

 **Hecho:** Ver**Por hacer:** Hacer un envío

Ejercicio individual

Instrucciones

Realiza cada una de las siguientes actividades de manera individual. Asegúrate de documentar los comandos y pasos que sigas en cada ejercicio.

1. Recuperando Registros con Django ORM

- Crea un modelo llamado **Producto** con los siguientes campos:
 - **nombre** (CharField, máximo 100 caracteres)
 - **precio** (DecimalField, 5 dígitos, 2 decimales)
 - **disponible** (BooleanField)
- Usa el ORM de Django para recuperar todos los registros de la tabla **Producto**.

2. Aplicando Filtros en Recuperación de Registros

- Utiliza filtros en Django ORM para obtener:
 - Todos los productos con un precio mayor a 50.
 - Productos cuyo nombre empiece con la letra "A".
 - Productos disponibles.



3. Ejecutando Queries SQL desde Django

- Escribe una consulta SQL que obtenga los productos cuyo precio sea menor a 100.
- Ejecuta la consulta utilizando `raw()` en Django.

4. Mapeando Campos de Consultas al Modelo

- Ejecuta una consulta SQL personalizada con `raw()`.
- Asegúrate de mapear correctamente los resultados a una instancia del modelo **Producto**.

5. Realizando Búsquedas de Índice

- Investiga qué son los índices en bases de datos y su utilidad en Django.
- Crea un índice en el campo **nombre** del modelo **Producto**.
- Verifica el impacto en la eficiencia de búsqueda.

6. Exclusión de Campos del Modelo

- Recupera todos los productos pero excluyendo el campo **disponible**.
- Explica cómo Django maneja la omisión de ciertos campos en consultas.

7. Añadiendo Anotaciones en Consultas

- Usa `annotate()` para calcular un campo adicional llamado **precio_con_impuesto**, donde el impuesto sea del 16%.
- Muestra el resultado con el nuevo campo.

8. Pasando Parámetros a `raw()`

- Ejecuta una consulta con `raw()`, pero esta vez utilizando parámetros en lugar de valores fijos.
- Explica la diferencia y beneficios de esta técnica.

9. Ejecutando SQL Personalizado Directamente

- Usa `connection.cursor()` para ejecutar un SQL `INSERT`, `UPDATE` o `DELETE` directamente en la base de datos.
- Explica cuándo es recomendable usar esta técnica.

10. Conexiones y Cursores

- Crea una conexión manual a la base de datos en Django.
- Recupera datos usando un cursor y explica sus ventajas y desventajas frente al ORM.

11. Invocación a Procedimientos Almacenados

- Investiga qué son los procedimientos almacenados y cómo se usan en Django.
- Invoca un procedimiento almacenado desde Django usando `cursor.callproc()`.

Entrega

- Archivo comprimido zip o liga a repositorio Github con los archivos del proyecto.
- Ejecución: Individual
- Duración: 1 jornada de clases

Agregar entrega

Estado de la entrega

Estado de la entrega	Todavía no se han realizado envíos
Estado de la calificación	Sin calificar