

ACTIVIDAD 3

Pregunta 1 — Flexibilidad de Esquema

MongoDB permite modificar la estructura de los documentos sin necesidad de cambiar toda la base de datos.

Esto es una gran ventaja, porque en un juego roguelike las mecánicas pueden cambiar constantemente (nuevos objetos, enemigos, o estadísticas).

Sin embargo, la desventaja es que puede volverse desorganizado si no se controla bien, ya que cada documento podría tener campos diferentes.

Usar Mongoose ayuda a mantener orden mientras seguimos teniendo flexibilidad.

Pregunta 2: Relaciones (Embedding vs Referencing)

- **Ejemplo de Referencia:**
Usamos referencias en el campo "userId" dentro de "characters" y "runs", porque un jugador puede tener varios personajes y runs, y no queremos duplicar toda su información cada vez.
Además, si el usuario cambia su nombre, se actualiza automáticamente en todos los lugares donde se usa la referencia.
- **Ejemplo de Incrustación:**
Usamos incrustación en el campo "stats" dentro de "characters" y "enemies".
Esto es útil porque esas estadísticas siempre se leen junto con el personaje o enemigo, así que no es necesario buscarlas en otra colección.

Pregunta 3: Denormalización (Duplicación de Datos)

En nuestro juego, podríamos duplicar algunos datos para mejorar el rendimiento: Por ejemplo, al guardar el "username" del jugador dentro del documento "runs". Así, cuando mostramos una tabla de puntuaciones (leaderboard), no necesitamos hacer muchas consultas entre colecciones.

Aunque significa más trabajo si el usuario cambia su nombre, hace que el juego cargue más rápido en tiempo real.

Colecciones del Proyecto (Juego Multijugador Roguelike en Godot)

Aquí están únicamente las colecciones que necesitará tu base de datos en MongoDB.

Colección: users

```
{
  "username": { "type": "String", "required": true, "unique": true },
  "email": { "type": "String", "required": true, "unique": true },
  "password": { "type": "String", "required": true },
  "createdAt": { "type": "Date", "default": "Date.now" }
}
```

Colección: characters

```
{
  "owner": { "type": "ObjectId", "ref": "users", "required": true },
  "class": { "type": "String", "required": true },
  "health": { "type": "Number", "required": true },
  "damage": { "type": "Number", "required": true },
  "speed": { "type": "Number", "required": true },
  "abilities": [ "String" ],
  "inventory": [
    {
      "itemId": { "type": "ObjectId", "ref": "items" },
      "quantity": { "type": "Number", "default": 1 }
    }
  ]
}
```

Colección: items

```
{
  "name": { "type": "String", "required": true },
  "type": { "type": "String", "required": true },
  "rarity": { "type": "String", "required": true },
  "effects": [ "String" ],
  "value": { "type": "Number", "default": 0 }
}
```

Colección: enemies

```
{
  "name": { "type": "String", "required": true },
  "health": { "type": "Number", "required": true },
  "damage": { "type": "Number", "required": true },
  "speed": { "type": "Number", "required": true },
  "tier": { "type": "String", "required": true }
}
```

Colección: waves

```
{
  "waveNumber": { "type": "Number", "required": true },
  "enemies": [
    {
      "enemyId": { "type": "ObjectId", "ref": "enemies" },
      "quantity": { "type": "Number", "required": true }
    }
  ]
}
```

```
]
}
```

Colección: matches

```
{
  "players": [ { "type": "ObjectId", "ref": "users" } ],
  "status": { "type": "String", "default": "waiting" },
  "currentWave": { "type": "Number", "default": 1 },
  "maxWaves": { "type": "Number", "default": 10 },
  "createdAt": { "type": "Date", "default": "Date.now" }
}
```

Colección: sessions (Sesiones activas de usuarios)

```
{
  "userId": { "type": "ObjectId", "ref": "users", "required": true },
  "token": { "type": "String", "required": true },
  "createdAt": { "type": "Date", "default": "Date.now" },
  "expiresAt": { "type": "Date", "required": true }
}
```

Colección: friends (Lista de amigos entre jugadores)

```
{
  "userId": { "type": "ObjectId", "ref": "users", "required": true },
  "friends": [
    {
      "friendId": { "type": "ObjectId", "ref": "users" },
      "addedAt": { "type": "Date", "default": "Date.now" }
    }
  ]
}
```

Colección: leaderboard (Tabla de clasificación global)

```
{
  "userId": { "type": "ObjectId", "ref": "users", "required": true },
  "score": { "type": "Number", "required": true },
  "position": { "type": "Number", "required": true },
  "updatedAt": { "type": "Date", "default": "Date.now" }
}
```

Colección: profiles (Perfiles públicos de los usuarios)

```
{
  "userId": { "type": "ObjectId", "ref": "users", "required": true },
  "bio": { "type": "String", "default": "" },
  "avatarUrl": { "type": "String", "default": "" },
  "joinedAt": { "type": "Date", "default": "Date.now" }
}
```

```
}
```

<https://github.com/david56w/Bunny-inferno>