



---

# UD8: BASES DE DATOS NOSQL

---

## Pràcticas no evaluables Boletín C

---

### Bases de Datos (BD) CFGs DAM/DAW

Abelardo Martínez y Pau Miñana.  
Curso 2023-2024

---

# Aspectos a tener en cuenta

---

Estas actividades son opcionales y no evaluables pero es recomendable hacerlas para un mejor aprendizaje de la asignatura.

🚫 **Si buscas las soluciones por Internet o preguntas al oráculo de ChatGPT, te estarás engañando a ti mismo.** Ten en cuenta que ChatGPT no es infalible ni todopoderoso. Es una gran herramienta para agilizar el trabajo una vez se domina una materia, pero usarlo como atajo en el momento de adquirir habilidades y conocimientos básicos perjudica gravemente tu aprendizaje.

Si lo utilizas para obtener soluciones o asesoramiento respecto a las tuyas, revisa cuidadosamente las soluciones propuestas igualmente. Intenta resolver las actividades utilizando los recursos que hemos visto y la documentación extendida que encontrarás en el "Aula Virtual".

## 📖 Recomendaciones:

- **No uses NUNCA tildes, ni eñes, ni espacios, ni caracteres no alfanuméricos** (salvo el guión bajo) **en los metadatos** (nombres de elementos de una base de datos) y **usa letras minúsculas**.
- Sé coherente con el uso de mayúsculas/minúsculas.

# ÍNDICE

## 1. Base Imperial

1.1. Script 1: Contar pilotos

1.2. Script 2: Datos misiones con coste mayor que 5

1.3. Script 3: Datos de las misiones con coste total mayor que 20

1.4. Script 4: Costes por piloto

1.5. Script 5: Piloto que más costes ha generado

1.6. Script 6: Abatidos

1.7. Script 7: Abatidos por tipo nave

1.8. Script 8: Costes por tipo nave

1.9. Script 9: Tipo de nave con más costes

1.10. Script 10: Misiones en planetas boscosos

## 1. Base Imperial

---

Vamos a modelar los pilotos y misiones de una base imperial, teniendo en cuenta el número de naves enemigas abatidas.

Para ello, necesitamos una base de datos en MongoDB llamada *imperialbaseMDB*, con una colección llamada *pilotos* que incluya los siguientes campos con los siguientes valores:

piloto_id	nombre	apellidos	mision	abatidos
pr01	Lord	Darth Vader	ARRAY	10
pr02	Moff	Tarkin		
pr03	Almirante	Thrawn	ARRAY	
pr04	Biggs	Darklighter	ARRAY	5
pr05	Barón	Soontir Fel	ARRAY	2
pr06	Maarek	Stele	ARRAY	3
pr07	Mulchive	Wermis	ARRAY	6
pr08	Loka	Hask	ARRAY	7
pr09	Tycho	Celchu	ARRAY	4
pr10	PT001	Piloto DS-61-1		3
pr11	Kylo	Ren		0

El array de *mission* contendrá documentos con las misiones realizadas por cada piloto, la nave usada y su coste:

idpiloto	cod	nave	coste
pr01			
	MA001	Crucero imperial nodriza	5
	MI003	Crucero imperial nodriza	7.16
	ME004	Tiefighter Mustafar	2.33
pr03			
	MT002	Crucero logístico	6.58
pr04			
	MI003	Tiefighter	10.24
	ME004	Crucero de transporte	7.89
pr05			
	MA001	Tiefighter	3.25
	ME004	Crucero de transporte	1.76
pr06			
	MA001	Tiefighter	5.01
	ME004	Crucero de transporte	2.43
pr07			
	MA001	Tiefighter	4
	ME004	Crucero de transporte	3.29
pr08			
	MA001	Tiefighter	3.29
	ME004	Crucero de transporte	4.05
pr09			
	MA001	Tiefighter	2.57
	ME004	Crucero de transporte	1.65

El script para crear esa base de datos es el siguiente:

```
use imperialbaseMDB
db.createCollection("pilotos")

var p1 = {_id: "pr01", nombre: "Lord", apellidos: "Darth Vader",
  mision:[
    {cod:"MA001", nave: "Crucero imperial nodriza", coste: 5},
    {cod:"MI003", nave: "Crucero imperial nodriza", coste: 7.16},
    {cod:"ME004", nave: "Tiefighter Mustafar", coste: 2.33}],
  abatidos: 10}
var p2 = {_id: "pr02", nombre: "Moff", apellidos: "Tarkin"}
var p3 = {_id: "pr03", nombre: "Almirante", apellidos: "Thrawn",
  mision:{cod:"MT002", nave: "Crucero logístico", coste:6.58}}
var p4 = {_id: "pr04", nombre: "Biggs", apellidos: "Darklighter",
  mision:[
    {cod:"MI003", nave: "Tiefighter", coste: 10.24},
    {cod:"ME004", nave: "Crucero de transporte", coste: 7.89}],
  abatidos: 5}
var p5 = {_id: "pr05", nombre: "Barón", apellidos: "Soontir Fel",
  mision:[
    {cod:"MA001", nave: "Tiefighter", coste: 3.25},
    {cod:"ME004", nave: "Crucero de transporte", coste: 1.76}],
  abatidos: 2}
var p6 = {_id: "pr06", nombre: "Maarek", apellidos: "Stele",
  mision:[
    {cod:"MA001", nave: "Tiefighter", coste: 5.01},
    {cod:"ME004", nave: "Crucero de transporte", coste: 2.43}],
  abatidos: 3}
var p7 = {_id: "pr07", nombre: "Mulchive", apellidos: "Wermis",
  mision:[
    {cod:"MA001", nave: "Tiefighter", coste: 4},
    {cod:"ME004", nave: "Crucero de transporte", coste: 3}],
  abatidos: 6}
var p8 = {_id: "pr08", nombre: "Loka", apellidos: "Hask",
  mision:[
    {cod:"MA001", nave: "Tiefighter", coste: 3.29},
    {cod:"ME004", nave: "Crucero de transporte", coste: 4.05}],
  abatidos: 7}
var p9 = {_id: "pr09", nombre: "Tycho", apellidos: "Celchu",
  mision:[
    {cod:"MA001", nave: "Tiefighter", coste: 2.57},
    {cod:"ME004", nave: "Crucero de transporte", coste: 1.65}],
  abatidos: 4}
var p10 = {_id: "pr10", nombre: "PT001", apellidos: "DS-61-1",
  abatidos: 3}

db.pilotos.insertMany([p1, p2, p3, p4, p5, p6, p7, p8, p9, p10])
```

Crea los scripts necesarios para realizar las siguientes tareas:

### 1.1. Script 1: Contar pilotos

Se requiere contar el número de pilotos que tienen *abatidos* y, o bien su *nave* es "Tiefighter Mustafar" o han realizado alguna misión.

### 1.2. Script 2: Datos misiones con coste mayor que 5

Se requiere listar los datos de las misiones (*nombre*, *apellidos* del piloto, *cod*, *nave* y *coste*) con *coste* superior a 5, ordenados por nombre y apellidos del piloto. Para los costes se consideran las misiones individualmente por cada piloto, no el total

### 1.3. Script 3: Datos de las misiones con coste total mayor que 20

Se requiere listar las misiones(*cod*) y el coste total que cada una ha tenido, para las misiones con coste total mayor que 20, ordenadas por coste total de mayor a menor.

### 1.4. Script 4: Costes por piloto

Se requiere listar los pilotos (*nombre* y *apellidos*) y los costes totales que han generado entre todas sus misiones, ordenados de menor a mayor coste.

### 1.5. Script 5: Piloto que más costes ha generado

Se requiere listar el *nombre*, *apellidos* y coste total del piloto que más ha costado, considerando todas las misiones.

### 1.6. Script 6: Abatidos

a) Se requiere contar el número total de *abatidos* por todas las tropas imperiales b) Se requiere el *nombre*, *apellidos* y *abatidos* del piloto que más naves enemigas ha abatido.

### 1.7. Script 7: Abatidos por tipo nave

Se requiere listar el nombre de cada tipo de *nave* y el total de *abatidos* que se han conseguido con la misma, independientemente del piloto/misión. ¿Qué error contiene esta consulta en su planteamiento?

### 1.8. Script 8: Costes por tipo nave

Se requiere listar el nombre de cada tipo de *nave* y el total de costes generados con la misma, independientemente del piloto/misión.

## 1.9. Script 9: Tipo de nave con más costes

Se requiere listar el nombre del tipo de *nave* que ha generado más costes entre todas sus misiones.

## 1.10. Script 10: Misiones en planetas boscosos

Considerando la siguiente colección de planetas:

```
use imperialbaseMDB
db.createCollection("planetas")

var j_plan1 = {_id: "PL01", nombre: "Endor", tipo: "boscoso",
  misiones: ["MA001"]}
var j_plan2 = {_id: "PL02", nombre: "Tatooine", tipo: "desértico"}
var j_plan3 = {_id: "PL03", nombre: "Yavin IV", tipo: "boscoso",
  misiones: ["MI003", "ME004"]}
var j_plan5 = {_id: "PL05", nombre: "Wayland", tipo: "montañoso",
  misiones: ["MT002"]}

db.planetas.insertMany([j_plan1, j_plan2, j_plan3, j_plan5])
```

Se requiere listar el *nombre,apellidos* y *cod* de misión de los pilotos que han realizado misiones en planetas boscosos, así como el nombre del *planeta*.