

Actividad 1 - Diseño y operaciones CRUD en Bases de datos NoSQL

Presentado por:

Cristian Leandro Pérez Peláez

Natalia Sierra Salamando



IBERO

Corporación Universitaria Iberoamericana

Facultad de Ingeniería

Bases De Datos Avanzadas

Profesor: Jorge Castañeda

Domingo, 17 de noviembre de 2024

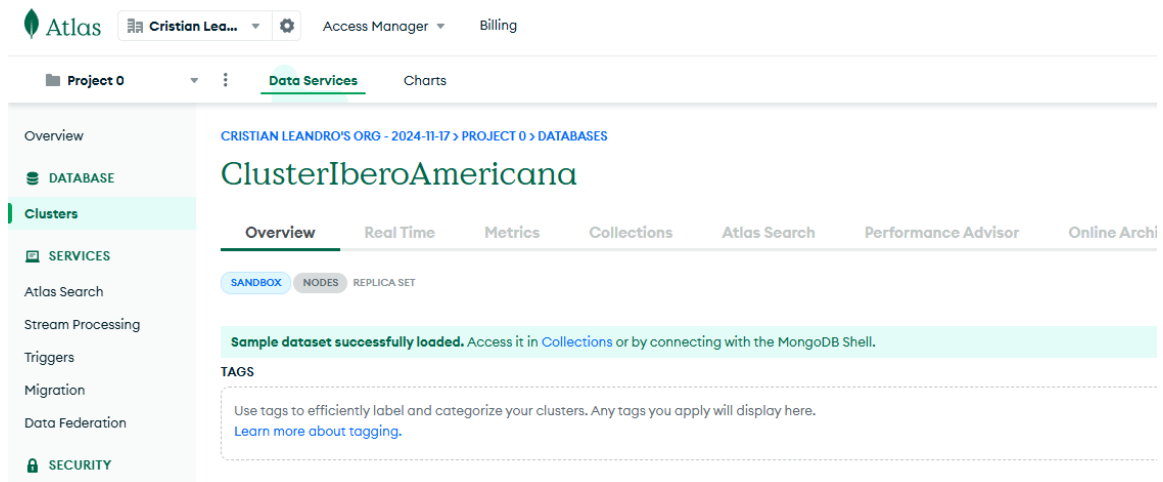
Actividad 1 - Diseño y operaciones CRUD en Bases de datos NoSQL

