DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN - DCCO-SS

CARRERA EN TECNOLOGÍAS DE LA INFORMACIÓN

Nombre: Guerra Jennifer
Materia: Modelado Avanzado de Base de Datos
Proyecto: Tarea 2 Individual - El Cronista de Datos NoSQL

Descripción del Proyecto

Este repositorio implementa un **Bestiario Digital de Criaturas Fantásticas** utilizando MongoDB como base de datos NoSQL. El proyecto demuestra las ventajas de las bases de datos orientadas a documentos frente a los modelos relacionales tradicionales, aprovechando la flexibilidad de esquema de MongoDB para representar la heterogeneidad natural de criaturas fantásticas de diferentes animes.

Estructura del Repositorio

Figura 1. *Estructura del repositorio*

```
mision-mongodb-jenn/
│
├── README.md                    # Este archivo
├── misiones_mongodb.js          # Script con operaciones CRUD
├── ANALISIS_NOSQL.md           # Análisis y reflexión teórica
└── README.pdf                   # Versión PDF de este documento
```

Configuración del Entorno

Creación del Cluster en MongoDB Atlas

El proyecto fue desarrollado utilizando MongoDB Atlas, la plataforma en la nube de MongoDB. El proceso de configuración incluyó:

1. Creación de cuenta en MongoDB Atlas

- Acceso a https://www.mongodb.com/cloud/atlas
- Registro con cuenta institucional

2. Configuración del Cluster

- Creación de un cluster gratuito (M0 Sandbox)
- Región seleccionada: AWS / N. Virginia (us-east-1)

- Nombre del cluster: Cluster0

3. Configuración de Seguridad

- Creación de usuario de base de datos con credenciales
- Configuración de IP Access List (permitir acceso desde cualquier IP: 0.0.0.0/0)
- Habilitación de autenticación mediante usuario y contraseña

4. Obtención de la cadena de conexión

Figura 2. *Conexión en MongoDB Compass*

```
mongodb+srv://<username>:<password>@cluster0.xxxxx.mongodb.net/
```

Conexión mediante MongoDB Compass

MongoDB Compass fue utilizado como interfaz gráfica principal para interactuar con la base de datos:

1. **Instalación de MongoDB Compass**
   - Descarga desde https://www.mongodb.com/products/compass
   - Instalación en sistema local
2. **Conexión al Cluster**
   - Apertura de MongoDB Compass
   - Ingreso de la cadena de conexión proporcionada por Atlas
   - Formato: *mongodb+srv://<username>:<password>@cluster0.xxxxx.mongodb.net/*
   - Verificación de conexión exitosa
3. **Interfaz de Trabajo**
   - Visualización de bases de datos existentes
   - Acceso a MongoShell integrado en Compass para ejecución de comandos

Ejecución del Proyecto

Método de Ejecución Utilizado

El script *misiones_mongodb.js* fue ejecutado utilizando **MongoShell dentro de MongoDB Compass**:

1. Apertura de MongoDB Compass
2. Conexión al cluster de Atlas
3. Acceso a la pestaña **“_MongoSH”** (MongoShell integrado)
4. Ejecución de comandos línea por línea del archivo *misiones_mongodb.js*

Proceso de Implementación

Figura 3. *Creación de Base de Datos y Colección*

```
// Conectarse y crear la base de datos
use bestiario

// Crear la colección de criaturas
db.createCollection("criaturas")
```

2. Operaciones de Inserción (CREATE)

Se utilizó *insertOne()* para insertar criaturas individuales con estructuras variables:

Figura 4. *Inserción uno a uno*

```
db.criaturas.insertOne({
    nombre: "Rey de las Bestias Marinas",
    anime: "One Piece",
    tipo: "Rey Marino",
    clasificacion: "depredador apex",
    tamaño_aproximado_km: 5,
    habitat_exclusivo: "Calm Belt",
    numero_de_escamas: "incontables",
    tipo_escamas: "casi impenetrables",
    dieta: "carnívoro - devora barcos enteros",
    debilidad_fatal: "usuarios de Akuma no Mi en el agua",
    inmune_a: ["cañones convencionales", "espadas normales"],
    velocidad_nado: "extremadamente rápida",
    territorios: ["East Blue Calm Belt", "Grand Line Calm Belt"],
    avistamientos_registrados: 47,
    barcos_destruidos: 200,
    fecha_registro: new Date()
})
```

Se utilizó *insertMany()* para insertar múltiples documentos en una sola operación:

Figura 5. *Inserción de varios documentos*

```
db.criaturas.insertMany([
    {
        nombre: "Titán Colosal",
        anime: "Attack on Titan",
        tipo: "Titán Cambiante",
        habitat: "Murallas de Paradis",
        nivel_peligro: 10,
        altura_metros: 60,
        habilidades: [
            "Explosión de vapor",
            "Regeneración",
            "Transformación"
        ],
        estadisticas: {
            fuerza: 9800,
            velocidad: 2000,
            resistencia: 9500
        },
        debilidad: "nuca",
        usuario: "Bertholdt Hoover",
        puede_hablar: false,
        fecha_registro: new Date()
    },
    {
        nombre: "Hollow Vasto Lorde",
        anime: "Bleach",
        tipo: "Hollow Evolucionado",
        habitat: "Hueco Mundo",
        nivel_peligro: 9,
        mascara: true,
        agujero_hollow: "pecho",
        habilidades: [
            "Cero",
            "Sonido",
            "Regeneración instantánea",
            "Hierro"
        ],
        poder_espiritual: 9500,
        rareza: "extremadamente raro",
        inteligencia: "humana completa",
        fecha_registro: new Date()
    }
])
```

3. Operaciones de Lectura (READ)

Se implementaron diferentes tipos de consultas para recuperar información

Figura 6. *Recuperación de información*

```
// Mostrar todas las criaturas
db.criaturas.find().pretty()

// Filtrar criaturas por hábitat específico
db.criaturas.find({ habitat: "Profundidades del Abismo" })

// Buscar criaturas con nivel de peligro mayor a 8
db.criaturas.find({ nivel_peligro: { $gt: 8 } })
```

4. Operaciones de Actualización (UPDATE)

Se realizaron actualizaciones tanto individuales como masivas:

Figura 1. *Actualización de documentos*

```
// Actualizar un documento específico (agregar habilidad)
db.criaturas.updateOne(
    { nombre: "Kaiju No. 8" },
    { $push: { habilidades: "Modo Berserk" } }
)

// Actualizar múltiples documentos (incrementar nivel de peligro)
db.criaturas.updateMany(
    { habitat: "Profundidades del Abismo" },
    { $inc: { nivel_peligro: 1 } }
)
```

Características Técnicas Implementadas

Flexibilidad de Esquema

El proyecto demuestra cómo MongoDB permite que cada documento tenga su propia estructura única sin necesidad de un esquema rígido predefinido. Cada criatura posee atributos específicos:

- **Kaiju No. 8**: *numero_nucleo*, *forma_humana*, *puede_transformarse*
- **Seiryu**: *fruta_del_diablo*, *forma_dragon*, *haki_dominado*
- **Guardián del Abismo**: *nucleos_vitales*, *jinkis_bajo_control*, *territorio*

Documentos Auto-Contenidos

Cada criatura se almacena como un documento JSON completo que incluye:

- Objetos anidados: *forma_humana*, *estadisticas_combate*, *territorio*
- Arrays: *habilidades*, *ataques_especiales*, *peleas_legendarias*
- Datos complejos sin necesidad de realizar operaciones JOIN

Operadores MongoDB Utilizados

Operador	Propósito	Ejemplo de Uso
$gt	Comparación "mayor que"	*{ nivel_peligro: { $gt: 8 } }*
$push	Agregar elemento a un array	*{ $push: { habilidades: "nueva" } }*
$inc	Incrementar valor numérico	*{ $inc: { nivel_peligro: 1 } }*

Análisis y Reflexión

Para un análisis detallado sobre las diferencias entre SQL y NoSQL, tipos de bases de datos NoSQL, y casos de estudio reales, consultar el archivo: ***ANALISIS_NOSQL.md***

Tecnologías Utilizadas

Tecnología	Versión	Propósito
MongoDB Atlas	Cloud	Plataforma de base de datos en la nube
MongoDB Compass	Latest	Interfaz gráfica y herramienta de administración
MongoShell	Integrado	Ejecución de comandos y scripts
JavaScript	ES6+	Lenguaje para operaciones CRUD

Referencias

1. Integrate.io. (2025). "SQL vs NoSQL: 5 Critical Differences". Recuperado de https://www.integrate.io/blog/the-sql-vs-nosql-difference/
2. Integrate.io. (2025). "Understanding NoSQL Databases". Recuperado de https://www.integrate.io/blog/understanding-nosql-databases/
3. MongoDB Official Documentation. https://docs.mongodb.com
4. MongoDB, Inc. (2025). "Toyota Connected targets at least 99.99% availability with MongoDB assistance". https://www.mongodb.com/solutions/customer-case-studies/toyota-connected
5. Neo4j Graph Database Documentation. https://neo4j.com/docs/
6. Fowler, M. "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence"