

UNIVERSIDAD MAYOR DE SAN ANDRES

CARRERA DE INFORMATICA

## PROGRAMACION WEB III

**INF-133**



link: <https://github.com/oOmonstersOo/Tarea-Proyecto-Prueba.git>

### Tarea: Proyecto Prueba

Estudiante : Mauricio Marco Zamora Rivas

C.I. : 10094296

Docente: Lic. Marcelo Aruquipa

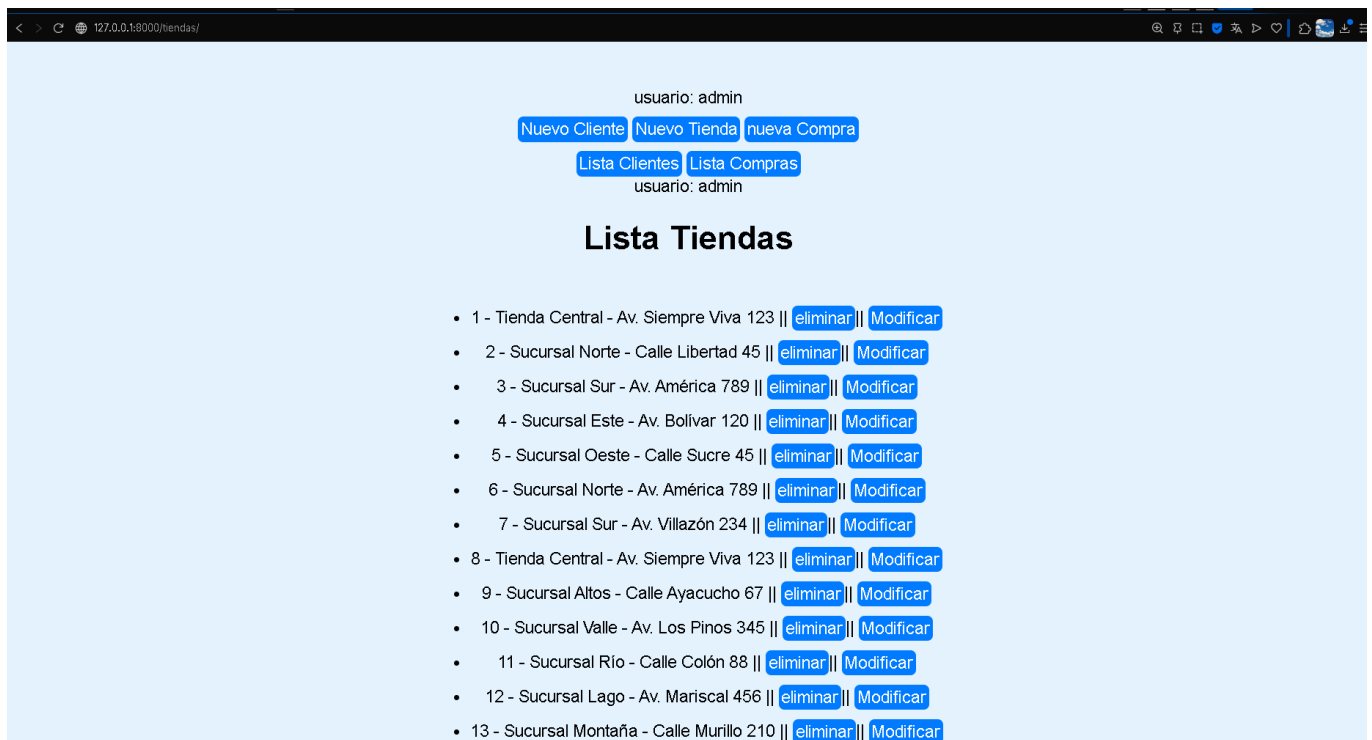
Fecha: 14 de noviembre de 2025

LA PAZ - BOLIVIA

2025

## CAPTURAS DE PANTALLA

a. Se realizo el CRUD para la tabla Clientes, completar para el resto de las tablas





b. Modificar el siguiente detalle al momento de realizar una compra

Fecha:  Hoy

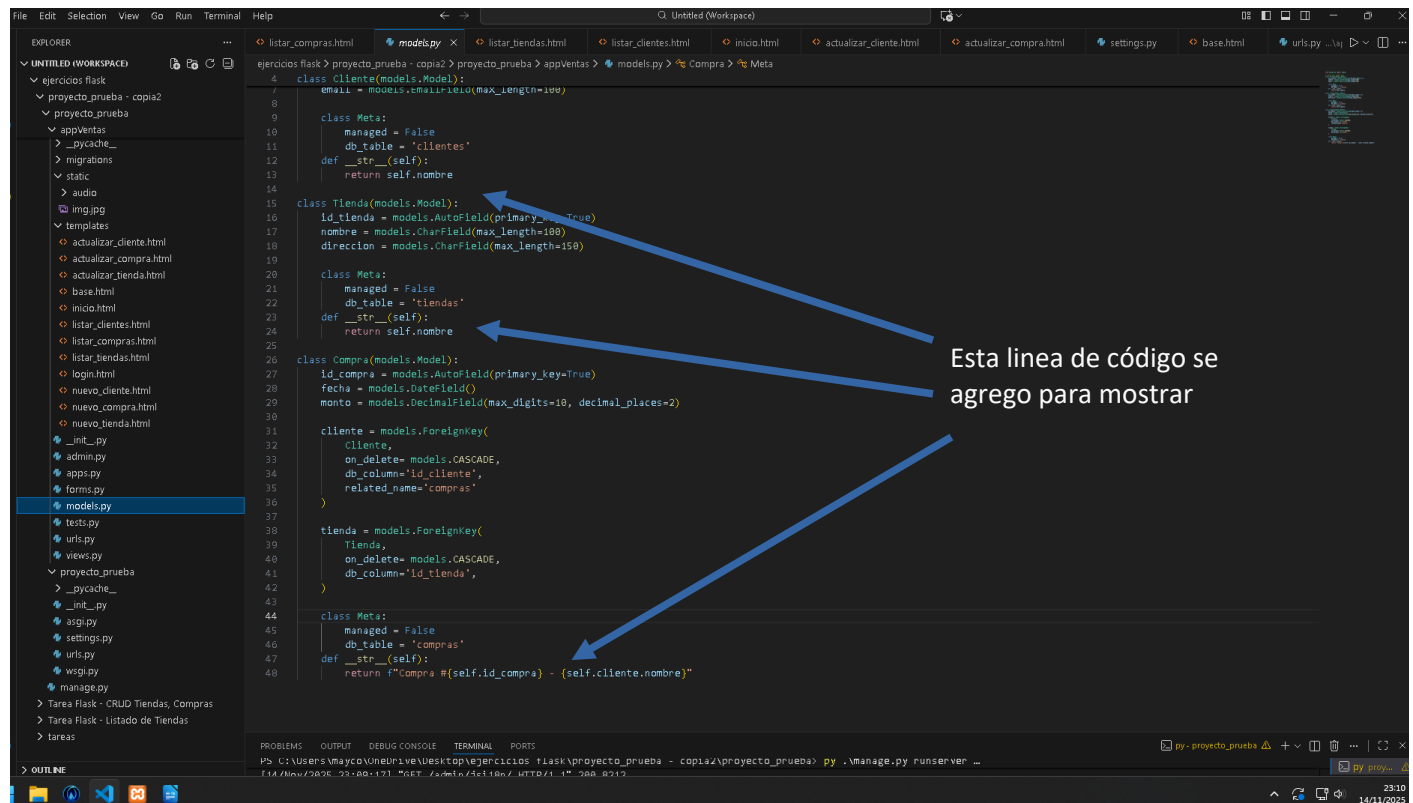
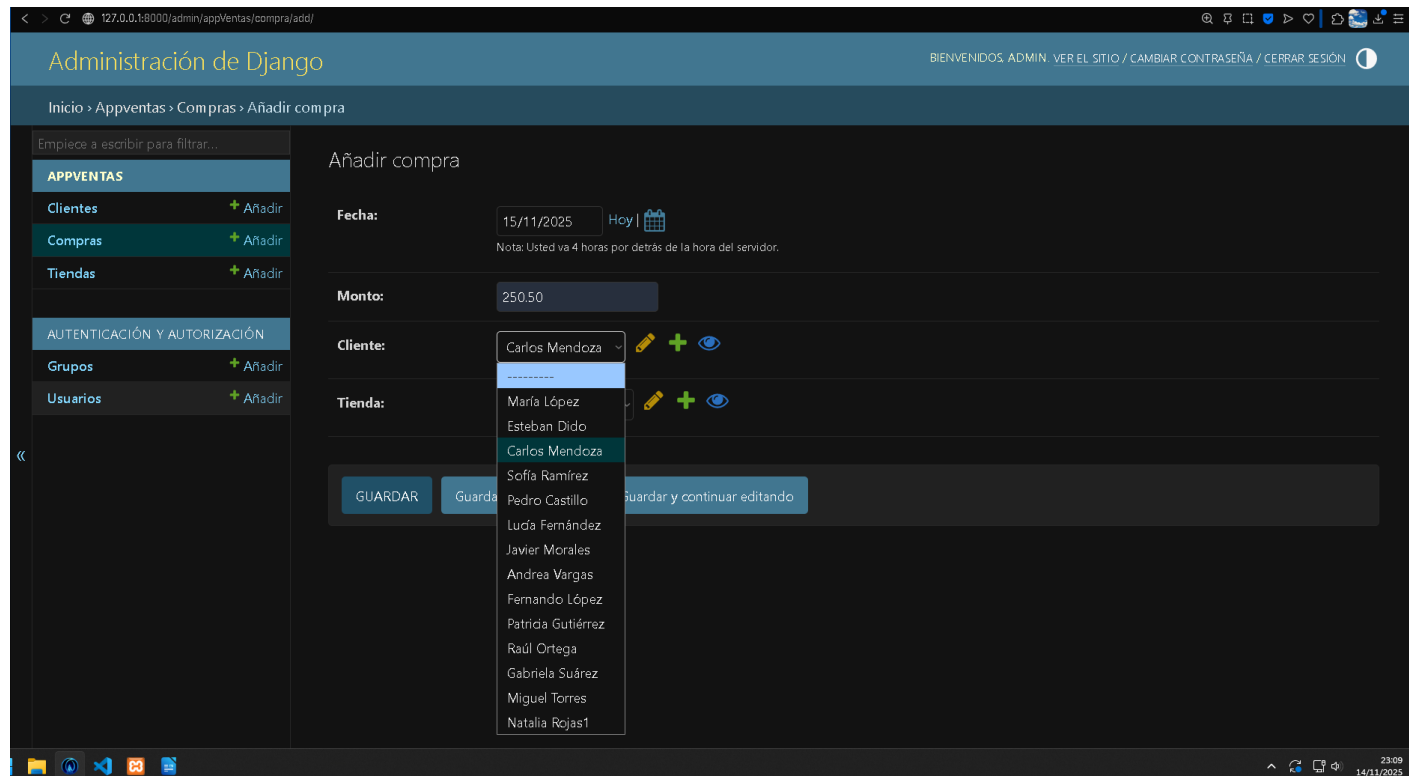
Nota: Usted va 4 horas por detrás de la hora del :

Monto:

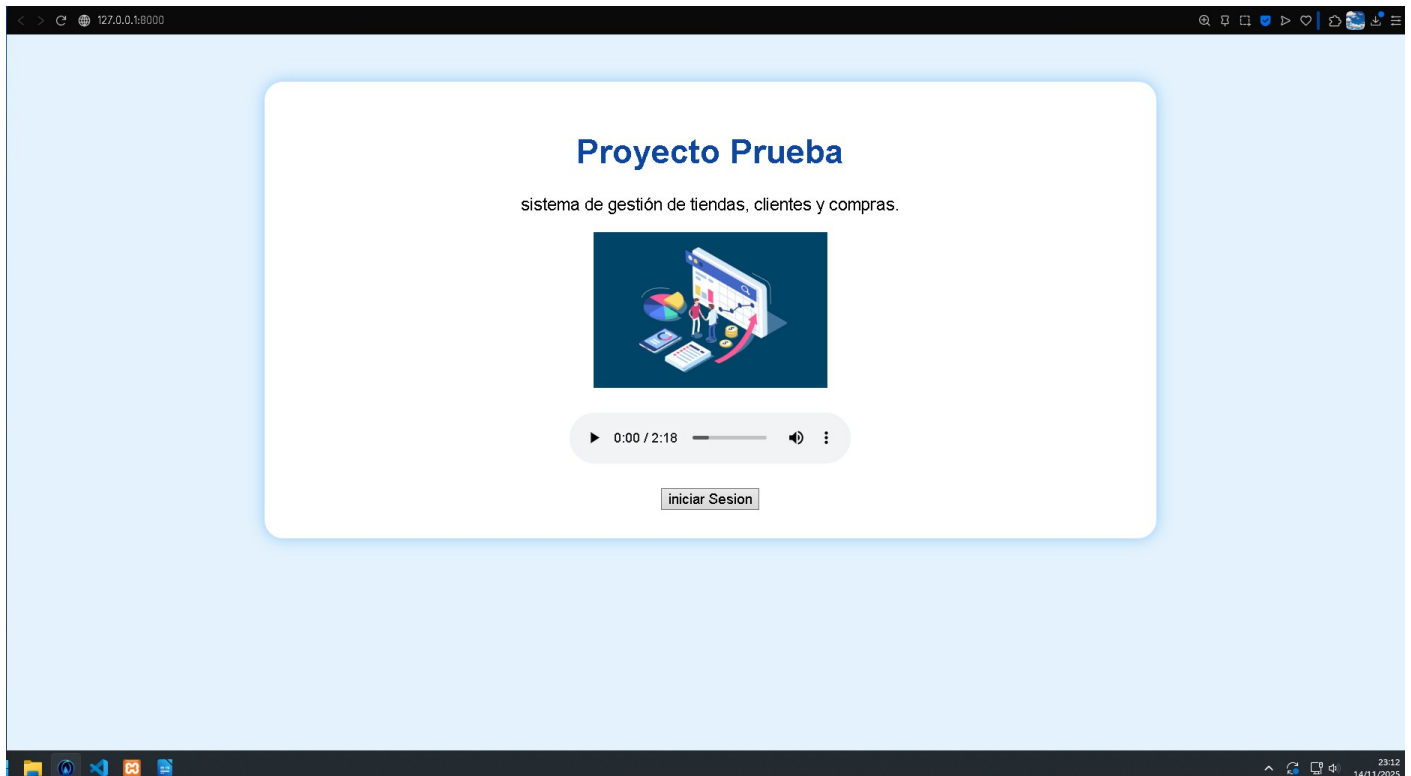
Cliente:

Tienda:

- Cliente object (2)
- Cliente object (3)
- Cliente object (4)
- Cliente object (5)



c. Implemente la pagina inicial del proyecto: diseñe su propia pagina inicial de presentación (imagen, colores y sonido)

