

M8_AE2_ABP

Objetivo

Cada estudiante realizará una revisión exhaustiva de su proyecto desarrollado a lo largo del curso, identificará áreas de mejora, aplicará ajustes finales y documentará su proceso de aprendizaje.

Instrucciones

1. Revisión del Producto:

- Abre tu proyecto y examina cada componente.

Proyecto seleccionado, aplicación web en Django implementando CRUD para una organización sin fines de lucro que necesita una plataforma para gestionar voluntarios que participan en distintos eventos comunitarios.

- Responde en un documento:

- ¿Qué funcionalidades implementaste con éxito?

Gestión de Datos (CRUD)

Para los Voluntarios: Registro completo con validaciones (Email único, teléfono).

Para los Eventos: Creación y edición de eventos con descripción y fechas.

Para las Asignación: Relación "Muchos a Muchos" (`ManyToManyField`) que permite vincular múltiples voluntarios a distintos eventos.

Personalización visual sobre Bootstrap 5

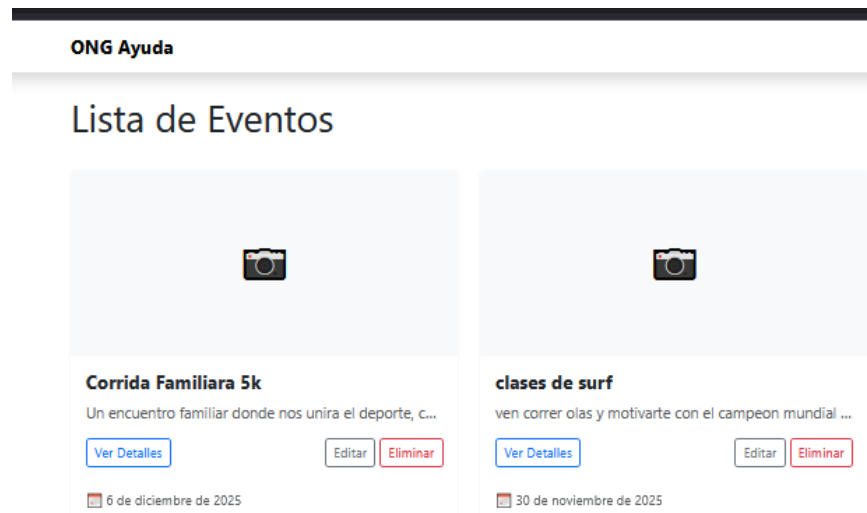
Seguridad: Protección en todos los formularios mediante

`{% csrf_token %}`.

- ¿Qué partes aún podrían mejorarse?

El modelo Auth la autenticación y autorización de usuarios, actualmente cualquier usuario podría crear, eliminar o editar eventos.

La Visual, Intente hacer una galería de imágenes con cards para cada evento, pero no resulto del todo, no supe como hacer que se mostrara las imágenes.



Personalización visual sobre Bootstrap 5 para darle diseño a la sección “listado de voluntarios” .



Mejoras sustanciales en el Formulario, por ejemplo incluyendo mensajes.

- ¿Hay errores o fallos identificados durante su uso?

Fueron solventados durante la ejecución de ese proyecto, pero las mejoras podrían arrojar nuevos errores.

Dejo por acá el enlace al repositorio original antes de implementar las estrategias sugeridas en esta actividad:

https://github.com/EmyQuintana/m7_ae6.git

2. Depuración y Mejora:

- Corrige errores en código, diseño o funcionalidad.
- Refactoriza el código si es necesario para mejorar su legibilidad y eficiencia.
- Agrega documentación clara en comentarios o README.md si aún no lo has hecho.

Refactorización:

1) Refactorización de `views.py` manteniendo las funciones, pero haciéndolas robustas y profesionales. Se actualiza `voluntariado/views.py` en el código se dejan comentarios de los cambios, algunos son:

- Uso de `transaction.atomic` para evitar código roto en caso de que falle el asociar eventos con voluntarios.
- Uso de `messages` para mostrar mensaje al usuario.
- `try/except IntegrityError` para los correos, si el correo ingresado ya existe se captura el error y muestra un mensaje al usuario.

