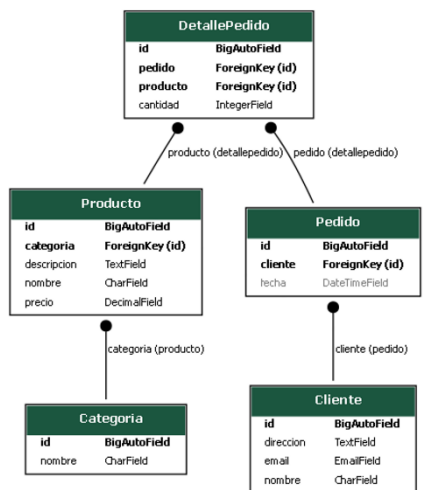


 Tema de la sesión:	Automatización Django (django-extensions + graphviz)	 Fecha:	22/10/25
¿Qué hicimos hoy?	Generar el diagrama ER desde Django.		
Hoy comencé con:	☐ Con curiosidad ☐ Con entusiasmo ☐ Sin ganas ☐ Con aburrimiento		
 Conceptos clave que aprendí	 Código clave		
• django-extensions: • pygraphviz: • graph_models: • -a: • -o mi_diagrama.png:	PASO 3: CREAR NUESTROS MODELOS. En models.py, escribe algo como esto, actualizar con tu modelo: <pre>from django.db import models # 🧑 Modelo Cliente class Cliente(models.Model): nombre = models.CharField(max_length=100) email = models.EmailField() direccion = models.TextField() def __str__(self): return self.nombre # 📁 Modelo Categoria class Categoria(models.Model): nombre = models.CharField(max_length=50) def __str__(self): return self.nombre # 📦 Modelo Producto class Producto(models.Model): nombre = models.CharField(max_length=100) precio = models.DecimalField(max_digits=10, decimal_places=2) descripcion = models.TextField() categoria = models.ForeignKey(Categoria, on_delete=models.CASCADE) def __str__(self): return self.nombre # 🛒 Modelo Pedido class Pedido(models.Model): fecha = models.DateTimeField(auto_now_add=True) cliente = models.ForeignKey(Cliente, on_delete=models.CASCADE) productos = models.ManyToManyField(Producto, through='DetallePedido') def __str__(self): return f"Pedido {self.id} - {self.cliente.nombre}"</pre>		
 Pasos que seguí	# 📄 Modelo DetallePedido (tabla intermedia) <pre>class DetallePedido(models.Model): pedido = models.ForeignKey(Pedido, on_delete=models.CASCADE) producto = models.ForeignKey(Producto, on_delete=models.CASCADE) cantidad = models.IntegerField(default=1) def __str__(self): return f"{self.cantidad} x {self.producto.nombre}"</pre>		
PASO 1: INSTALAR LO QUE NECESITAMOS INSTALAR GRAPHVIZ MANUALMENTE 1. Descarga Graphviz desde: https://graphviz.org/download/ 2. Instala el .msi para Windows - o Durante la instalación, MARCAR: "Add Graphviz to the system PATH for all users" 3. Reinicia la terminal después de instalar. 4. Abre tu terminal y Activa tu entorno virtual. <pre>pip install django-extensions pygraphviz o bien pip install django-extensions pip install pydotplus dot -V</pre> Deberías ver la versión de Graphviz.	 <pre> graph TD DetallePedido -- producto (detallepedido) --> Producto DetallePedido -- pedido (detallepedido) --> Pedido Producto -- categoria (producto) --> Categoria Pedido -- cliente (pedido) --> Cliente </pre>		
PASO 2: CONFIGURAR DJANGO En tu archivo settings.py, agrega esto: <pre># En INSTALLED_APPS, añade al final: INSTALLED_APPS = ['django.contrib.admin', 'django.contrib.auth', # ... tus otras apps ... 'django_extensions', # 🖐 ¡AÑADE ESTA LÍNEA!]</pre>			
PASO 4: GENERAR DIAGRAMAS Ejecutar en consola los códigos para generar los distintos diagramas.			

 Tema de la sesión:	Automatización Django (django-extensions + graphviz)		 Fecha:	22/10/25
¿Qué hicimos hoy?	Generar el diagrama ER desde Django.			
Hoy comencé con:	☐ Con curiosidad ☐ Con entusiasmo ☐ Sin ganas ☐ Con aburrimiento			
 Conceptos clave que aprendí	 Código clave	 Hoy aprendí que . . .		
Modelo de bases de datos: Diagrama Entidad – Relación: Django ORM: Herramientas:	PASO 4: GENERAR DIAGRAMAS Diagrama en .png: <pre>python manage.py graph_models -a -o mi_diagrama.png</pre> Diagrama en .PFD: <pre>python manage.py graph_models -a -o diagrama.pdf</pre> Diagrama preciso : <pre>python manage.py graph_models -a -o mis_tablas.png --include-models=Cliente,Categoria,Producto,Pedido,DetallePedido</pre> # PDF básico <pre>python manage.py graph_models -a -o mi_diagrama.pdf --include-models=Cliente,Categoria,Producto,Pedido,DetallePedido</pre>	# Desinstala pygraphviz <pre>pip uninstall pygraphviz</pre> # Instala estas alternativas <pre>pip install django-extensions pydotplus</pre> Como hacer que la computadora dibuje por nosotros		
		 Las dudas que me quedaron		
 Pasos que seguí	 Resultados /salida	Mi  en clase		
1. Instalar → pip install django-extensions pygraphviz 2. Configurar → Agregar a INSTALLED_APPS 3. Crear modelos → En models.py 4. Generar diagrama → python manage.py graph_models -a -o diagrama.png 5. ¡Listo! → Tienes tu diagrama ER automático. 6. Visualizarlos por fuer de VSCode .		Estado del aprendizaje	☐ 🌞 ¡Todo claro! ☐ 🌤️ Entendí algunas cosas. ☐ ☁️ Lluvia de dudas, ayuda. ☐ 🤖 Me voló la cabeza.	
		Contribuí a la clase con:	☐ pregunta, ☐ escuchando, ☐ ayudando a un compañero, ☐ compartiendo una idea.	
		Mi nivel de energía y atención fue:		
		☐ Alto ☐ Medio ☐ Bajo		