



PRÁCTICA III

Grupo 08

Gloria Blanca Llorente Pardo
Ángel Luis Rodríguez Otero
Raúl Somavilla González
Elena Milara Mencía
Javier Castro Magro

ÍNDICE

1. Identificación de las consultas más frecuentes	2
2. Colecciones propuestas y descripción de los documentos que las componen	3
2.1 Json Usuarios	3
2.2 Json Partida	3
2.3 Json Mapa.....	5
2.4 Json Objetos	7
2.5 Json Habilidades	9
3. Creación de documentos.....	10
4. Diseño de operaciones CRUD	11
4.1 Consultas más frecuentes	11
4.2 Consultas de filtrado	14
4.3 Consultas de ordenación	17
4.4 Consultas de agrupación	23
5. Creación de Índices y estudio de rendimiento	26
5.1 Colección usuarios:.....	26
5.2 Colección Partidas:	27
5.3 Colección Personajes:.....	28
5.4 Colección Boses y enemigos:	29
5.5 Colección sitios:.....	31
5.6 Colección NPCs:	32
5.7 Colección Armas y Consumibles.....	32
5.8 Colección Armaduras:	34
5.9 Colección Hechizos:.....	35
5.10 Colección Formas:	36

1. IDENTIFICACIÓN DE LAS CONSULTAS MÁS FRECUENTES

Nombre y contraseña: Cada vez que un jugador inicie el juego, se deberá de acceder a la base de datos para cargar su nombre y contraseña en caso de que inicie la sesión de forma automática o verificar los datos que introduzca el jugador con los que proporcionó en su registro.

Estadísticas de personajes: Los jugadores podrían querer consultar las estadísticas de sus personajes, como fuerza, destreza, constitución, inteligencia, sabiduría y carisma. También podrían querer ver sus habilidades específicas o hechizos disponibles...

Inventario y objetos: Los jugadores pueden querer consultar su inventario para ver qué objetos tienen actualmente y qué objetos pueden necesitar buscar o adquirir. Esto incluiría armas, armaduras y consumibles como pociones, pergaminos y otros objetos útiles.

Mapa y ubicaciones: Los jugadores podrían querer consultar el mapa del juego para ver su ubicación actual, así como las ubicaciones de interés cercanas, como ciudades, mazmorras o puntos de referencia importantes.

Estadísticas de juego: Los jugadores podrían estar interesados en consultar estadísticas sobre su progreso en el juego, como el tiempo total de juego.

Información sobre enemigos y NPCs: Consultar información sobre enemigos y criaturas específicas podría ser útil para los jugadores al planificar estrategias de combate o para aprender más sobre el mundo del juego (dónde encontrarlos, acto en el que aparecen, características, habilidades...).

Habilidades y mejoras: Los jugadores pueden querer consultar información sobre cómo mejorar las habilidades de sus personajes o a la hora de aprender nuevas habilidades, poder conocer su daño, qué hechizos curan y cuáles no...

2. COLECCIONES PROPUESTAS Y DESCRIPCIÓN DE LOS DOCUMENTOS QUE LAS COMPONEN

Para la práctica se han creado cinco documentos JSON para realizar las consultas de manera más sencilla. Se han elegido los cinco bloques más importantes del juego: Usuarios, Partida, Mapa, Objetos y Habilidades.

Cada documento JSON está dividido a su vez en colecciones.

LEYENDA

array object string number boolean null

2.1 JSON USUARIOS

COLECCIÓN USUARIOS

Sus documentos están formados por: un id "ID_Usuario" el nombre del usuario y su contraseña.

```
{"ID_Usuario": 1, "Nick": "Zephyr", "Password": "GwYhEILZgM"}
```

Usuarios	[{"ID_Usuario":1,"Nick":"Zephyr","Password":"GwYhEILZgM"}, {"ID_Usuario":2,"Nick":"L...
0	{"ID_Usuario":1,"Nick":"Zephyr","Password":"GwYhEILZgM"}
ID_Usuario	1
Nick	"Zephyr"
Password	"GwYhEILZgM"

2.2 JSON PARTIDA

En Partida se han creado dos colecciones, la información de la partida que el jugador está jugando en el momento, y los personajes que están jugando en esa partida.