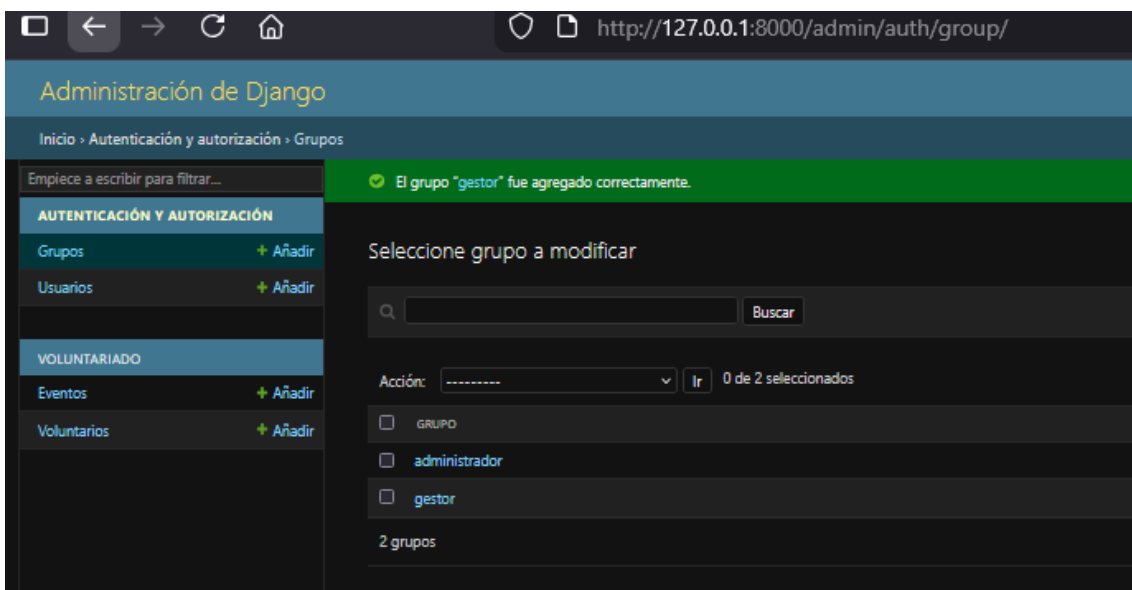
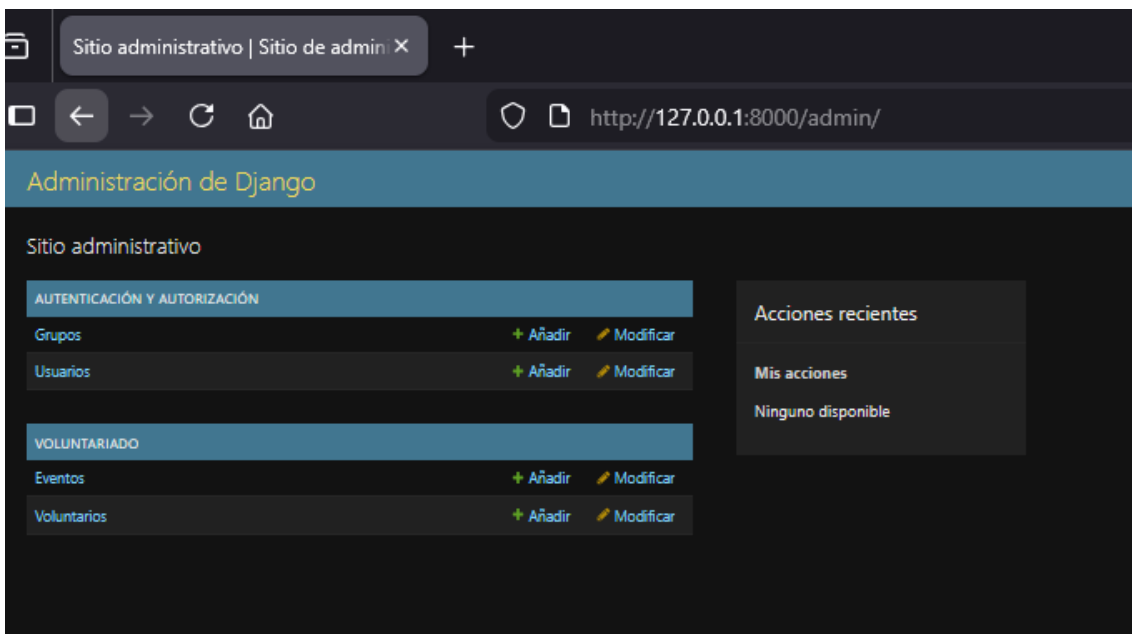
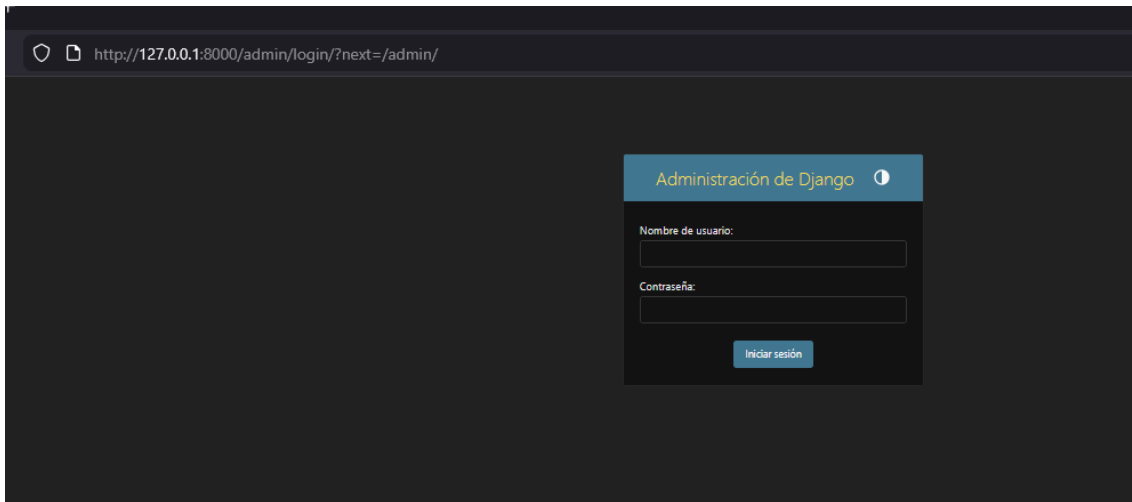
2) Refactorización de `Template Base.html` mejorando el código para que el usuario vea los mensajes como "Evento creado..."

