

# **MongoDB con Python**

Acceso a MongoDB con Python	AI y Big Data – Sistemas de Big Data
-----------------------------	--------------------------------------

---

## Índice

Tarea 1 Comprobaciones previas (0,5 puntos)	4
Tarea 2 Inserciones de registros (0,5 puntos)	4
Tarea 3 Consultas (3,5 puntos)	5
Tarea 4 Actualizaciones (2 puntos)	6
Tarea 5 Consulta avanzada (1,5 puntos)	7
Tarea 6 Interacción con un Dataframe (2 puntos)	7

## Descripción

1. Debes realizar las tareas que se detallan a continuación en un **Notebook de Jupyter**.
2. Utiliza los **markdowns**, creando una estructura que permita seguir correctamente la ejecución. **Comenta también las instrucciones de código** para aportar mayor claridad.
3. Para cada tarea lee los detalles y los elementos a entregar antes de ejecutar la tarea.
4. Carga únicamente las librerías que vayas a utilizar.
5. Asegúrate de ejecutar todas las celdas. Una opción para verificar que no te olvidas de ninguna es ir a *Kernel* → *Restart Kernel and Run All Cells...*
6. No te olvides de incluir comentarios en el código.
7. **Entrega** tanto el notebook en **formato .ipynb** como exportado a **formato .pdf**.
8. Dispones en el aula virtual del fichero *ventas.json*. Es el mismo fichero que utilizaste en la actividad anterior. Debes cargarlo en la bbdd supermercado como la colección **pedidos**. No es necesario que esté incluido en el fichero jupyterlab, con una captura del comando en consola es suficiente.
9. El resto de operaciones, que son las solicitadas en las tareas, debes implementarlas *exclusivamente* mediante código Python en el notebook que entregarás.

## Tarea 1 Comprobaciones previas (0,5 puntos)

Muestra el número de documentos que se almacenan en la colección ***pedidos***

## Tarea 2 Inserciones de registros (0,5 puntos)

**Inserta tres documentos** en la colección utilizando un único comando.

El primero tendrá como *ID\_Pedido* el 79159 e *Importe total* 3.45

El segundo tendrá el mismo *ID\_Pedido* (79159), *Importe total* de 0.23 y *Número de items* 5.

El tercero tendrá el mismo *ID\_Pedido* (79159) y *Número de items* 8.

Verifica que se han insertado los tres documentos.

## Tarea 3 Consultas (3,5 puntos)

Debes realizar las siguientes consultas:

1. **Dirección asociada** al pedido 34789. Solo nos interesa recuperar la dirección (con todos sus campos)
2. **Número de pedidos que NO** disponen de dirección del cliente
3. Datos de cliente con el **pedido de menor *Importe total***. Se debe mostrar:
  - *ID\_Pedido*
  - *Importe total*
  - *Todos los datos del cliente: ID\_Cliente, Nombre, Apellido1 y Apellido2 y su dirección asociada (calle, número, ...)*
4. **Recupera aquellos pedidos** (y solo el ID\_Pedido) cuyo Importe total está entre 2 y 2.1 euros.
5. **Pedidos** que, entre sus ítems, figuran *Pan* y *Jabón*. Devuelve solo *ID\_Pedido*.

## Tarea 4 Actualizaciones (2 puntos)

Te has dado cuenta de que al sumar el número de ítems en los pedidos, la persona encargada de recoger los datos tuvo un error, y dejó de incluir un elemento que todos tenían en común: la bolsa.

Debes realizar la siguiente actualización solo sobre aquellos pedidos que tienen *Número de ítems*:

Incrementar en una unidad el valor actual de *Número de ítems*

Añadir a la lista *Items* un nuevo documento:

- Nombre: "Bolsa"
- Importe unitario: 0.05
- Cantidad: 1

Además de realizar la actualización, muestra el número de documentos que han sido actualizados.

## Tarea 5 Interacción con un Dataframe (2 puntos)

Realiza las siguientes operaciones:

Debes cargar todos los documentos existentes en la colección *pedidos* en un **dataframe** llamado *df\_pedidos*.

El dataframe debe utilizar como índice el campo *ID\_Pedido* de la colección.

A continuación, inserta en el Dataframe una nueva columna llamada **Importe medio**, que sea resultado de **dividir el Importe total por el Número de ítems**.

Guarda el **dataframe** resultante, con la nueva columna, en una colección de la bbdd *supermercado* llamada **resultadofinal**.

En la colección resultante **no debe haber** documentos **con campos NaN**