//////////////////////////DATOS ADMINISTRADOR//////////////////////////////////////////////

usuario\_admin = Usuario.objects.create(

nombre="Gustavo Hernandez",

correo="gustavoalbertohm@gmail.com",

password="admin123",

rol=1

)

usuario = Usuario.objects.get(correo="gustavoalbertohm@gmail.com")

usuario.set\_password("admin123")

/////////////////////MODELO BASE DE DATOS USUARIO////////////////////////////////////

from django.db import models

class Usuario(models.Model):

ROLES = (

(1, 'Administrador'),

(2, 'Abogado'),

(3, 'Asistente Administrativo'),

)

id = models.BigAutoField(primary\_key=True)

nombre = models.CharField(max\_length=255)

correo = models.EmailField(unique=True)

password = models.CharField(max\_length=128)

rol = models.IntegerField(choices=ROLES)

class Meta:

db\_table = 'usuarios'

/////////////////////////CREAR USUARIO NORMAL////////////////////////////////////

usuario = Usuario.objects.create\_user(

correo="profesor@gmail.com ",

nombre="Manuel Carrasquero",

password="12345",

rol=2

)

usuario\_admin = Usuario.objects.create(

nombre="Gustavo Hernandez",

correo="gustavoalbertohm@gmail.com",

contrasena="admin123",

rol=1

)

print(usuario\_admin)

////////////////////////////////FUNCION CREAR USUARIOS////////////////////////////////

# Endpoint para crear usuario

@csrf\_exempt

def crear\_usuario(request):

if request.method == "POST":

try:

datos = json.loads(request.body)

nombre = datos.get('nombre')

email = datos.get('email')

rol = datos.get('rol')

if not nombre or not email or not rol:

return JsonResponse({"error": "Todos los campos son obligatorios."}, status=400)

# Crear usuario en la base de datos

usuario = Usuario.objects.create(

nombre=nombre,

email=email,

rol=rol,

password="098", # Contraseña predeterminada

fecha\_creacion=now()

)

# Opcional: enviar correo

enviar\_correo\_usuario(email, nombre)

# Devolver datos completos del usuario creado

return JsonResponse({

"id": usuario.id,

"nombre": usuario.nombre,

"email": usuario.email,

"rol": usuario.rol,

"fecha\_creacion": usuario.fecha\_creacion.strftime('%Y-%m-%d %H:%M:%S') # Formatear fecha

}, status=201)

except json.JSONDecodeError:

return JsonResponse({"error": "El cuerpo de la solicitud debe estar en formato JSON."}, status=400)

return JsonResponse({"error": "Método no permitido."}, status=405)

//////////////////////////////////////USUARIOFORM.JS//////////////////////////////////////////////////

 const agregarUsuario = (nuevoUsuario) => {

        setUsuarios([...usuarios, nuevoUsuario]);

    };