3) Refactorización de `Models.py` se agrega la clase `Meta` la que ordena automáticamente los voluntarios o eventos y para que el panel admin de use correctamente el "voluntario/Voluntarios" según sea el caso.

Nombres legibles: Para que en el panel de administración de Django no diga "Voluntarios" sino "Voluntario" o "Voluntarios" correctamente (singular/plural).

4) Refactorización de `voluntariado/forms.py` mejorando el estilo con Bootstrap.

5) Refactorización del `Template lista_voluntarios.html` cambiándolo de Lista a Tabla para que se vea más profesional.



1. Descripción Final del Producto

"La plataforma desarrollada es una aplicación web completa basada en Django para la gestión operativa de la 'ONG Emy'. El objetivo principal del sistema es administrar eficientemente la base de datos de voluntarios y la planificación de eventos deportivos.

- El producto final permite realizar las siguientes funciones clave:
- Gestión de Eventos: Crear, editar y listar eventos deportivos, incluyendo detalles como fecha, descripción, una foto de portada y, lo más importante, un control de aforo máximo.
- Gestión de Voluntarios: Registrar nuevos participantes con sus datos de contacto y una foto de perfil, permitiendo asignarlos fácilmente a los eventos disponibles.
- Visualización Dinámica: En lugar de listas simples, el sistema muestra la información en tarjetas (Cards) con indicadores visuales, como barras de progreso para ver cuántos cupos quedan disponibles en un evento.
- Sistema de Usuarios: La aplicación cuenta con un módulo de autenticación seguro donde los usuarios pueden registrarse, iniciar y cerrar sesión, accediendo a un panel de control personalizado."

2. Mejoras Aplicadas tras la Revisión (Refactorización)

"Durante la etapa de refactorización y mejora del código, implementé cambios importantes para que el proyecto fuera más profesional y seguro:

- Seguridad y Roles: Pasé de un sistema abierto a uno protegido. Creé grupos de usuarios (Administradores y Gestores). Ahora, un 'Gestor' puede crear y editar eventos, pero solo el Administrador tiene permiso para eliminar registros. Además, los botones de acción se ocultan automáticamente si el usuario no tiene permisos.
- Manejo de Imágenes (Media): Solucioné la carga de archivos estáticos. Ahora los eventos y voluntarios pueden tener imágenes reales que se suben desde el formulario y se guardan en el servidor, mejorando mucho la estética.
- Control de Aforo: Agregué lógica matemática en los modelos y templates. Ahora el sistema calcula automáticamente cuántos inscritos hay frente a la capacidad total y muestra una alerta si el evento está lleno.
- Interfaz de Usuario (UX): Mejoré los formularios de Django. En lugar de listas interminables, implementé cajas con scroll para seleccionar voluntarios, evitando que el diseño se rompa, y agregué una barra de navegación que cambia según si el usuario está logueado o no."

3. Reflexión Personal

"Desarrollar este proyecto ha sido una curva de aprendizaje muy productiva. Al principio, mi enfoque estaba solo en 'que funcione' y entregar, pero durante la refactorización entendí la importancia de escribir código limpio y ordenado.

Lo que más me desafió fue la lógica de los permisos de usuario y el manejo de archivos multimedia, ya que un pequeño error en la configuración hacía que nada se viera o que cualquiera pudiera borrar datos. Solucionar esos errores me ayudó a entender mejor.

Me siento satisfecha con el resultado final.