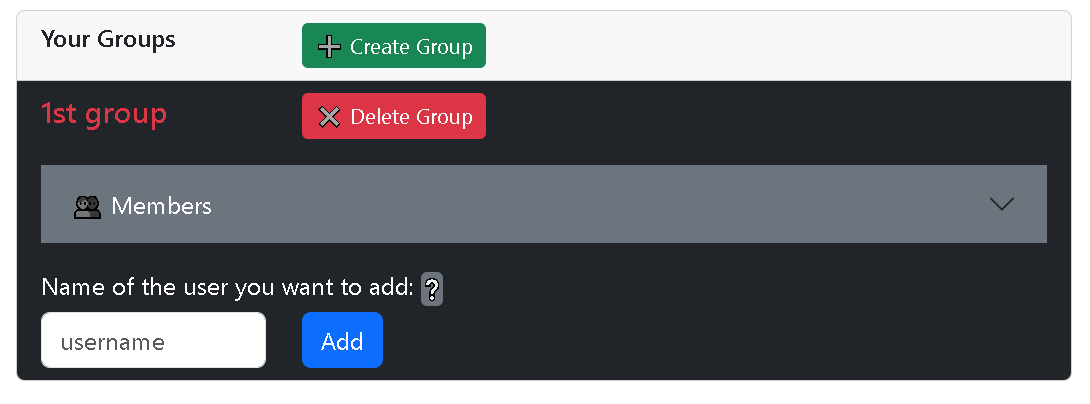


Create group/ Manage group => Estas dos opciones te permiten crear o administrar un grupo.

Create Group te lleva a la siguiente página donde puedes ingresar el nombre del grupo.



Manage group te lleva a la siguiente página donde te aparecen los grupos a los que perteneces.

****

Si eres líder puedes eliminar el grupo con la opción:



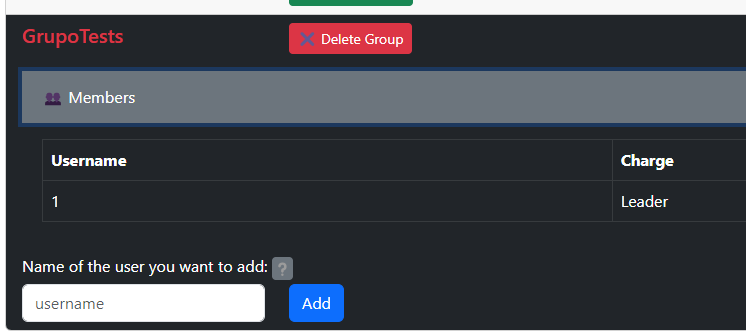
Y si eres miembro puedes salir del grupo con:



La sección de “name of the user you want to add” le permite al usuario ingresar el nombre de un usuario que quiere agregar (esto con la condición que el nombre de aquel usuario exista).

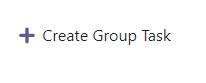
6.Group Tasks:

Para crear tareas grupales, debes estar en “Manage Groups” y hacer click en el nombre del grupo donde quieres crear la tarea

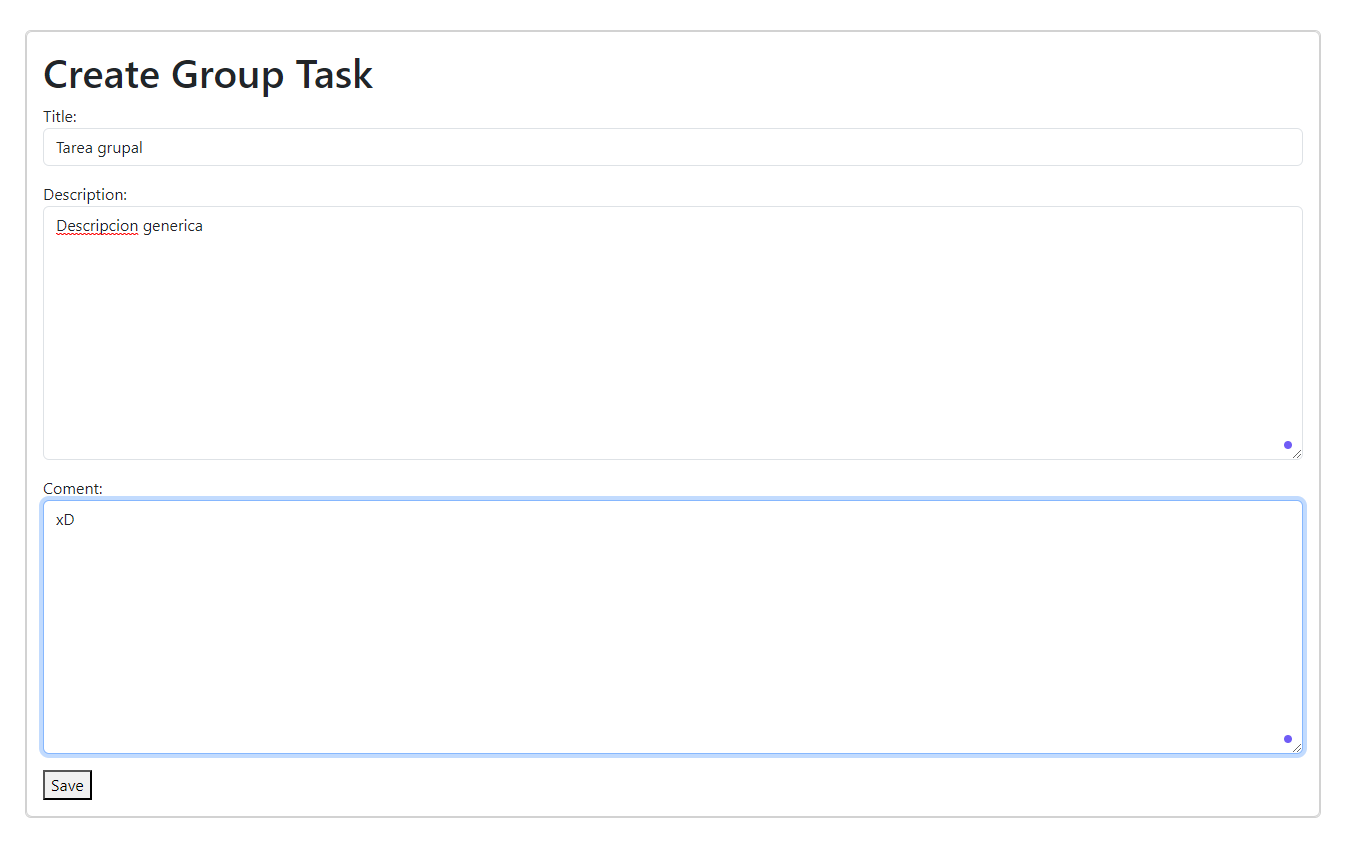


EJ:Hacer click en “GrupoTests”

Una vez en esta pagina, podras ver las tareas grupales, si es que han sido creadas, para crear una nueva tarea debes hacer click en el siguiente boton:



Introducir los datos y luego hacer click en “Save”



## 5. Términos y condiciones de uso.

Términos y Condiciones de Uso para la instalación del programa "Django Task app"

Estos Términos y Condiciones de Uso (en adelante, "Términos") establecen los términos legales y las condiciones que rigen el uso de la aplicación de software "Django Task app" (en adelante, la "Aplicación") desarrollada por “Anonymous Group” .Al instalar y utilizar la Aplicación, usted ("Usuario" o "Usted") acepta regirse por estos Términos. Si no está de acuerdo con estos Términos, no instale ni utilice la Aplicación.

1. Licencia de Uso:

a. Concesión de Licencia: Se le otorga una licencia limitada, no exclusiva, intransferible y revocable para instalar y utilizar la Aplicación en un dispositivo compatible, de acuerdo con estos Términos.

b. Restricciones: Usted se compromete a no copiar, modificar, distribuir, vender, sublicenciar ni realizar ingeniería inversa sobre la Aplicación, ni a permitir que terceros lo hagan.

2. Privacidad:

a. Recopilación de Datos: Podemos recopilar ciertos datos personales y no personales relacionados con su uso de la Aplicación, de acuerdo con nuestra Política de Privacidad. Al utilizar la Aplicación, Usted acepta la recopilación y el uso de sus datos de acuerdo con dicha Política.

b. Seguridad: Tomamos medidas razonables para proteger su información personal, pero no podemos garantizar la seguridad completa de los datos transmitidos o almacenados. Usted utiliza la Aplicación bajo su propio riesgo.

3. Responsabilidad:

a. Uso Responsable: Usted es el único responsable de su uso de la Aplicación y acepta utilizarla de manera legal y ética. No debe utilizar la Aplicación para actividades ilegales, maliciosas o que violen los derechos de terceros.

b. Exclusión de Responsabilidad: La Aplicación se proporciona "tal cual" y no ofrecemos garantías de ningún tipo, ya sea expresas o implícitas, incluidas, entre otras, las garantías de comerciabilidad, idoneidad para un propósito particular y no infracción. No seremos responsables de ningún daño directo, indirecto, incidental, especial o consecuente derivado de su uso de la Aplicación.

4. Propiedad Intelectual:

a. Derechos de Propiedad: La Aplicación y todo el contenido relacionado, incluidos los derechos de autor, marcas comerciales y otros derechos de propiedad intelectual, son propiedad exclusiva del Desarrollador.

b. Uso Autorizado: Usted acepta utilizar la Aplicación únicamente con fines personales y no comerciales. No puede modificar, copiar, distribuir, transmitir, mostrar, realizar, reproducir, publicar, licenciar, crear trabajos derivados, transferir ni vender ningún contenido de la Aplicación.

5. Modificaciones:

Nos reservamos el derecho de modificar, actualizar o descontinuar la Aplicación o estos Términos en cualquier momento sin previo aviso. Es su responsabilidad revisar periódicamente estos Términos para estar informado de cualquier cambio. El uso continuado de la Aplicación después de dichas modificaciones constituirá su aceptación de los Términos modificados.

6. Ley Aplicable y Jurisdicción:

Estos Términos se regirán e interpretarán de acuerdo con las leyes del país o región. Cualquier disputa derivada de estos Términos estará sujeta a la jurisdicción exclusiva de los tribunales del país o región competentes.

Si tiene alguna pregunta o inquietud sobre estos Términos, contáctenos a través de información de contacto del desarrollador.

Al instalar y utilizar la Aplicación, usted reconoce haber leído, entendido y aceptado estos Términos y se compromete a cumplir con todas las disposiciones establecidas en ellos.