

# Informe Gestor Mensajes



Autor: Maria Emilia Alvarez  
Materia: Programacion web II  
Profesor: Julio Cesar CASCO  
Fecha: 10/9

## Introducción

El proyecto TableroMensajes es una aplicación web desarrollada con Django que permite a los usuarios gestionar mensajes de manera eficiente. Su objetivo es facilitar la creación, visualización y eliminación de mensajes entre usuarios, aplicando conceptos fundamentales del framework Django, como el patrón de diseño Modelo-Vista-Plantilla (MVT) y el uso del Django Template Language (DTL).

## Desarrollo del Proyecto

### *Configuración del Proyecto Django*

Creación del Entorno Virtual: Cree un entorno virtual utilizando `virtualenv` para gestionar las dependencias del proyecto, asegurando que no haya conflictos con otras aplicaciones.

Clonación del Repositorio: Después de activar el entorno virtual, clone el repositorio del proyecto mediante el comando **`git clone https://github.com/mealvarez/GestorMensaje2.0`** desde GitHub para obtener la estructura inicial y el código base necesario.

Configuración del Proyecto: Configure un nuevo proyecto Django llamado TableroMensajes mediante el comando **`django-admin startproject`**.

Creación de la Aplicación: Dentro del proyecto, cree una aplicación llamada mensajes utilizando el comando python **`manage.py startapp mensajes`**.

### Configuración de urls.py

Creación del archivo urls.py en la aplicación:

En la carpeta de la aplicación mensajes, cree un archivo llamado urls.py para definir las rutas de la aplicación.

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
    path('mensajes/', views.ver_mensajes, name='ver_mensajes'),
    path('buscar/', views.buscar_mensaje, name='buscar_mensaje'),
    path('eliminar/<int:mensaje_id>/', views.eliminar_mensaje,
name='eliminar_mensaje'),
    path('crear/', views.crear_mensaje, name='crear_mensaje'),
]
```

Inclusión de Rutas en el Proyecto Principal:

En el archivo urls.py del proyecto principal, incluí las rutas de la aplicación mensajes.

```
from django.contrib import admin
from django.urls import path, include

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('mensajes.urls')),
]
```

## *Implementación de Funcionalidades*

### Crear Mensajes:

La vista crear\_mensaje renderiza un formulario en la plantilla crear\_mensaje.html que permite a los usuarios ingresar el remitente, destinatario y contenido del mensaje. Al enviar el formulario, se crea un nuevo mensaje utilizando el modelo Mensaje.

### Ver Mensajes Recibidos y Enviados:

La vista ver\_mensajes muestra dos tablas en la plantilla mensajes.html: una para los mensajes recibidos y otra para los enviados. Se utilizan filtros en el modelo Mensaje para mostrar solo los mensajes del usuario actual.

### Buscar Mensajes:

La vista buscar\_mensaje renderiza un formulario de búsqueda en la plantilla buscar.html. Al enviar el formulario, se realiza una consulta al modelo Mensaje utilizando el campo de búsqueda y se muestran los resultados en la misma plantilla.

### Eliminar Mensajes:

En la tabla de mensajes de la plantilla mensajes.html, se incluye un formulario de eliminación para cada mensaje.

Al enviar el formulario de eliminación, se obtiene el ID del mensaje y se elimina utilizando el modelo Mensaje.

## *Herencia y reutilización de Plantillas*

Utilice herencia de plantillas en base.html para crear un layout común que se extiende en otras plantillas como home.html, mensajes.html, crear\_mensaje.html y buscar.html.

El Django Template Language (DTL) lo utilice para renderizar datos dinámicamente en las plantillas, incluyendo la visualización de mensajes y resultados de búsqueda.

Se agregaron estilos CSS personalizados en base.html para mejorar la presentación de las plantillas.

(Los implemente de esta forma debido a que no me los reconoció desde un archivo css que cree en la carpeta assets/css/styles.css, intente de diferentes maneras pero ninguna me funciono. De todos modos dejé la carpeta creada para futuras modificaciones.)

#### Validaciones:

Implemente validaciones en los formularios de creación y búsqueda de mensajes. Asegurándose de que los campos requeridos fueran completados utilizando atributos HTML5 como required, así como validaciones en el lado del servidor mediante el uso de formularios de Django.

En caso de errores de validación, se muestran mensajes de error claros al usuario, permitiendo la corrección de datos antes de volver a enviar el formulario.

#### Vistas

##### Vistas Basadas en Funciones:

- home: Renderiza la página principal de la aplicación, proporcionando una introducción y acceso a las diferentes funcionalidades.
- ver\_mensajes: Muestra los mensajes recibidos y enviados del usuario actual. Utiliza métodos GET para obtener los mensajes de la base de datos.
- buscar\_mensaje: Permite buscar mensajes según criterios específicos. Utiliza el método POST para enviar la consulta y mostrar los resultados.
- eliminar\_mensaje: Elimina un mensaje específico utilizando el método POST para confirmar la acción.
- crear\_mensaje: Crea un nuevo mensaje a partir de los datos del formulario utilizando el método POST.

#### Modelo Mensaje

Defini un modelo Mensaje con los campos remitente, destinatario, contenido y fecha\_envio. Este modelo permite almacenar y gestionar los mensajes en la base de datos.

El campo fecha\_envio lo configure para que se complete automáticamente con la fecha y hora actuales al crear un nuevo mensaje.

