

# ACTIVIDAD EVALUABLE 4

## AE4

**BASES DE DATOS 22/23**  
CFGS DAM

## BASES DE DATOS NO RELACIONALES: MONGODB

### **Autores:**

Abelardo Martínez

Licencia Creative Commons



**Reconocimiento - NoComercial - CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

## 1. DESCRIPCIÓN

- Se pide diseñar en MongoDB los scripts necesario para cada caso.

Entrega **solo** las consultas referentes al bloque que corresponda con la última cifra de tu DNI/NIE  
**0-4 => MODELO B      5-9 => MODELO A**

## 2. PLAZO DE ENTREGA Y PORCENTAJES

- **Porcentajes en la EVALUACIÓN:** 30% de la nota total es para las evaluables
- **Porcentajes de la ACTIVIDAD:** 50% de las evaluables (hay dos por evaluación)
- **Plazo de entrega:** 23:59 del 7 de mayo de 2023 (4 semanas)

## 3. CALIFICACIÓN

- La entrega no es obligatoria ni hay nota mínima. Cada parte se calificará de 0 a 10 según las indicaciones de este documento.

## 4. RECURSOS

- Debes estudiar todos los materiales que te hemos proporcionado, prestando especial atención a los boletines de ejercicios.
- **Sigue los pasos indicados en la plantilla que te proporcionamos para facilitar la corrección de la actividad. La plantilla es idéntica para los 2 modelos.**
- En los scripts puedes usar variables javascript o no, como prefieras.

## 5. PLAGIO

- Tarea INDIVIDUAL. En caso de sospecha de originalidad será requerida una entrevista oral.

## 6. INSTRUCCIONES DE ENTREGA

- La tarea se entregará en un único PDF que contendrá los ejercicios solicitados.

## 7. SOLUCIONES Y RESULTADOS

- Recibirás la calificación desglosada por cada criterio, y el total, junto con cualquier comentario que brinde sugerencias sobre cómo podrías haberlo hecho mejor.

## PASOS A SEGUIR

1. Descarga del Aula Virtual la plantilla ÚNICA.
2. Recomendaciones:
  - Puedes usar las variables javascript que consideres para hacer más legible la solución.
  - Asegúrate de que tu script devuelve **únicamente** los registros y campos solicitados:
    - **90% de la nota de cada script: SOLUCIÓN CORRECTA (devuelve solo lo que se pide)**
  - Asegúrate de que tu script está en el formato indicado, siendo lo más estructurado y legible posible:
    - **10% de la nota de cada script: FORMATO DEL SCRIPT**

## MODELO A: BASE IMPERIAL

### SCRIPT 1: CREAR Y POBLAR LA BASE DE DATOS (2 PUNTOS)

Vamos a modelar los transportes y misiones de pilotos 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:

_id	nombre	apellidos	transporte	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 Transporte contendrá documentos con los pilotos, transportes realizados y su coste por transporte:

idpiloto	transporte	nave	coste		idpiloto	transporte	nave	coste
pr01					pr06			
	MA001	Crucero imperial nodriza	5			MA001	Tiefighter	5.01
	MI003	Crucero imperial nodriza	7.16			ME004	Crucero de transporte	2.43
	ME004	Tiefighter Mustafar	2.33		pr07			
pr03						MA001	Tiefighter	4
	MT002	Crucero logístico	6.58			ME004	Crucero de transporte	3
pr04					pr08			
	MI003	Tiefighter	10.24			MA001	Tiefighter	3.29
	ME004	Crucero de transporte	7.89			ME004	Crucero de transporte	4.05
pr05					pr09			
	MA001	Tiefighter	3.25			MA001	Tiefighter	2.57
	ME004	Crucero de transporte	1.76			ME004	Crucero de transporte	1.65

Crea el script (en javascript) necesario para (1º) borrar la base de datos, (2º) crear la base de datos, (3º) crear la colección indicada y (4º) insertar todos los datos, teniendo en cuenta que los campos sin datos no deben aparecer en ese elemento de la colección y el tipo de datos debe ser el más adecuado. Una vez insertados, deberás (5º) listar TODOS los campos de todos los registros, ordenados por apellidos y nombre.

### SCRIPT 2: ACTUALIZAR LA INFORMACIÓN (2 PUNTOS)

Se requiere incluir un nuevo campo llamado “**categoría**” para catalogar la destreza de cada piloto.

- Para los pilotos con más de 5 abatimientos (>5) escribirá en el campo **categoría**:
  - “Barón rojo”
- Para los pilotos que no tienen abatidos (falta el campo o es igual a cero) escribirá en **categoría**:
  - “Incompetente”
- Para los pilotos con menos de 3 abatimientos (<3) y mayor que cero, escribirá en el campo **categoría**:
  - “Suficiente”

Crea el script en javascript necesario para (1º) actualizar el campo “categoría” en cada uno de los casos indicados y (2º) listar SOLO apellidos, nombre y categoría de TODOS los documentos ordenados por apellidos y nombre.

### SCRIPT 3: MINERÍA DE DATOS (3 PUNTOS)

a) Se requiere listar los datos de los transportes (nombre y apellidos del piloto, transporte, nave y coste) con coste superior a 5, ordenados por nombre y apellidos del piloto. Crea el script necesario.

b) Se requiere listar el nombre, apellidos y coste total del piloto que más ha costado en todas los transportes. Crea el script necesario.

### SCRIPT 4: CONTADORES (2 PUNTOS)

Se requiere (1º) contar el número de pilotos que tienen abatidos y, o bien su nave es “Tiefighter Mustafar” o han realizado algún transporte. Crea el script necesario.

### SCRIPT 5: BORRADO (1 PUNTO)

Se requiere (1º) listar los nombres y apellidos de los pilotos a los que, o bien tienen 0 abatidos o les falta el campo abatido y luego (2º) borrarlos. Crea el script necesario.

## MODELO B: ACADEMIA JEDI

### SCRIPT 1: CREAR Y POBLAR LA BASE DE DATOS (2 PUNTOS)

Vamos a modelar los cursos y alumnos de una academia Jedi, teniendo en cuenta las notas obtenidas.

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

_id	nombre	apellidos	curso	asignatura	notat	notap
es01	Qui-Gon	Jinn	Curso maestro Jedi	ARRAY	9.5	10
es02	Obi-Wan	Kenobi	Curso maestro Jedi	ARRAY	9.7	9.8
es03	Anakin	Skywalker	Curso básico padawan nivel 1		10	10
es04	Ahsoka	Tano	Curso avanzado padawan nivel 2		9.75	9.95
es05	Luke	Skywalker	Curso avanzado padawan nivel 2	ARRAY	9.5	9.85
es06	Rey	Skywalker	Curso básico padawan nivel 1	ARRAY	4	
es07	Baby	Yoda	Curso básico padawan nivel 1	ARRAY	8.5	
es08	Tallisibeth	Enwandung-Esterhazy	Curso avanzado padawan nivel 2	ARRAY	4.5	9.87
es09	Chirrut	Îmwe	Curso básico padawan nivel 2	ARRAY	3	4
es10	Sta-Den	Eekin	Curso maestro Jedi		9.95	10
es11	Kit	Fisto	Curso maestro Jedi		9.9	10

El Array de Asignatura contendrá documentos con los alumnos, asignaturas realizadas, sus maestros y coste por asignatura:

idalumno	asignatura	maestro	coste		idalumno	asignatura	maestro	coste
es01					es06			
	Telequinesis	Yoda	1.25			Fundamentos Jedi	Luke Skywalker	2.01
	Lucha con sable láser	Yoda	4.35			Lucha con sable láser	Luke Skywalker	8.43
	Idiomas	Kit Fisto	0.5		es07			
es02						Telequinesis	Luke Skywalker	4
	Telequinesis	Qui-Gon Jinn	2			Control de la fuerza	Luke Skywalker	3
	Lucha con sable láser	Qui-Gon Jinn	5.87		es08			
	Idiomas	Mace Windu	0.74			Telequinesis	Kit Fisto	1.29
	Control de la fuerza	Yoda	2.33			Lucha con sable láser	Kit Fisto	4.05
es05					es09			
	Telequinesis	Yoda	1.05			Adaptación Jedi	Sta-Den Eekin	2.57
	Lucha con sable láser	Obi-Wan Kenobi	7.76			Lucha con palo Kendo	Sta-Den Eekin	9.65

Crea el script en javascript necesario para (1º) borrar la base de datos, (2º) crear la base de datos, (3º) crear la colección indicada y (4º) insertar todos los datos, teniendo en cuenta que los campos sin datos no deben aparecer en ese elemento de la colección y el tipo de datos debe ser el más adecuado. Una vez insertados, deberás (5º) listar TODOS los campos de todos los registros, ordenados por apellidos y nombre.

### SCRIPT 2: ACTUALIZAR LA INFORMACIÓN (2 PUNTOS)

Se requiere incluir un nuevo campo llamado “**observaciones**” para anotar comentarios sobre cada alumno.

- Para los alumnos que tienen más de 5 tanto en la nota teórica (notat) como en la práctica (notap) escribirá en el campo **observaciones**:
  - “Aprobado”
- Para los alumnos a los que les falta alguna nota (de las dos notas) escribirá en **observaciones**:
  - “Faltan pruebas por evaluar”
- Para los alumnos con alguna nota suspendida (<5), escribirá en el campo **observaciones**:
  - “Prueba/s suspendidas”

Crea el script en javascript necesario para (1º) actualizar el campo “categoria” en cada uno de los casos indicados y (2º) listar SOLO apellidos, nombre y observaciones de TODOS los documentos ordenados por apellidos y nombre.

### SCRIPT 3: MINERÍA DE DATOS (3 PUNTOS)

a) Se requiere listar los datos de las asignaturas (nombre y apellidos del alumno, asignatura, maestro y coste) con coste superior a 5, ordenados por nombre y apellidos del alumno. Crea el script necesario.

b) Se requiere listar el nombre, apellidos y coste total del alumno que más ha costado en todas las asignaturas. Crea el script necesario.

### SCRIPT 4: CONTADORES (2 PUNTOS)

Se requiere (1º) contar el número de estudiantes que tienen las dos notas y, o bien han aprobado las dos partes teórica y práctica, o bien han suspendido las dos partes. Crea el script necesario.

### SCRIPT 5: BORRADO (1 PUNTO)

Se requiere (1º) listar los nombres y apellidos de los estudiantes a los que, o bien les falta alguna nota, o bien han suspendido la parte práctica (<5) y luego (2º) borrarlos. Crea el script necesario.