

Parte 2: Proyecto (90%)

Enunciado

Título: "Gestión de Denuncias Ciudadanas"

Contexto: En muchas ciudades, los ciudadanos enfrentan problemas como baches en las calles, alumbrado público deficiente, basura acumulada y otros inconvenientes urbanos. Sin una plataforma adecuada, estos problemas pueden tardar en resolverse.

Objetivo: Desarrollar una aplicación web en Django de Gestión de Denuncias con las siguientes características:

1. **Gestión de Usuarios:**
 - a. Registrarse y autenticarse en la plataforma.
 - b. Cerrar sesión.
2. **Gestión de Denuncias:**
 - a. Crear, editar y eliminar denuncias sobre problemas urbanos.
 - b. Adjuntar imágenes y asignar una categoría a las denuncias.
 - c. Ver denuncias de otros usuarios y su estado (pendiente/resuelto).
3. **Gestión de Categorías:**
 - a. Definir un conjunto de categorías predefinidas para las denuncias (por ejemplo, "bache", "alumbrado", "basura", "otro").
 - b. Permitir a los usuarios seleccionar una categoría al crear una denuncia.
4. **Interfaz de Usuario:**
 - a. Utilizar plantillas base con un encabezado (header) que incluya un menú de navegación y un pie de página (footer).
 - b. Aplicar estilos CSS para mejorar la apariencia de la aplicación.
 - c. Proteger las vistas que requieran autenticación.

Ítems a Realizar

1. **Modelos (denuncias/models.py, usuarios/models.py):**
 - a. **denuncias/models.py:**
 - i. **Denuncia:**
 1. **titulo** (CharField): Título de la denuncia.
 2. **descripcion** (TextField): Descripción detallada.
 3. **imagen** (ImageField): Imagen adjunta.

4. `categoria` (CharField, choices=CATEGORIAS): Categoría de la denuncia.
5. `estado` (CharField, default='pendiente'): Estado de la denuncia.
6. `fecha_creacion` : Fecha de creación.
7. `usuario` : Usuario que creó la denuncia.
- ii. CATEGORIAS: Constante con las categorías predefinidas.

b. usuarios/models.py:

- i. Utilizar el modelo User de Django.

2. Formularios (denuncias/forms.py, usuarios/forms.py):

a. denuncias/forms.py:

- i. DenunciaForm:
 1. Basado en el modelo Denuncia.
 2. Incluye todos los campos del modelo.

b. usuarios/forms.py:

- i. CustomUserCreationForm:
 1. Basado en UserCreationForm.
 2. Añade el campo email.

3. Vistas (denuncias/views.py, usuarios/views.py):

a. usuarios/views.py:

- i. `registro` (registro):
 1. Método: GET (muestra el formulario), POST (procesa el formulario).
 2. Autorización: No requiere.
 3. Funcionalidad: Registra un nuevo usuario utilizando CustomUserCreationForm.
- ii. `iniciar_sesion` (login):
 1. Método: GET (muestra el formulario), POST (procesa el formulario).
 2. Autorización: No requiere.
 3. Funcionalidad: Inicia sesión de un usuario utilizando AuthenticationForm.
- iii. `cerrar_sesion` (logout):
 1. Método: GET.
 2. Autorización: Requiere autenticación.
 3. Funcionalidad: Cierra la sesión del usuario.

b. denuncias/views.py:

- i. `lista_denuncias` (denuncias):
 - 1. Método: GET.
 - 2. Autorización: Requiere autenticación.
 - 3. Funcionalidad: Muestra todas las denuncias.
 - ii. `crear_denuncia` (denuncias/crear):
 - 1. Método: GET (muestra el formulario), POST (procesa el formulario).
 - 2. Autorización: Requiere autenticación.
 - 3. Funcionalidad: Crea una nueva denuncia utilizando `DenunciaForm`.
 - iii. `editar_denuncia` (denuncias/editar/<int:id>):
 - 1. Método: GET (muestra el formulario), POST (procesa el formulario).
 - 2. Autorización: Requiere autenticación.
 - 3. Funcionalidad: Edita una denuncia existente utilizando `DenunciaForm`.
 - iv. `eliminar_denuncia` (denuncias/eliminar/<int:id>):
 - 1. Método: GET (muestra la confirmación), POST (elimina la denuncia).
 - 2. Autorización: Requiere autenticación.
 - 3. Funcionalidad: Elimina una denuncia existente.
- 4. URLs (`gestion_denuncias/urls.py`, `denuncias/urls.py`, `usuarios/urls.py`):**
- a. **`gestion_denuncias/urls.py`:**
 - i. `/admin/`: Panel de administración de Django.
 - ii. `/usuarios/`: Incluye `usuarios.urls`.
 - iii. `/`: Incluye `denuncias.urls`.
 - b. **`usuarios/urls.py`:**
 - i. `/registro/`: Vista registro.
 - ii. `/login/`: Vista `iniciar_sesion`.
 - iii. `/logout/`: Vista `cerrar_sesion`.
 - c. **`denuncias/urls.py`:**
 - i. `denuncias/`: Vista `lista_denuncias`.
 - ii. `denuncias/crear/`: Vista `crear_denuncia`.
 - iii. `denuncias/editar/<int:id>/`: Vista `editar_denuncia`.
 - iv. `denuncias/eliminar/<int:id>/`: Vista `eliminar_denuncia`.
- 5. Plantillas (`templates/base.html`, `usuarios/templates/*.html`, `denuncias/templates/*.html`):**

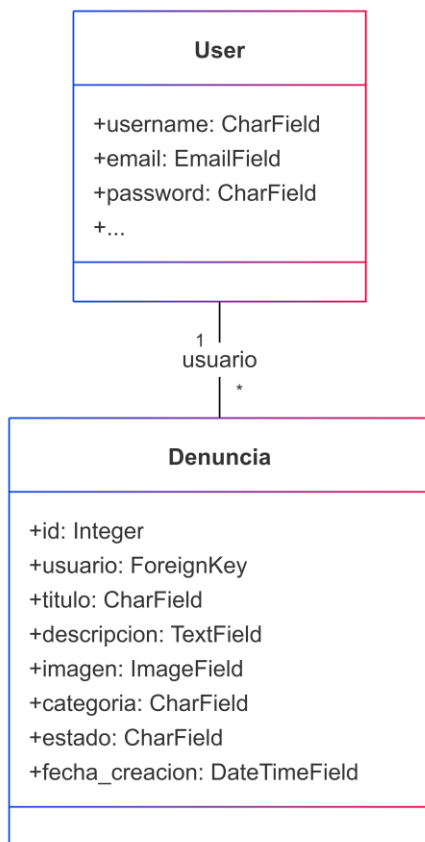
a. **templates/base.html:**

- i. Estructura HTML común.
- ii. Header con menú de navegación.
- iii. Footer.
- iv. Bloques .
- v. Carga de archivos estáticos.

b. **usuarios/templates/*.html:**

- i. registro.html: Formulario de registro.
- ii. login.html: Formulario de inicio de sesión.
- iii. lista_denuncias.html: Lista de denuncias.
 - 1. Autenticación.
- iv. crear_denuncia.html: Formulario de creación.
 - 1. Autenticación.
- v. editar_denuncia.html: Formulario de edición.
 - 1. Autenticación.
- vi. eliminar_denuncia.html: Confirmación de eliminación.
 - 1. Autenticaciónco.

Diagrama UML del Modelo de Datos



Explicación del Diagrama UML:

- **User:**
 - Representa a los usuarios registrados en la plataforma.
 - Incluye atributos como username, email y password.
- **Denuncia:**
 - Representa las denuncias creadas por los usuarios.
 - Incluye atributos como titulo, descripcion, imagen, categoria y estado.
 - Tiene una relación de clave externa (ForeignKey) con el modelo User, indicando que cada denuncia pertenece a un usuario.
- **Relación:**
 - La línea entre User y Denuncia indica una relación de uno a muchos (1..*), donde un usuario puede tener múltiples denuncias.