```
from django.db import models
from django.contrib.auth.models import User

class Mensaje(models.Model):
    remitente = models.CharField(max_length=100)
    destinatario = models.CharField(max_length=100)
```

```
contenido = models.TextField()
fecha_envio = models.DateTimeField(auto_now_add=True)

def __str__(self):
    return f"{self.remitente} -> {self.destinatario}: {self.contenido}"
```

### *Archivo forms.py*

El archivo forms.py es crucial en mi aplicación, ya que define los formularios utilizados para la entrada de datos. En este archivo, creé formularios para la creación y búsqueda de mensajes. Esto me permitió aplicar validaciones y asegurar que los datos ingresados por los usuarios fueran correctos. Al utilizar Django, el sistema genera automáticamente el HTML necesario para estos formularios, lo que me simplificó el proceso de desarrollo.

```
from django import forms
from .models import Mensaje

class MensajeForm(forms.ModelForm):
    class Meta:
        model = Mensaje
        fields = ['remitente', 'destinatario', 'contenido']
```

## **Problemas encontrados y soluciones**

Implementación de Ver Mensajes: La funcionalidad de ver mensajes recibidos y enviados me presentó dificultades en su implementación inicial. Mi idea principal era utilizar un sistema de login con usuarios predefinidos en un script pero se me hizo demasiado compleja, lo que me llevó a optar por mostrar solo los mensajes del usuario actual.

Validaciones: Al principio, las validaciones no se me implementaron correctamente, lo que permitía el envío de formularios incompletos. Lo solucioné añadiendo validaciones tanto en el lado del cliente como en el servidor.

Manejo de Errores: En la búsqueda de mensajes encontré problemas al manejar consultas vacías. Así que opté por implementar una lógica para manejar estos casos y mostrar un mensaje adecuado al usuario.

## **Conclusión**

El desarrollo de este trabajo práctico me permitió aplicar conceptos vistos en clase. Aunque enfrenté algunos desafíos, como mostrar mensajes de diferentes usuarios, opté por simplificar la funcionalidad a solo los mensajes del usuario actual. En general, disfruté el proceso, ya que me ayudó a comprender mejor la herencia y la renderización en Django.

## Versiones usadas

```
(tp2entorno) mealvarezz@mealvarezz-N14WCE128:~/Escritorio/ProgramacionWebII/tp2entorno/GestorMensaje2.0/TableroMensaje$ pip freeze
asgiref==3.8.1
Django==5.1.1
sqlparse==0.5.1
(tp2entorno) mealvarezz@mealvarezz-N14WCE128:~/Escritorio/ProgramacionWebII/tp2entorno/GestorMensaje2.0/TableroMensaje$
(tp2entorno) mealvarezz@mealvarezz-N14WCE128:~/Escritorio/ProgramacionWebII/tp2entorno/GestorMensaje2.0/TableroMensaje$ python3 --version
Python 3.12.3
```

## Repositorio

<https://github.com/mealvarezz/GestorMensaje2.0>

## Bibliografía

- Vincent, W. S. (2023). Django for Beginners. Independently published.
- Página oficial Django - Django Documentation
- Apuntes de Unidad 2 - Julio César Casco
- Clases grabadas en plataforma Moodle

## Anexo

### Capturas de la Aplicación

#### Página Principal



#### Bienvenido a la Aplicación de Mensajes

Utiliza el menú para navegar por las diferentes secciones.

#### Mensajes



##### Mensajes Recibidos

Remitente	Contenido	Fecha	Acciones
Lucas	Holaaa! Bien bien, ¿vos?	09/09/2024 09:25	<button>Eliminar</button>

##### Mensajes Enviados

Destinatario	Contenido	Fecha	Acciones
Lucas	Hola, ¿como va todo?	09/09/2024 09:24	<button>Eliminar</button>
Horacio	¿Al final venís?	10/09/2024 17:52	<button>Eliminar</button>

#### Buscar



## Buscar Mensajes

  

### Resultados:

Remitente	Destinatario	Contenido	Fecha
Aira	Lucas	Hola, ¿como va todo?	09/09/2024 09:24
Lucas	Aira	Holaaa! Bien bien, ¿vos?	09/09/2024 09:25
Aira	Horacio	¿Al final venis?	10/09/2024 17:52

## Crear mensaje



## Crear Nuevo Mensaje

Remitente:

Destinatario:

Contenido: