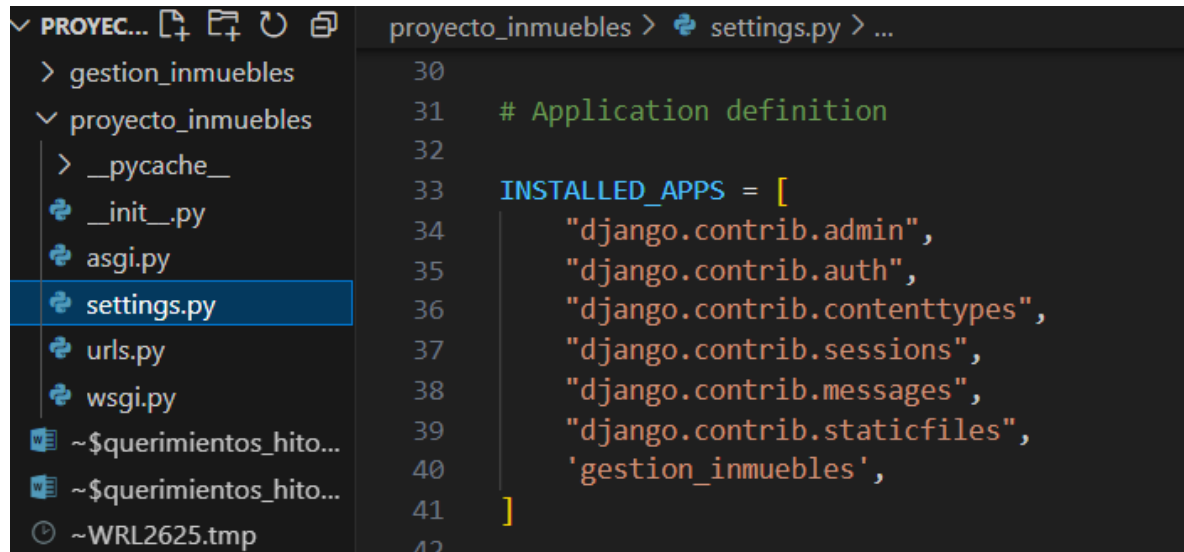


## Requerimientos


### 1. Configuración de la Autenticación (3 puntos)

a. Asegurarse de que `django.contrib.auth` y `django.contrib.contenttypes` están en `INSTALLED_APPS`.



```
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     "django.contrib.admin",
35     "django.contrib.auth",
36     "django.contrib.contenttypes",
37     "django.contrib.sessions",
38     "django.contrib.messages",
39     "django.contrib.staticfiles",
40     'gestion_inmuebles',
41 ]
42
```

b. Configurar URLs para las vistas de autenticación (registro, inicio y cierre de sesión).



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

urlpatterns = [
    path('register/', views.register, name='register'), # URL para el registro
    path('login/', views.login_view, name='login'), # URL para inicio de sesión
    path('logout/', views.logout_view, name='logout'), # URL para cerrar sesión
]
```

## Registrarse

Username:  Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:  Enter the same password as before, for verification.

Registrar

## Iniciar sesión

Username:

Password:

Iniciar sesión

- c. Crear un superusuario para acceder al panel de administración.
2. Creación de la Vista y Formulario de Registro (3 puntos)
    - a. Crear la vista de registro utilizando UserCreationForm.

