

# Acceso a Datos

## Prueba

UD 05: Desarrollo de Aplicaciones con Bases de Datos Documentales

UD 06: Programación de Componentes de Acceso a Datos

## Tabla de contenidos

Instrucciones generales:.....	1
Criterios de evaluación:.....	2
Actividad única.....	3
Rúbrica.....	4
Traza de ejemplo.....	5

## Instrucciones generales:

- El alumnado deberá descargarse este enunciado de la Moodle durante los primeros 10 minutos de clase.
- Cada alumno **es responsable** de tener en su sistema un servidor de base de datos MongoDB, virtualenv , un intérprete de Python y todas las librerías pip necesarias para ejecutar las actividades de las UD5 y 6. Si un alumno no tiene listo esto antes de la prueba, no podrá descargar nada durante el transcurso de la misma, dado que está prohibido conectarse a Internet.
- Esta prueba consta de 1 actividad. El resultado de la actividad se subirá a Moodle. Debe ser un archivo con un programa en Python. El nombre del archivo debe ser “<nombre>\_<apellido1>\_<apellido2>\_prueba5y6.py”. Todo en minúsculas y sin tildes. Por ejemplo “amanda\_salas\_casas\_prueba5y6.py”
- Después de los 10 primeros minutos de clase, se dispone de 1h30min de tiempo para realizar la prueba. Dentro de este tiempo:
  - o El ordenador del alumnado permanecerá desconectado de Internet. Tanto por cable como de forma inalámbrica.
  - o Se deberá permanecer en absoluto silencio.
  - o No se podrá utilizar ningún dispositivo electrónico
  - o No se podrá consultar nada en Internet.
  - o Si algún/a alumn@ tiene una pregunta para el profesor, levantará la mano y esperará a ser atendido/a.
  - o Cuando el/la estudiante haya terminado su prueba, levantará la mano, pedirá permiso para conectar su ordenador a Internet y subirá a Moodle su prueba, siempre bajo la atenta mirada del profesor.
  - o **Cuando el/la estudiante haya realizado su entrega seguirán vigentes todas las normas descritas en estas instrucciones** (no usar dispositivos electrónicos, mantener silencio, etc....)

- A partir de la hora y 30 minutos, los alumnos que no hayan entregado la prueba dejarán de trabajar en la misma y pondrán su teclado encima del ordenador. A partir de este momento el alumno que desee entregar su prueba, lo hará copiando su **fichero py** en un pen drive que entregará el profesor. Este mecanismo se usará durante el tiempo restante hasta que termine la clase. El orden elegido al azar para entregar la prueba en pendrive será éste:

- |                                |                                    |
|--------------------------------|------------------------------------|
| 1. González Justo, Inés        | 11. Pedrera López, Ángela Kaijing  |
| 2. Rodríguez Cardenal, Carmen  | 12. Mejías De Matos, Francisco     |
| 3. Pineda Toro, Antonio        | 13. Rodríguez Gómez, Jesús Daniel  |
| 4. Alonso Fernández, Felipe    | 14. Alcaide Pérez, Mario           |
| 5. Bárcena López, Guillermo    | 15. López Navarro-Casas, Olalla    |
| 6. Solano Delgado, Isaac       | 16. González Rodríguez, Fernando   |
| 7. Rodríguez Segura, Pablo     | 17. Iglesias Fernández, José María |
| 8. Aguilar Castellanos, Raquel | 18. Ramos Navarro, Andrés          |
| 9. Bernáldez Carrasco, Manuel  | 19. Romero Iglesias, Manuel        |
| 10. Rodríguez Peña, Pablo      |                                    |

- El/la estudiante ha de tener en cuenta que el profesor ejecutará su script en su propio ordenador.
- El/la estudiante tendrá en cuenta que la **base de datos a usar se denominará 1dam**. El usuario para acceder a la base de datos se llama “usuario” y su clave será “usuario”. Si el alumno no tiene preparado esto antes de realizar la prueba será **responsabilidad suya** el perjuicio que le pueda acarrear durante la ejecución de la misma. Deberá llevar a cabo esta tarea durante los días anteriores a la prueba y no el mismo día en que tenga lugar.
- El/la estudiante deberá descargarse la documentación que considere oportuna para la realización de la prueba durante los días anteriores a la misma y no el mismo día de la prueba. Si el alumno no tiene preparado esto antes de realizar la prueba será **responsabilidad suya** el perjuicio que le pueda acarrear durante la ejecución de la misma.

