
Proyecto No. 2

MongoDB - PAREJAS o TRÍOS

I. Modalidad y fecha de entrega

- a) El proyecto debe realizarse en parejas o grupos de tres personas. Deberán asignarse a los grupos en canvas.
- b) Debe ser enviado antes de la fecha límite de entrega: **Martes 6 de mayo a las 18:59 horas** a más tardar
- c) El orden de los grupos a presentar será sorteado aleatoriamente previo a la semana de entrega. Las presentaciones de proyecto se basarán en la dinámica de *show and tell* al catedrático durante el período de clase, cada grupo deberá presentar las funcionalidades (requeridas y extras) de su proyecto.
- d) Algunos grupos podrán escoger entregar hasta el jueves 8 de mayo antes de las 17:20 horas, pero deberán cambiar de lugar con otro grupo que presentaba ese día (en caso de que originalmente presentaran el martes). Ambos grupos deberán acordar este movimiento. Si decide entregar hasta el jueves, no podrá obtener más del 75% de su nota.
- e) Consultar la rúbrica de evaluación.

II. Objetivo y descripción de la actividad

El objetivo de este proyecto es desarrollar un

El objetivo de este proyecto es desarrollar un **Sistema de Gestión de Pedidos y Reseñas de Restaurantes** a través de una REST API **funcional** que facilite la gestión y almacenamiento de datos relacionados con pedidos y reseñas de restaurantes en MongoDB (se recomienda utilizar Atlas).

Esta aplicación permitirá a los usuarios interactuar con el sistema para realizar y gestionar pedidos, dejar reseñas sobre los restaurantes y consultar información relevante en tiempo real. La aplicación abordará aspectos clave, como el uso de documentos embebidos y referenciados, operaciones CRUD, ordenamientos, proyecciones, manejo eficiente de archivos grandes, agregaciones y manipulación de arrays.

Los usuarios podrán registrar restaurantes, realizar pedidos y gestionar reseñas, utilizando estructuras de documentos que combinen información embebida y referenciada. Por ejemplo, un pedido contendrá detalles de los platillos seleccionados, el estado del pedido y el usuario que lo realizó, mientras que las reseñas podrán estar asociadas tanto a restaurantes como a pedidos específicos. La aplicación permitirá realizar operaciones CRUD, ofreciendo la capacidad para administrar restaurantes, pedidos y/o comentarios.

Además, se implementarán capacidades de ordenamiento para clasificar pedidos por fecha, estado o monto total, y las reseñas por calificación o relevancia. Se emplearán proyecciones para optimizar la carga de datos, mostrando únicamente la información necesaria en cada contexto. Asimismo, se utilizarán agregaciones para generar reportes, como los restaurantes mejor calificados o los platillos más vendidos.

Este sistema garantizará una gestión eficiente de los pedidos y reseñas, proporcionando una experiencia fluida para los usuarios y optimizando la administración de datos mediante MongoDB.

Instrucciones generales y observaciones

Se debe desarrollar una REST API funcional que permita gestionar data para almacenarla/consumirla en MongoDB. Su aplicación, deberá incluir y manejar los siguientes aspectos:

- Documentos Embedded y Referenced
- Operaciones CRUD
- Sorts

- Projects
- Manejo y almacenamiento de archivos
- Agregaciones
- Manejo de arrays
- Creación y uso de índices

Se recomienda consultar la rúbrica de evaluación para mayor detalle. Lo más importante es la interacción que hay con los datos que maneja MongoDB y cómo desde el REST API se pueden manipular estos datos. Recuerden que no basta con generar los métodos del API como tal, sino que los implementen en una aplicación funcional que haga sentido. Es importante que como grupo puedan manejar la lógica funcional como lo más importante de su proyecto, para que dicha aplicación tenga sentido más que la implementación de código que permita interactuar con el API sin un propósito.

ETAPA 01 – Modelado de datos

Una vez definidos los grupos y asignados en canvas, deberán pensar en el caso de uso a modelar e implementar sobre MongoDB. Este caso de uso deberá de utilizar por lo menos las siguientes colecciones y deberá argumentar cómo piensa implementar los aspectos a evaluar en el proyecto.

1. Restaurantes
2. Usuarios
3. Artículos del menú
4. Órdenes (pedidos)
5. Reseñas

Adicionalmente, su base de datos debe estar configurada para rechazar todas aquellas consultas que no utilicen índices, por lo que se deberá diseñar y crear los índices necesarios para poder ejecutar todas las consultas de su servicio REST. Algunos tipos de índices que deben implementar son:

- Simples
- Compuestos
- Multikey
- Geoespaciales
- Texto

ETAPA 02 - Elaboración del servicio REST

Con su modelo de datos implementado, se encargará de implementar la solución en el lenguaje de programación a su preferencia. Se recomienda que utilice una instancia en Atlas para fomentar y facilitar el trabajo colaborativo. Lo más importante será cómo implementen la solución con los aspectos a evaluar de la rúbrica sobre una aplicación funcional que haga sentido con el caso de uso solicitado. Aspectos como el frontend serán extras.

ETAPA 03 - Presentación

Presentará los resultados de su proyecto y el funcionamiento del mismo durante los periodos de clase. La forma de evaluación será empezar explicando su proyecto y caso de uso, e ir mostrando cómo su solución cumple con cada uno de los aspectos a evaluar.

III. Temas a reforzar

- Funcionamiento general de MongoDB y sus componentes
- Modelado de datos
- Operaciones CRUD
- Manejo eficiente de archivos

- Agregaciones y proyecciones
- Ordenamiento de datos y uso de arrays
- Usabilidad de una base de datos orientada a documentos

IV. Entregables

1. Código fuente desarrollado. (Repositorio con historial de cambios)
2. Video general del funcionamiento de su aplicación, que no dure más de **10 minutos**.
3. Documentación de todas las configuraciones realizadas en Tableau, Power BI, Atlas, MongoDB Charts, etc., en caso haya realizado alguno de estos puntos extra. Puede apoyarse de capturas de pantalla y demás recursos para evidenciar cada paso de las configuraciones realizadas y del entregable (gráficas) como tal.

V. Rúbrica de Evaluación

Criterio		Puntaje
Avances	Etapas 1	5
	Etapas 2	5
	Precarga de data con mongoimport	5
Índices	Simple	1
	Compuestos	1
	Multikey	1
	Geoespaciales	1
	Texto	1
	Mínimo 2 por colección	-
Documentos	Embedded	5
	Referenciados	5
CRUD	Creación de Documentos - Creación de 1 documento - Creación de varios documentos	10
	Consulta de Documentos - Filtros - Proyecciones - Ordenamiento - Skip - Limite	10
	Actualización de Documentos - Actualizar 1 documento - Actualizar varios documentos	10
	Eliminación de Documentos - Eliminar 1 documento - Eliminar varios documentos	10

GridFS	Manejo de Archivos: poder interactuar con la aplicación y ver los cambios reflejados en los archivos fuente que soportan la data de su proyecto en MongoDB. Deberá de existir al menos una colección con al menos 50,000 documentos creados inicialmente.	5
Agregaciones y otros	Simples (count, distinct, etc.)	5
	Complejas (pipelines de agregación complejas)	10
	Manejo de Arrays (\$push, \$pull, \$addToSet, etc)	5
	Manejo de documentos embebidos.	5
Extras	Operaciones BULK (bulkwrite) Cualquier tipo, hasta un máximo de 5 puntos extra.	5
	Mongo Charts Cualquier chart que tenga un sentido de negocio, 2 puntos por gráfica embebida, hasta un máximo de 6 puntos.	6
	BI Connectors Integración con herramientas de BI como Power BI, Tableau, etc. Máximo 4 pts.	4
	Frontend/HCI, interfaz amigable. Máximo 5 puntos.	5

Nota: Lo marcado en **amarillo** hace referencia a los puntos extra. El proyecto tiene un valor total de 15 puntos netos pero será calificado sobre 100. Podrá llegar a obtener hasta 120 puntos en total.