```

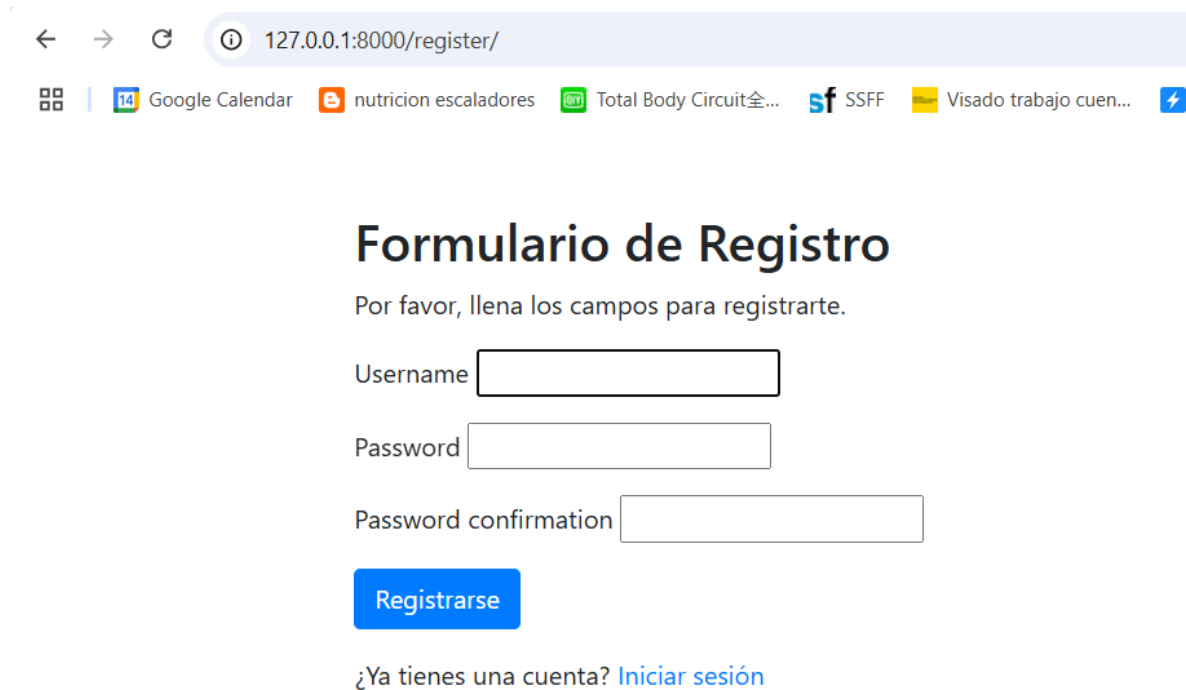
from django.shortcuts import render, redirect
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth import login, authenticate, logout
from django.contrib.auth.forms import AuthenticationForm
from django.contrib.auth.decorators import login_required

def register(request):
    if request.method == "POST":
        form = UserCreationForm(request.POST)
        if form.is_valid():
            form.save()
            username = form.cleaned_data.get("username")
            password = form.cleaned_data.get("password1")
            user = authenticate(username=username, password=password)
            login(request, user)
            return redirect("home")
        else:
            form = UserCreationForm()

    return render(request, "registration/register.html", {"form": form})

```

b. Crear el template HTML para el formulario de registro.



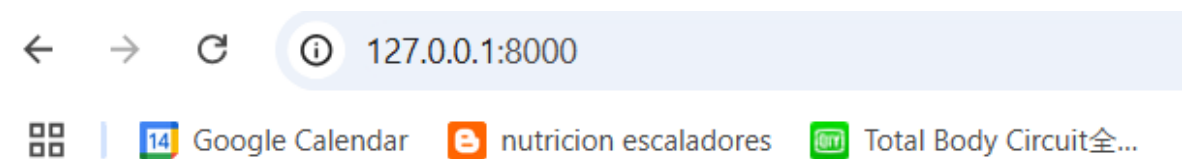
The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/register/'. The browser's taskbar at the bottom includes icons for Google Calendar, nutricion escaladores, Total Body Circuit, SSFF, and Visado trabajo cuen... The main content of the page is a registration form titled 'Formulario de Registro'. Below the title, it says 'Por favor, llena los campos para registrarte.' The form contains three input fields: 'Username', 'Password', and 'Password confirmation'. Below these fields is a blue button labeled 'Registrarse'. At the bottom of the form, there is a link that says '¿Ya tienes una cuenta? [Iniciar sesión](#)'.

```

gestion_inmuebles > templates > registration > register.html > ...
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Registro de Usuario</title>
7   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
8 </head>
9 <body>
10   <div class="container">
11     <h2 class="mt-5">Formulario de Registro</h2>
12     <p>Por favor, llena los campos para registrarte.</p>
13
14     <form method="POST">
15       {% csrf_token %} <!-- Protege contra ataques CSRF -->
16
17       <!-- Mostrar el formulario en un formato de lista -->
18       <div class="form-group">
19         {{ form.username.label }}
20         {{ form.username }}
21         {% if form.username.errors %}
22           <div class="alert alert-danger">
23             {% for error in form.username.errors %}
24               <p>{{ error }}</p>
25             {% endfor %}
26           </div>
27         {% endif %}
28       </div>
29
30       <div class="form-group">
31         {{ form.password1.label }}
32         {{ form.password1 }}
33         {% if form.password1.errors %}
34           <div class="alert alert-danger">
35             {% for error in form.password1.errors %}
36               <p>{{ error }}</p>
37             {% endfor %}
38           </div>
39         {% endif %}
40       </div>
41
42       <div class="form-group">
43         {{ form.password2.label }}
44         {{ form.password2 }}
45         {% if form.password2.errors %}
46           <div class="alert alert-danger">
47             {% for error in form.password2.errors %}
48               <p>{{ error }}</p>
49             {% endfor %}
50           </div>
51         {% endif %}
52       </div>
53
54       <!-- Botón para enviar el formulario -->
55       <button type="submit" class="btn btn-primary">Registrarse</button>
56     </form>
57
58     <p class="mt-3">¿Ya tienes una cuenta? <a href="{% url 'login' %}">Iniciar sesión</a></p>
59   </div>
60
61   <!-- Scripts de Bootstrap -->
62   <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
63   <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"></script>
64   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
65 </body>
66 </html>

```

c. Asegurarse de que los usuarios pueden registrarse correctamente.



# Bienvenido a la página de inicio

¡Te has registrado exitosamente!

3. Creación de las Vistas de Inicio y Cierre de Sesión (3 puntos)

- a. Crear las vistas de inicio y cierre de sesión utilizando LoginView y LogoutView.

```
from django.contrib.auth.views import LoginView, LogoutView

# Vista de inicio de sesión
class CustomLoginView(LoginView):
    template_name = 'registration/login.html'
    redirect_authenticated_user = True


# Vista de cierre de sesión
class CustomLogoutView(LogoutView):
    next_page = '/'
```

- b. Crear los templates HTML para los formularios de inicio y cierre de sesión.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Inicio de sesión</title>
</head>
<body>
    <h2>Iniciar sesión</h2>
    <form method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Iniciar sesión</button>
    </form>
    <p>¿No tienes una cuenta? <a href="{% url 'register' %}">Regístrate</a></p>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cierre de sesión</title>
</head>
<body>
  <h2>Has cerrado sesión correctamente</h2>
  <p><a href="{% url 'login' %}">Iniciar sesión de nuevo</a></p>
</body>
</html>
```

c. Asegurarse de que los usuarios pueden iniciar y cerrar sesión correctamente.



← → ↻ ⓘ 127.0.0.1:8000/login/?next=/logout/

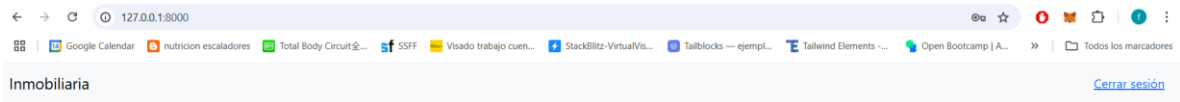
🗄️ | 📅 Google Calendar 📄 nutricion escaladores 📺 Total Body Circuit全... S

## Iniciar sesión

Username:

Password:

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



Bienvenido al inicio

[Cerrar sesión](#)

© 2024 Proyecto Inmobiliario. Todos los derechos reservados.

← → ↺ ⓘ 127.0.0.1:8000/logout/

🗑️ | 📅 Google Calendar 📄 nutricion escaladores 📺 Total Body Circuit全... 📄 sf

Has cerrado sesión correctamente

[Iniciar sesión de nuevo](#)

#### 4. Gestión de Permisos y Grupos de Usuarios (1 punto)

##### a. Configurar permisos específicos para diferentes tipos de usuarios.

Permissions

☒ Active  
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☐ Staff status  
Designates whether the user can log into this admin site.

☐ Superuser status  
Designates that this user has all permissions without explicitly assigning them.

Groups

Available groups

Choose all

Chosen groups

Remove all

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions

Available user permissions

Administration | log entry | Can change log entry

Administration | log entry | Can delete log entry

Administration | log entry | Can view log entry

Authentication and Authorization | group | Can add group

Authentication and Authorization | group | Can delete group

Authentication and Authorization | group | Can view group

Authentication and Authorization | permission | Can add permission

Authentication and Authorization | permission | Can change permission

Authentication and Authorization | permission | Can delete permission

Authentication and Authorization | permission | Can view permission

Choose all

Chosen user permissions

Authentication and Authorization | group | Can change group

Remove all

Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.

b. Crear y asignar grupos de usuarios con permisos específicos.

The image displays two screenshots of the Django administration interface, specifically the 'Groups' section.

**Top Screenshot:** The 'Groups' page is shown. A green message at the top indicates 'The group "Arrendadores" was added successfully.' Below this, the 'Select group to change' section is visible, showing a search bar and a list of groups. The 'Arrendadores' group is selected.

**Bottom Screenshot:** The 'Add group' page is shown. The 'Name' field is filled with 'Arrendadores'. The 'Permissions' section is active, showing a list of available permissions. The 'Chosen permissions' section is also visible, showing the selected permissions for the 'Arrendadores' group.