1. Requisitos para la creación de la base de datos:

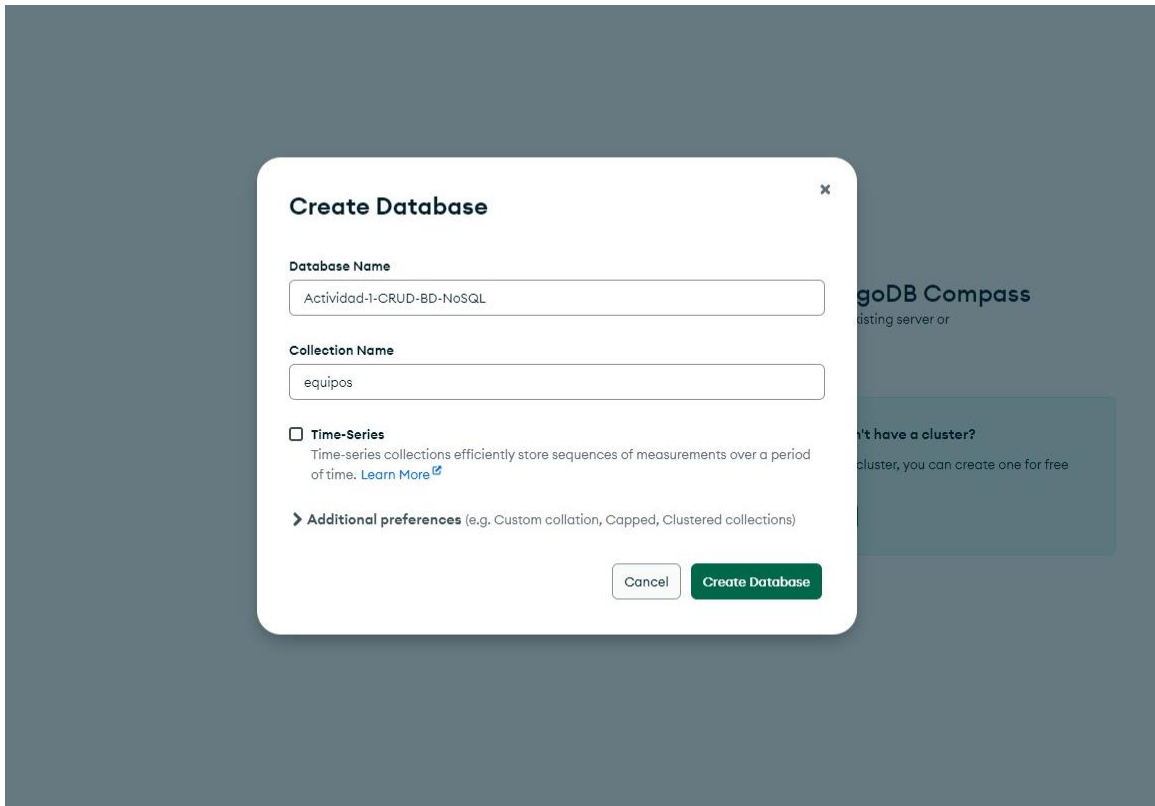
- **Formato del Torneo de futbol y sus reglas:**
 - 20 equipos participantes.
 - Cada equipo debe tener entre 15 y 25 jugadores.
 - Los equipos se enfrentan una vez y compiten entre todos.
 - Una victoria otorga 3 puntos, un empate 1 punto, y una derrota 0 puntos.
 - Los equipos se deben ordenar en la tabla de posiciones según sus puntos.
- **Datos importantes del torneo:**
 - **Jugadores:** Incluyen nombre, edad, posición, goles, asistencias, tarjetas amarillas, tarjetas rojas y el equipo al que pertenecen.
 - **Entrenadores:** Cada equipo tiene un entrenador con nombre, edad, experiencia, estrategia y equipo asignado.
 - **Árbitros:** Hay un grupo de árbitros asignados aleatoriamente a los encuentros, los cuales deben tener nombre, edad, experiencia y partidos asignados.
 - **Equipos:** Los equipos deben contener nombre, ciudad, entrenador, jugadores, puntos, victorias, derrotas, empates, goles a favor, goles en contra, diferencia de goles.
 - **Partidos:** Los datos relevantes son importante para la organización por eso se debe tener en cuenta que cada partido contenga fecha, lugar, equipo local, equipo visitante, arbitro, resultado, goles del equipo local, goles del equipo visitante, ganador, estadísticas, posesión local, posesión visitante, tarjetas amarillas, tarjetas rojas.
- **Informes Requeridos:**
 - Tabla de posiciones actualizada.
 - Detalles de los encuentros, incluyendo resultados.
 - Estadísticas individuales por jugador (goles, asistencias, etc.).
 - Histórico de partidos por equipo (victorias, derrotas, empates).

2. Diseño de las bases de datos:

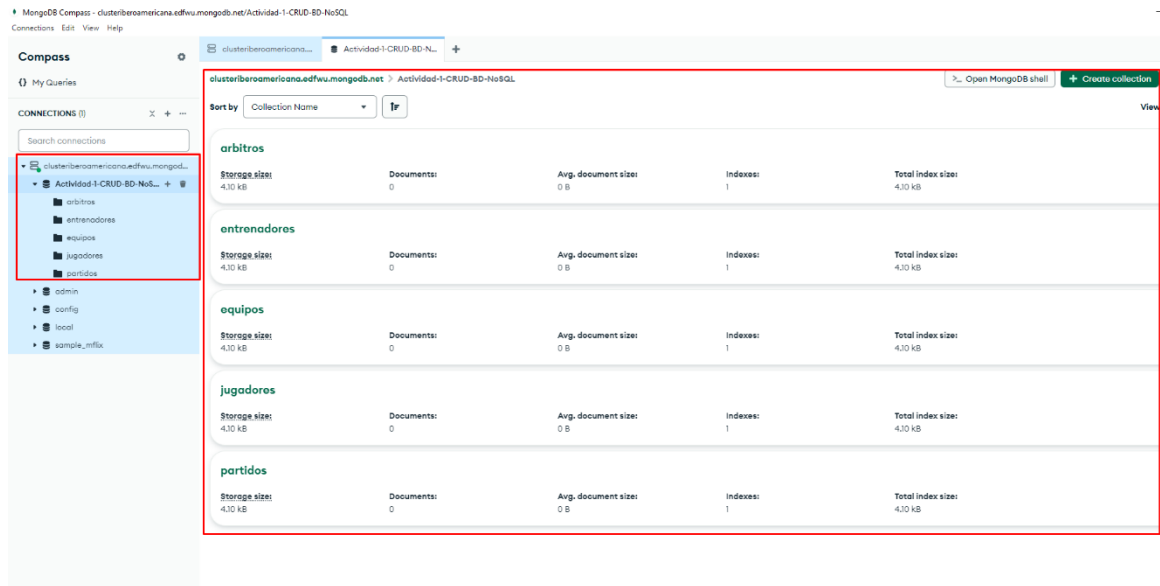
Para el diseño de nuestra base de datos con Mongo DB, hemos usado Mongo DB Atlas para la creación del cluster llamado **clusterIberoAmericana** y generación de conexión URI:



Para la creación de este cluster se realizó la creación de la base de datos **Actividad-1-CRUD-BD-NoSQL** y la primera colección **equipos**:



Conectamos nuestro cluster de mongo DB Atlas con Mongo DB Compass:



Estructura/modelo de cada colección:

- **Colección: equipos:**

```
{
  "_id": "equipo_1",
  "nombre": "Tigres FC",
  "ciudad": "Bogotá",
  "entrenadorId": "entrenador_1",
  "jugadores": ["jugador_1", "jugador_2", "jugador_3"],
  "puntos": 10,
  "victorias": 3,
  "derrotas": 2,
  "empates": 1,
  "golesAFavor": 12,
  "golesEnContra": 8,
  "diferenciaGoles": 4
}
```

- **Colección jugadores:**

```
{  
  "_id": "jugador_1",  
  "nombre": "Carlos Pérez",  
  "edad": 24,  
  "posicion": "Delantero",  
  "equipold": "equipo_1",  
  "goles": 5,  
  "asistencias": 2,  
  "tarjetasAmarillas": 1,  
  "tarjetasRojas": 0  
}
```

- **Colección entrenadores:**

```
{  
  "_id": "entrenador_1",  
  "nombre": "Juan López",  
  "edad": 45,  
  "experiencia": "10 años",  
  "estrategia": "Ofensiva",  
  "equipold": "equipo_1"  
}
```

- **Colección arbitros:**

```
{  
  "_id": "arbitro_1",  
  "nombre": "Ana García",  
  "edad": 35,  
  "experiencia": "8 años",  
}
```

```
"partidosAsignados": ["partido_1", "partido_2"],  
"tarjetasSacadas": 3  
}
```

- **Colección partidos:**

```
{  
  "_id": "partido_1",  
  "fecha": "2024-11-20",  
  "lugar": "Estadio Metropolitano",  
  "equipoLocalId": "equipo_1",  
  "equipoVisitanteId": "equipo_2",  
  "arbitroId": "arbitro_1",  
  "resultado": {  
    "golesLocal": 2,  
    "golesVisitante": 1,  
    "ganador": "equipo_1"  
  },  
  "estadisticas": {  
    "posesionLocal": 55,  
    "posesionVisitante": 45,  
    "tarjetasAmarillas": ["jugador_3", "jugador_5"],  
    "tarjetasRojas": []  
  }  
}
```

3. Operación o query para generar cada informe requerido

- Tabla de posiciones actualizada:
 - `db.equipos.find({}, { nombre: 1, puntos: 1, victorias: 1, empates: 1, derrotas: 1 }).sort({ puntos: -1 });`
- Detalles de los encuentros, incluyendo resultados.
 - `db.partidos.find({ _id: "partido_1" });`
- Estadísticas individuales por jugador (goles, asistencias, etc.).
 - `db.jugadores.find({ _id: "jugador_1" }, { nombre: 1, goles: 1, asistencias: 1, tarjetasAmarillas: 1, tarjetasRojas: 1 });`
- Histórico de partidos por equipo (victorias, derrotas, empates).
 - `db.partidos.find({ $or: [{ equipoLocalId: "equipo_1" }, { equipoVisitanteId: "equipo_1" }] });`

4. Operaciones CRUD para la colección equipo:

- Crear un equipo:

```
db.equipos.insertOne({
  _id: "equipo_3",
  nombre: "Leones FC",
  ciudad: "Cali",
  entrenadorId: "entrenador_3",
  jugadores: ["jugador_5", "jugador_6"],
  puntos: 0,
  victorias: 0,
  derrotas: 0,
  empates: 0,
  golesAFavor: 0,
  golesEnContra: 0,
  diferenciaGoles: 0
});
```