← ↻ ⓘ 127.0.0.1:8000/usuarios/login/ 🔍 ☆ 📧 ⚙️ 🏠 ☰ 🌐

Gestión de Denuncias Ciudadanas

Iniciar sesión Registrarse

Iniciar sesión

Username:

Password:

¿No tienes una cuenta? [Registrarse](#)

© 2025 Gestión de Denuncias - IES Isidra Guzmán

Crud de Denuncias:



The screenshot shows a web browser window with the URL `127.0.0.1:8000/denuncias/crear/`. The page title is "Gestión de Denuncias Ciudadanas". The navigation bar includes links for "Denuncias", "Crear denuncia", and "Cerrar sesión". The main heading is "Crear denuncia".

The form contains the following fields:

- Título:** A text input field containing "Baches de calle".
- Descripción:** A text area containing "Desperfecto en calle".
- Imagen:** A file upload button labeled "Elegir archivo" and a filename "lenguajes-de-pro...-mas-usados-1.jpg".
- Categoría:** A dropdown menu with "Bache en la calle" selected. A dropdown menu is open showing options: "Bache en la calle", "Alumbrado público deficiente", "Basura acumulada", and "Otro".
- Crear:** A button to submit the form.

At the bottom of the form, there is a copyright notice: "© 2025 Gestión de Denuncias - IES Isidra Guzmán". The Windows taskbar is visible at the bottom of the screen.

← ↻ ⓘ 127.0.0.1:8000/denuncias/

Gestión de Denuncias Ciudadanas

Lista de denuncias

• Ruidos molestos

Ruidos molestos música alta

Categoría: bache

Estado: pendiente

[Editar](#) [Eliminar](#)

• Baches de calle

Peligrosos baches de carretera

Categoría: bache

Estado: pendiente

[Editar](#) [Eliminar](#)

Gestión de Denuncias Ciudadanas

[Denuncias](#)

[Crear denuncia](#)

[Cerrar sesión](#)

Editar denuncia

Título:

Ruidos molestos música alta

Descripción:

Imagen: Currently: [denuncias/cap1_QuBW8ly.PNG](#) ☐ Clear
Change: No se ha seleccionado ningún archivo

Categoría:

© 2025 Gestión de Denuncias - IES Isidra Guzmán

← ↻ ⓘ 127.0.0.1:8000/denuncias/eliminar/2/

Gestión de Denuncias Ciudadanas

[Denuncias](#)

[Crear denuncia](#)

[Cerrar sesión](#)

Eliminar denuncia

¿Estás seguro de que quieres eliminar la denuncia "Ruidos molestos"?

[Cancelar](#)

© 2025 Gestión de Denuncias - IES Isidra Guzmán

Criterios Calificación

El programa tiene que funcionar poder acceder a cada una de las secciones. Para tener en cuenta cada uno de los apartados tiene que funcionar dentro del conjunto del apartado. Es decir, el código tiene que ser funcional y estar compenetrado con el resto de partes.

1. Gestión de Usuarios (3 puntos):

- a. 1 punto por la implementación del registro de usuarios con formulario.
- b. 1 punto por la implementación del inicio de sesión.
- c. 1 punto por la implementación del cierre de sesión.

2. Gestión de Denuncias (4 puntos):

- a. 1 punto por la implementación de la creación de denuncias.
- b. 1 punto por la implementación de la visualización de denuncias.
- c. 1 punto por la implementación de la edición de denuncias.
- d. 1 punto por la implementación de la eliminación de denuncias.

3. Interfaz de Usuario (3 puntos):

- a. 1 punto por la implementación de plantillas base con header y footer.
- b. 1 punto por la aplicación de estilos CSS.
- c. 1 punto por la protección de vistas con autenticación.

4. Funcionalidad Total (Dependencia):

- a. Si la aplicación no funciona, la nota máxima será de 0 puntos.