### Criterios de evaluación:

- 5a) Se han valorado las ventajas e inconvenientes de utilizar bases de datos documentales nativas.
- 5b) Se ha establecido la conexión con la base de datos.
- 5c) Se han desarrollado aplicaciones que efectúan consultas sobre el contenido de la base de datos.
- 5d) Se han añadido y eliminado colecciones de la base de datos.
- 5e) Se han desarrollado aplicaciones para añadir, modificar y eliminar documentos de la base de datos.
- 6a) Se han valorado las ventajas e inconvenientes de utilizar programación orientada a componentes.
- 6b) Se han identificado herramientas de desarrollo de componentes.
- 6c) Se han programado componentes que gestionan información almacenada en ficheros.
- 6d) Se han programado componentes que gestionan mediante conectores información almacenada en bases de datos.
- 6e) Se han programado componentes que gestionan información usando mapeo objeto relacional.
- 6f) Se han programado componentes que gestionan información almacenada en bases de datos objeto relacionales y orientadas a objetos.
- 6g) Se han programado componentes que gestionan información almacenada en una base de datos documental nativa.

6h) Se han probado y documentado los componentes desarrollados.

6i) Se han integrado los componentes desarrollados en aplicaciones.

---

## Actividad única

La tienda de gadgets electrónicos "GadgetHub" ha decidido digitalizar su inventario utilizando MongoDB. En esta tienda, se venden productos innovadores como drones, auriculares, asistentes inteligentes y cámaras de acción. Como desarrollador, tu tarea es crear una solución en Python para gestionar el inventario de esta tienda.

**En esta actividad no es necesario hacer uso de transacciones.**

Debes desarrollar un componente llamado ProductoManager con los siguientes métodos:

```
def __init__(self, db_name="1dam", collection_name="productos"):
def insertar_productos(self, productos):
def consultar_proyeccion_ordenada(self, filtro, proyeccion, orden):
def mostrar_todos_productos(self):
def actualizar_productos(self, filtro, actualizacion):
def contar_documentos(self):
def eliminar_documentos(self, filtro):
def consulta_compleja(self, filtro, proyeccion, orden):
def cerrar_conexion(self):
```

También debes desarrollar el código que usa el componente. Dicho código hará uso del sistema de logging (por pantalla y a un fichero) para mostrar cualquier mensaje y servirá para llevar a cabo las siguientes tareas:

### Tarea 1: Inserción de Productos

**Base de datos:** La base de datos se llamará 1dam, y la colección será productos.

Productos a insertar:

```
{"nombre": "Drone Phantom X", "categoria": "Drones", "precio": 1200.50, "stock": 8},
{"nombre": "Auriculares Sonic Boom", "categoria": "Auriculares", "precio": 299.99, "stock": 15},
{"nombre": "Cámara Action Pro", "categoria": "Cámaras", "precio": 499.99, "stock": 10},
{"nombre": "Asistente SmartBuddy", "categoria": "Asistentes Inteligentes", "precio": 199.99, "stock": 20},
{"nombre": "Cargador Solar Ultra", "categoria": "Accesorios", "precio": 49.99, "stock": 3}
```

### Tarea 2: Lectura y Proyección

1. Consulta los productos de la categoría "Auriculares" y realiza una **proyección** para mostrar solo:
  - Nombre
  - Precio
  - Stock
2. Ordena los resultados por **precio en orden descendente**.

### Tarea 3: Actualización

Actualiza (con una sólo línea de código dentro del método “actualizar productos”) el precio a 1300 euros para los 2 siguientes productos:

1. **Drone Phantom X**
2. **Cámara Action Pro**

El filtro a usar ha de comprobar que el precio esté dentro de la lista ["Drone Phantom X", "Cámara Action Pro"]. Antes y después de la modificación has de mostrar todos los registros.

### Tarea 4: Eliminación

Elimina los productos que cumplen la condición: Stock menor a 5. Esto afectará a los productos Cargador Solar Ultra y Drone Phantom X. Antes y después de la eliminación, cuenta el número de productos en la colección.

### Tarea 5: Consulta compleja

Implementa una **consulta compleja** que combine condiciones:

- Busca todos los productos con un precio mayor a **300** y cuyo stock sea menor a **15**.
- Ordena los resultados por **categoría en orden alfabético** y muestra únicamente:
  - Nombre
  - Categoría
  - Precio

## Rúbrica

Tarea	Descripción	Puntos
<b>Definición de Componente ProductoManager</b>	Estructura del componente definida con métodos adecuados. Implementación de control de errores con try/except en todos los métodos. Uso correcto de logging para registrar operaciones importantes.	2
<b>Tarea 1: Inserción de Productos</b>	Inserción correcta de los 5 productos proporcionados.	0,5
<b>Tarea 2: Lectura y Proyección</b>	Consulta con proyección correcta para mostrar solo los campos solicitados.  Ordenamiento correcto por precio en orden descendente.  Si no usas el método consultar_proyeccion_ordenada() tal y como se ha definido arriba, este apartado valdrá la mitad (1 punto)	2
<b>Tarea 3: Actualización</b>	Actualización correcta de los 2 productos en una sola operación. Si no lo haces en una sólo operación el apartado valdrá la mitad (1 punto).  Visualización de los registros antes y después de la actualización.	2
<b>Tarea 4: Eliminación</b>	Eliminación correcta de productos con stock menor a 5.	1

	Conteo de registros antes y después de la eliminación.	
<b>Tarea 5: Consulta Compleja</b>	Implementación correcta de la consulta con condiciones combinadas.  Proyección y ordenamiento por categoría aplicados correctamente. Si no usas el método <code>consulta_compleja()</code> tal y como se ha definido arriba, este apartado valdrá la mitad (0,75 puntos)	1,5
<b>Código de uso del componente</b>	Uso del componente  Todos los mensajes deben mostrarse a través del sistema de logging.	1

## Traza de ejemplo

A modo de guía y sin que sea obligatorio que tus mensajes sean como los que se muestran a continuación, se muestra una traza de ejemplo de la ejecución del programa:

```

2025-01-06 12:00:00 - INFO - Conexión a la base de datos establecida correctamente.
2025-01-06 12:00:02 - INFO - Se han insertado 5 productos.
2025-01-06 12:00:03 - INFO - Consulta con proyección y ordenamiento realizada.
2025-01-06 12:00:03 - INFO - {'nombre': 'Auriculares Sonic Boom', 'precio': 299.99, 'stock': 15}
2025-01-06 12:00:04 - INFO - Registros antes de la actualización:
2025-01-06 12:00:04 - INFO - {'nombre': 'Drone Phantom X', 'categoria': 'Drones', 'precio': 1200.5, 'stock': 8}
2025-01-06 12:00:04 - INFO - {'nombre': 'Auriculares Sonic Boom', 'categoria': 'Auriculares', 'precio': 299.99, 'stock': 15}
2025-01-06 12:00:04 - INFO - {'nombre': 'Cámara Action Pro', 'categoria': 'Cámaras', 'precio': 499.99, 'stock': 10}
2025-01-06 12:00:04 - INFO - {'nombre': 'Asistente SmartBuddy', 'categoria': 'Asistentes Inteligentes', 'precio': 199.99, 'stock': 20}
2025-01-06 12:00:04 - INFO - {'nombre': 'Cargador Solar Ultra', 'categoria': 'Accesorios', 'precio': 49.99, 'stock': 3}
2025-01-06 12:00:05 - INFO - Se han actualizado 2 documentos.
2025-01-06 12:00:05 - INFO - Registros después de la actualización:
2025-01-06 12:00:05 - INFO - {'nombre': 'Drone Phantom X', 'categoria': 'Drones', 'precio': 1300.0, 'stock': 8}
2025-01-06 12:00:05 - INFO - {'nombre': 'Auriculares Sonic Boom', 'categoria': 'Auriculares', 'precio': 299.99, 'stock': 15}
2025-01-06 12:00:05 - INFO - {'nombre': 'Cámara Action Pro', 'categoria': 'Cámaras', 'precio': 1300.0, 'stock': 10}
2025-01-06 12:00:05 - INFO - {'nombre': 'Asistente SmartBuddy', 'categoria': 'Asistentes Inteligentes', 'precio': 199.99, 'stock': 20}
2025-01-06 12:00:05 - INFO - {'nombre': 'Cargador Solar Ultra', 'categoria': 'Accesorios', 'precio': 49.99, 'stock': 3}
2025-01-06 12:00:06 - INFO - Total de documentos en la colección: 5
2025-01-06 12:00:06 - INFO - Se han eliminado 2 documentos.
2025-01-06 12:00:07 - INFO - Total de documentos en la colección: 3

```

2025-01-06 12:00:07 - INFO - Consulta compleja realizada.

2025-01-06 12:00:07 - INFO - {'nombre': 'Cámara Action Pro', 'categoria': 'Cámaras', 'precio': 1300.0}

2025-01-06 12:00:07 - INFO - {'nombre': 'Drone Phantom X', 'categoria': 'Drones', 'precio': 1300.0}

2025-01-06 12:00:08 - INFO - Conexión cerrada.