- Leer todos los equipos:
`db.equipos.find();`
- Leer/buscar un equipo en específico:
`db.equipos.find({ _id: "equipo_3" });`
- Actualizar un equipo:
`db.equipos.updateOne(
 { _id: "equipo_3" },
 { $set: { puntos: 3, victorias: 1 } }
);`
- Eliminar un equipo:
`db.equipos.deleteOne({ _id: "equipo_3" });`

5. Operaciones CRUD para la colección Jugadores:

- Crear un jugador:
`db.jugadores.insertOne({
 _id: "jugador_5",
 nombre: "Luis Martínez",
 edad: 23,
 posicion: "Portero",
 equipoid: "equipo_3",
 goles: 0,
 asistencias: 0,
 tarjetasAmarillas: 0,
 tarjetasRojas: 0
});`
- Leer todos los jugadores:
`db.jugadores.find();`

- Leer/buscar un jugador:

```
db.jugadores.find({ equipold: "equipo_1" });
```

- Actualizar en incrementar la cantidad de un valor, por ejemplo con el parámetro “\$inc” incrementamos la cantidad de goles y asistencias en este caso:

```
db.jugadores.updateOne(  
  { _id: "jugador_5" },  
  { $inc: { goles: 1, asistencias: 1 } }  
);
```

- Eliminar un jugador:

```
db.jugadores.deleteOne({ _id: "jugador_5" });
```

6. Operaciones CRUD para la colección entrenadores:

- Crear un entrenador:

```
db.entrenadores.insertOne({  
  _id: "entrenador_3",  
  nombre: "Carlos Gómez",  
  edad: 50,  
  experiencia: "15 años",  
  estrategia: "Defensiva",  
  equipold: "equipo_3"  
});
```

- Leer todos los entrenadores:

```
db.entrenadores.find();
```

- Leer un entrenador en específico:

```
db.entrenadores.find({ _id: "entrenador_3" });
```

- Actualziar un entrenador:

```
db.entrenadores.updateOne(
  { _id: "entrenador_3" },
  { $set: { estrategia: "Ofensiva" } }
);
```

- Eliminar un entrenador:

```
db.entrenadores.deleteOne({ _id: "entrenador_3" });
```

7. Operaciones CRUD para la colección arbitros:

- Crear un árbitro:

```
db.arbitros.insertOne({
  _id: "arbitro_2",
  nombre: "Pedro Sánchez",
  edad: 40,
  experiencia: "12 años",
  partidosAsignados: []
});
```

- Leer todos los árbitros:

```
db.arbitros.find();
```

- Leer/buscar un árbitro en específico, en este caso usamos una expresión regular para separar los números de la palabra años:

```
db.arbitros.find({ experiencia: { $regex: "^[1-9][0-9]* años" } });
```

- Actualizar un árbitro, en este caso asignarle un partido:

```
db.arbitros.updateOne(
  { _id: "arbitro_2" },
```

```
{ $push: { partidosAsignados: "partido_2" } }  
);
```

- Eliminar un árbitro:

```
db.arbitros.deleteOne({ _id: "arbitro_2" });
```

8. Operaciones CRUD para la colección partidos:

- Crear un partido:

```
db.partidos.insertOne({  
  _id: "partido_2",  
  fecha: "2024-11-21",  
  lugar: "Estadio Olímpico",  
  equipoLocalId: "equipo_2",  
  equipoVisitanteId: "equipo_3",  
  arbitroId: "arbitro_2",  
  resultado: {  
    golesLocal: 0,  
    golesVisitante: 2,  
    ganador: "equipo_3"  
  },  
  estadisticas: {  
    posesionLocal: 40,  
    posesionVisitante: 60,  
    tarjetasAmarillas: ["jugador_5"],  
    tarjetasRojas: []  
  }  
});
```

- Leer todos los partidos:

```
db.partidos.find();
```

- Leer partidos de un equipo:

```
db.partidos.find({ $or: [{ equipoLocalId: "equipo_3" }, { equipoVisitanteId: "equipo_3" } ] });
```

- Actualizar un partido:

```
db.partidos.updateOne(  
  { _id: "partido_2" },  
  { $set: { "resultado.golesLocal": 1, "resultado.ganador": null } }  
);
```

- Eliminar un partido:

```
db.partidos.deleteOne({ _id: "partido_2" });
```