Ejercicio Práctico: Sistema de Notas Seguras

Duración estimada: 90 minutos **Tecnologías:** Python, Django, uv **Objetivo:** Construir una aplicación web donde los **Profesores** pueden asignar calificaciones y los **Estudiantes** solo pueden ver sus propias notas.

---

1. Preparación del Entorno

Vamos a crear un proyecto desde cero utilizando `uv` para la gestión de dependencias.

1.1. Crear directorio y entorno virtual

Abre tu terminal y ejecuta los siguientes comandos uno por uno:

```
# 1. Crear carpeta del proyecto y entrar
mkdir notas_seguras
cd notas_seguras

# 2. Inicializar proyecto con uv
uv init
uv add django

# 3. Crear estructura del proyecto Django
uv run django-admin startproject config .
uv run python manage.py startapp escuela
```

1.2. Verificar instalación

Ejecuta el servidor para asegurar que todo está bien:

```
uv run python manage.py runserver
```

Deberías ver la página de bienvenida de Django en http://127.0.0.1:8000/. Detén el servidor con `Ctrl+C`.

---

2. Modelado de Datos (Models)

Vamos a personalizar el usuario para tener roles y crear la tabla de calificaciones.

Archivo: `escuela/models.py` **Acción:** Borra el contenido y pega lo siguiente:

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

# 1. USUARIO PERSONALIZADO (Roles)
```

```python
class Usuario(AbstractUser):
    ROLES = (
        ('profe', 'Profesor'),
        ('alumno', 'Estudiante'),
    )
    rol = models.CharField(max_length=10, choices=ROLES, default='alumno')

# 2. EL DATO PROTEGIDO (La Calificación)
class Calificacion(models.Model):
    # Relacionamos la nota con un alumno específico
    # limit_choices_to asegura que solo aparezcan alumnos en el selector del admin/forms
    alumno = models.ForeignKey(Usuario, on_delete=models.CASCADE, related_name='mis_notas',
    materia = models.CharField(max_length=50) # Ej: Matemáticas
    nota = models.DecimalField(max_digits=4, decimal_places=2) # Ej: 9.50

    def __str__(self):
        return f"{self.materia}: {self.nota} ({self.alumno.username})"
```

---

3. Configuración del Proyecto

Necesitamos decirle a Django que use nuestro usuario personalizado y dónde encontrar nuestra app.

Archivo: `config/settings.py` **Acción:** Realiza los siguientes cambios:

1. Busca `INSTALLED_APPS` y agrega `'escuela',` al final.
2. Al final de todo el archivo, agrega las configuraciones de autenticación.

```python
# En INSTALLED_APPS:
INSTALLED_APPS = [
    # ... apps por defecto ...
    'django.contrib.staticfiles',
    'escuela', # <--- AGREGAR ESTO
]

# ... (resto del archivo) ...

# AL FINAL DEL ARCHIVO AGREGAR:
AUTH_USER_MODEL = 'escuela.Usuario'
LOGIN_REDIRECT_URL = 'dashboard'
LOGOUT_REDIRECT_URL = 'login'
```

---

4. Base de Datos y Administrador

Aplicamos los cambios y preparamos el panel de administración.

4.1. Migraciones

En la terminal:

```
uv run python manage.py makemigrations
uv run python manage.py migrate
```

4.2. Configurar el Admin

Para poder gestionar usuarios y notas fácilmente.

Archivo: `escuela/admin.py` **Acción:** Reemplaza el contenido con:

```
from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from .models import Usuario, Calificacion

# Registramos el usuario usando UserAdmin para que se vea bien en el panel
admin.site.register(Usuario, UserAdmin)
admin.site.register(Calificacion)
```

4.3. Crear Superusuario

En la terminal:

```
uv run python manage.py createsuperuser
```

*(Usa usuario: **admin**, email: vacío, contraseña: **admin** o la que prefieras)*

---

5. Lógica de Seguridad (Vistas y Forms)

Aquí definimos quién puede ver qué.

5.1. Formulario

Primero, necesitamos un formulario para subir notas.

Archivo: Crea un nuevo archivo `escuela/forms.py` **Contenido:**

```
from django import forms
from .models import Calificacion

class CalificacionForm(forms.ModelForm):
    class Meta:
        model = Calificacion
        fields = ['alumno', 'materia', 'nota']
```

5.2. Vistas (Views)

La lógica principal de protección.

Archivo: `escuela/views.py` **Acción:** Reemplaza el contenido con:

```
from django.shortcuts import render, redirect
from django.contrib.auth.decorators import login_required
from .models import Calificacion
from .forms import CalificacionForm

@login_required
def dashboard(request):
    usuario = request.user

    # --- LÓGICA DE AUTORIZACIÓN (Row-Level Security) ---
    if usuario.rol == 'profe':
        # Si es profe, ve TODAS las notas ordenadas por alumno
        notas = Calificacion.objects.all().order_by('alumno')
    else:
        # Si es alumno, SOLO ve SUS notas (Filtro por fila)
        notas = Calificacion.objects.filter(alumno=usuario)

    return render(request, 'dashboard.html', {'notas': notas})

@login_required
def subir_nota(request):
    # --- CONTROL DE ACCESO (Role-Based Access Control) ---
    # Solo dejamos pasar si es profe
    if request.user.rol != 'profe':
        return redirect('dashboard') # O lanzar error 403

    if request.method == 'POST':
        form = CalificacionForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('dashboard')
    else:
        form = CalificacionForm()

    return render(request, 'subir_nota.html', {'form': form})
```

---

6. Interfaz Gráfica (Templates)

Vamos a crear las páginas HTML.

6.1. Crear carpetas

Crea la siguiente estructura de carpetas dentro de `escuela`:

1. `escuela/templates/`
2. `escuela/templates/registration/` (Nota: `registration` es obligatorio para el login por defecto)

6.2. Dashboard

Archivo: `escuela/templates/dashboard.html` **Contenido:**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Notas Seguras</title>
    <style>
      body {
        font-family: sans-serif;
        padding: 20px;
        max-width: 800px;
        margin: 0 auto;
      }
      table {
        width: 100%;
        border-collapse: collapse;
        margin-top: 20px;
      }
      th,
      td {
        border: 1px solid #ddd;
        padding: 8px;
        text-align: left;
      }
      th {
        background-color: #f2f2f2;
      }
      .btn {
        padding: 10px;
        text-decoration: none;
        background: #007bff;
        color: white;
        border-radius: 5px;
      }
      .logout {
        background: #dc3545;
        border: none;
```

```
      color: white;
      padding: 5px 10px;
      cursor: pointer;
      border-radius: 3px;
    }
    .header {
      display: flex;
      justify-content: space-between;
      align-items: center;
      margin-bottom: 20px;
    }
  </style>
</head>
<body>
  <div class="header">
    <h2>  Panel de Notas</h2>
    <form action="{% url 'logout' %}" method="post">
      {% csrf_token %}
      <span
        >Hola, <b>{{ user.username }}</b> ({{ user.get_rol_display }})</span
      >
      <button type="submit" class="logout">Cerrar Sesión</button>
    </form>
  </div>

  <hr />

  {% if user.rol == 'profe' %}
  <div style="margin: 20px 0;">
    <a href="{% url 'subir_nota' %}" class="btn">+ Agregar Nueva Nota</a>
  </div>
  {% endif %}

  <table>
    <thead>
      <tr>
        <th>Materia</th>
        <th>Nota</th>
        {% if user.rol == 'profe' %}
        <th>Alumno</th>
        {% endif %}
      </tr>
    </thead>
    <tbody>
      {% for c in notas %}
      <tr>
```

```
          <td>{{ c.materia }}</td>
          <td>{{ c.nota }}</td>
          {% if user.rol == 'profe' %}
          <td>{{ c.alumno.username }}</td>
          {% endif %}
        </tr>
        {% empty %}
        <tr>
          <td colspan="3">No hay notas registradas.</td>
        </tr>
        {% endfor %}
      </tbody>
    </table>
  </body>
</html>
```

6.3. Subir Nota

Archivo: `escuela/templates/subir_nota.html` **Contenido:**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Subir Nota</title>
    <style>
      body {
        font-family: sans-serif;
        padding: 20px;
        max-width: 600px;
        margin: 0 auto;
      }
    </style>
  </head>
  <body>
    <h2>  Subir Calificación</h2>
    <form method="post">
      {% csrf_token %} {{ form.as_p }}
      <br />
      <button type="submit">Guardar</button>
      <a href="{% url 'dashboard' %}" style="margin-left: 10px;">Cancelar</a>
    </form>
  </body>
</html>
```

6.4. Login

Archivo: `escuela/templates/registration/login.html` **Contenido:**

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Iniciar Sesión</title>
    <style>
      body {
        font-family: sans-serif;
        padding: 50px;
        text-align: center;
      }
    </style>
  </head>
  <body>
    <h2>  Acceso al Sistema</h2>
    <form method="post">
      {% csrf_token %} {{ form.as_p }}
      <button type="submit">Entrar</button>
    </form>
  </body>
</html>
```

---

7. Configuración de URLs

Conectamos todo.

Archivo: `config/urls.py` **Acción:** Reemplaza el contenido con:

```python
from django.contrib import admin
from django.urls import path, include
from escuela import views

urlpatterns = [
    path('admin/', admin.site.urls),
    # Incluye las URLs de autenticación (login, logout, password_change, etc.)
    path('accounts/', include('django.contrib.auth.urls')),

    # Nuestras vistas
    path('', views.dashboard, name='dashboard'),
    path('subir/', views.subir_nota, name='subir_nota'),
]
```

---

8. Validación y Pruebas

¡Hora de probar si el sistema es seguro!

8.1. Crear Datos de Prueba

1. Ejecuta el servidor: `uv run python manage.py runserver`
2. Entra a `http://127.0.0.1:8000/admin/` con tu superusuario.
3. En la sección **Usuarios**, crea:
   - Usuario: `ProfeJavier` -> Rol: **Profesor**
   - Usuario: `AlumnoLuis` -> Rol: **Estudiante**
   - Usuario: `AlumnoAna` -> Rol: **Estudiante**
4. Cierra sesión del admin.

8.2. Prueba de Profesor

1. Loguéate como `ProfeJavier`.
2. Deberías ver el botón **"+ Agregar Nueva Nota"**.
3. Agrega una nota para **Luis** (Matemáticas: 8) y otra para **Ana** (Historia: 10).
4. En el dashboard, deberías ver **ambas** notas.

8.3. Prueba de Aislamiento (Alumno)

1. Cierra sesión y entra como `AlumnoLuis`.
2. **Verificación 1:** ¿Ves el botón de agregar nota? -> **No debería estar.**
3. **Verificación 2:** ¿Ves la nota de Ana? -> **No deberías verla.** Solo la tuya.
4. **Verificación 3 (Hacking):** Intenta forzar la entrada escribiendo en la URL: `http://127.0.0.1:8000/subir/`.
   - *Resultado esperado:* El sistema debe redirigirte al dashboard automáticamente.