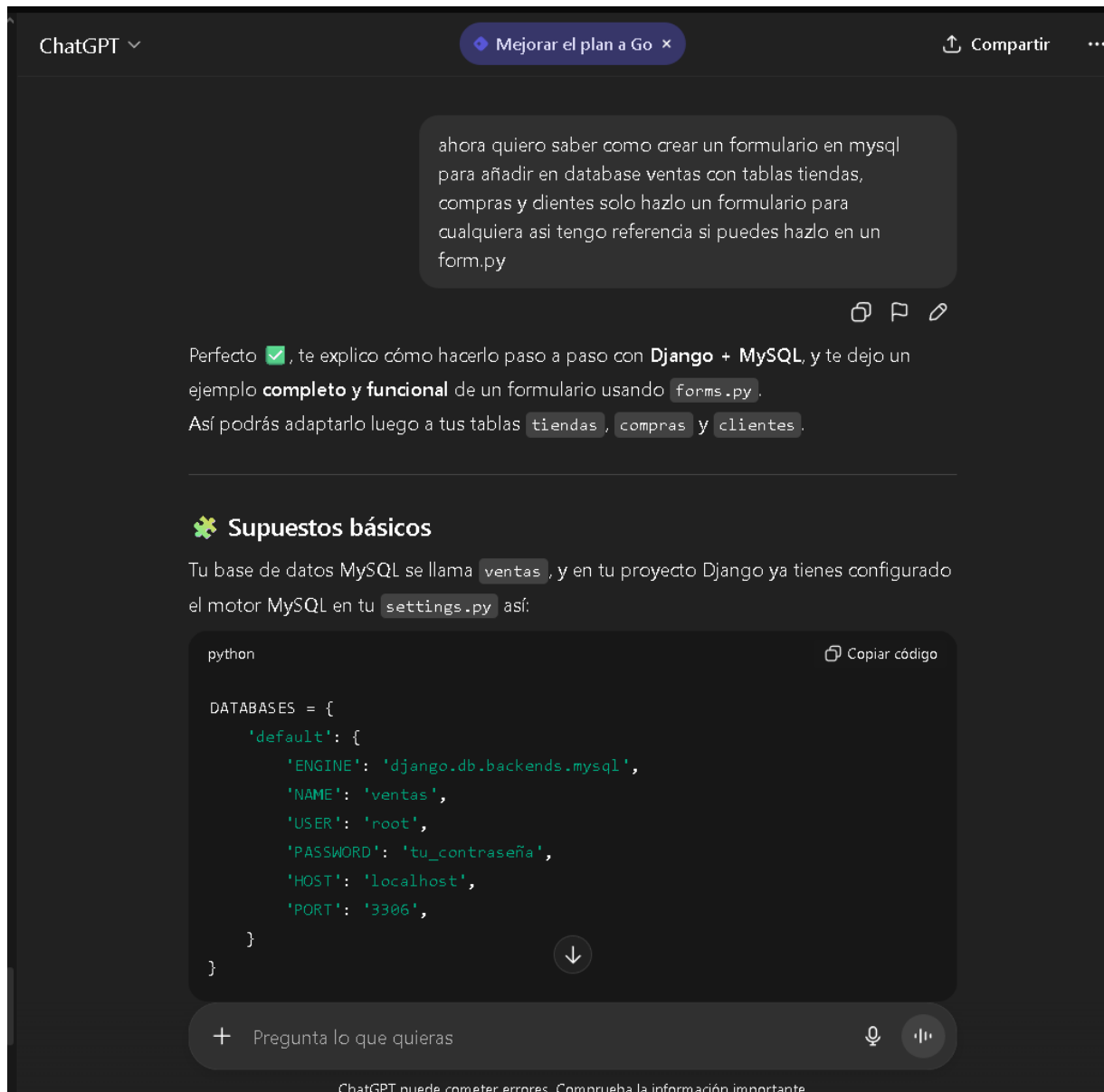


d. Aplicar authentication y cierre de sesión en el [proyecto prueba](#) para los usuarios creados en la clase el 1er usuario accede al CRUD, mientras el segundo usuario al RU







el prompt(s) utilizado para cumplir con cada inciso  
inciso a)




inciso b)




ChatGPT  [Mejorar el plan a Go](#)  [Compartir](#)


ahora cuando ingrese al py manage.py runserver entro en admin y quiero añadir compra me muestra cliente y tienda para seleccionar pero no muestra el nombre solo muestra cliente object(n) y tienda object(n) como lo cambio para que muestre el id o nombre

 **SOLUCIÓN: Agregar el método `__str__` en tus modelos**

En tu archivo:

 `appVentas/models.py`


Asegúrate de tener algo así:

 **Para el modelo Cliente**

python [Copiar código](#)


```
class Cliente(models.Model):
    nombre = models.CharField(max_length=100)
    # otros campos...

    def __str__(self):
        return self.nombre # ← aquí decides qué mostrar en el admin
```

 **Para el modelo Tienda**

python [Copiar código](#)

```
class Tienda(models.Model):
    nombre = models.CharField(max_length=100)
    direccion = models.CharField(max_length=200)
    # otros campos...
```



inciso c)

ChatGPT ▾

Mejorar el plan a Go ×

Compartir

c. Implemente la pagina inicial del proyecto: diseñe su propia pagina inicial de presentación (imagen, colores y sonido) esto como lo hago en ese proyecto que tenia de django luego por ultimo que es CRUD y RU

Perfecto, vamos por partes.

✓

## 1. ¿Cómo implementar la página inicial del proyecto Django?

Esta es la parte del enunciado:

**"Implemente la página inicial del proyecto: diseñe su propia página inicial de presentación (imagen, colores y sonido)"**

Lo más sencillo y correcto es crear una **vista**, una **URL**, y una **plantilla HTML** personalizada.

📌

### A. Crear la vista en `views.py`

Dentro de tu app (por ejemplo `appVentas`):

```
python

# appVentas/views.py
from django.shortcuts import render

def inicio(request):
    return render(request, 'inicio.html')
```

Copiar código

+

Pregunta lo que quieras


🎤

🔊

ChatGPT

Mejorar el plan a Go

Compartir

 **A. Crear la vista en `views.py`**


Dentro de tu app (por ejemplo `appVentas`):

```
python

# appVentas/views.py
from django.shortcuts import render

def inicio(request):
    return render(request, 'inicio.html')
```

Copiar código

 **B. Crear la URL de inicio**


En `proyecto_prueba/urls.py` agrega:

```
python



from django.contrib import admin
from django.urls import path
from appVentas.views import inicio

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', inicio, name='inicio'), # Página inicial
]
```

Copiar código

 **C. Crear la plantilla `inicio.html`**

+ Pregunta lo que quieras



inciso d)

ahora ya tengo eso me pide ahora esto d. Aplicar authentication y cierre de sesión en el proyecto prueba para los usuarios creados en la clase el 1er usuario accede al CRUD, mientras el segundo usuario al RU

Perfecto, ahora vamos a **agregar autenticación y control de permisos** para que:

- ✅ Usuario 1 → pueda usar CRUD completo (Crear, Leer, Actualizar, Eliminar)
- ❌ Usuario 2 → solo pueda usar RU (Leer y Actualizar — sin eliminar ni crear)

Esto se hace con **login + logout + permisos + decoradores**.


Te dejo todo listo paso por paso.

## ✅ 1. Crear LOGIN y LOGOUT en tu proyecto

### 1.1 Vista de Login

En `appVentas/views.py`:

python

 Copiar código

```
from django.contrib.auth import authenticate, login, logout
from django.shortcuts import render, redirect


def login_view(request):
    if request.method == "POST":
        username = request.POST["username"]
        password = request.POST["password"]
```

## ✓ 1. Crear LOGIN y LOGOUT en tu proyecto

### 1.1 Vista de Login

En `appVentas/views.py`:

python

 Copiar código


```
from django.contrib.auth import authenticate, login, logout
from django.shortcuts import render, redirect

def login_view(request):
```

## ✓ 2. Template para login

 `appVentas/templates/login.html`

html

 Copiar código

```
<h2>Iniciar sesión</h2>

{% if error %}
<p style="color:red">{{ error }}</p>
{% endif %}

<form method="POST">
    {% csrf_token %}
    <label>Usuario:</label>
    <input type="text" name="username"><br><br>

    <label>Contraseña:</label>
    <input type="password" name="password"><br><br>

    <button type="submit">Ingresar</button>
</form>
```

link: <https://github.com/oOmonstersOo/Tarea-Proyecto-Prueba.git>