COLECCIÓN PARTIDAS

Los documentos están formados por: "Escenario" que es el lugar en el que se encuentra el personaje, "Duración" tiempo que lleva jugando esa partida, "Progreso" porcentaje de juego que ha completado, toma valores entre 0-100 y "Usuario_Nick" que indica el nombre del usuario.

```
{"Escenario":1,"Duración":5,"Progreso":4,"Usuario_Nick":"Zephyr"}
```

Partidas	[{"Escenario":1,"Duración":5,"Progreso":4,"Usuario_Nick":"Zephyr"}, {"Escenario":2,"Duración":5,"Progreso":4,"Usuario_Nick":"Zephyr"}]
0	{"Escenario":1,"Duración":5,"Progreso":4,"Usuario_Nick":"Zephyr"}
Escenario	1
Duración	5
Progreso	4
Usuario_Nick	"Zephyr"

COLECCIÓN PERSONAJES

Documentos formados por: "ID_Personaje" de tipo numérico, "Nombre" nombre del personaje, "Raza" raza del personaje, "Clase" clase del personaje, "Avatar" booleano que indica si el personaje es creado por el jugador o por defecto del juego, por último sus características ("Fuerza", "Destreza", "Constitución", "Inteligencia", "Sabiduría" y "Carisma"), que son de tipo numérico y "HechizosDisponibles" que indica la cantidad de hechizos que puede realizar ese personaje.

```
{"ID_Personaje":1, "Nombre":"Laezel", "Raza":"Githyanki", "Clase":"Fighter", "Avatar":false, "Nivel":7, "Fuerza":17, "Destreza":13, "Constitucion":15, "Inteligencia":10, "Sabiduria":12, "Carisma":8, "HechizosDisponibles":13}
```

Personajes	[{"ID_Personaje":1, "Nombre":"Laezel", "Raza":"Githyanki", "Clase":"Fighter", "Avatar":false}, {"ID_Personaje":2, "Nombre":"Laezel", "Raza":"Githyanki", "Clase":"Fighter", "Avatar":false}]
0	{"ID_Personaje":1, "Nombre":"Laezel", "Raza":"Githyanki", "Clase":"Fighter", "Avatar":false}
ID_Personaje	1
Nombre	"Laezel"
Raza	"Githyanki"
Clase	"Fighter"
Avatar	false
Nivel	7
Fuerza	17
Destreza	13
Constitucion	15
Inteligencia	10
Sabiduría	12
Carisma	8
HechizosDisponibles	13

2.3 JSON MAPA

En este documento se han añadido las colecciones de la información que puede necesitar el jugador cuando abre el mapa. La colección con información de los sitios que hay, la colección con información de los enemigos y boses, además de la colección npcs con la información de cada uno.

COLECCIÓN SITIOS

Los documentos están formados por: "ID_Sitio" de tipo numérico, "Sitio" que indica el nombre del sitio, "Subterráneo" booleano que indica si esa zona del mapa tiene zonas subterráneas o no.

```
{"ID_Sitio":1,"Sitio":"Nautiloid","Subterráneo": false}
```

Sitios	[{"ID_Sitio":1,"Sitio":"Nautiloid","Subterráneo":false}, {"ID_Sitio":2,"Sitio":"Ravaged Beac"}]
0	{"ID_Sitio":1,"Sitio":"Nautiloid","Subterráneo":false}
ID_Sitio	1
Sitio	"Nautiloid"
Subterráneo	false

COLECCIÓN ENEMIGOS

Los documentos están formados por: "ID_Enemigo" de tipo numérico, "Nombre" que indica el nombre del enemigo, "Tipo" que indica qué tipo de enemigo es, características ("Fuerza", "Destreza", "Constitución", "Inteligencia", "Sabiduría" y "Carisma"), que son de tipo numérico y "Mapa_ID_Sitio" que indica la zona del mapa en la que aparece.

```
{"Id_Enemigo": 1,"Nombre": "Intellect Devourer","Tipo": "Aberrations","Fuerza": 6,"Destreza": 14,"Constitucion": 13,"Inteligencia": 12,"Sabiduria": 11,"Carisma": 10,"Mapa_ID_Sitio": 1}
```

Enemigos	[{"Id_Enemigo":1,"Nombre":"Intellect Devourer","Tipo":"Aberrations","Fuerza":6,"Destreza":14,"Constitucion":13,"Inteligencia":12,"Sabiduria":11,"Carisma":10,"Mapa_ID_Sitio":1,"NFases":1,"Acto":1,"Habilidades":"Attack of Opportunity"}]
0	[{"Id_Enemigo":1,"Nombre":"Intellect Devourer","Tipo":"Aberrations","Fuerza":6,"Destreza":14,"Constitucion":13,"Inteligencia":12,"Sabiduria":11,"Carisma":10,"Mapa_ID_Sitio":1,"NFases":1,"Acto":1,"Habilidades":"Attack of Opportunity"}]
Id_Enemigo	1
Nombre	"Intellect Devourer"
Tipo	"Aberrations"
Fuerza	6
Destreza	14
Constitucion	13
Inteligencia	12
Sabiduria	11
Carisma	10
Mapa_ID_Sitio	1

COLECCIÓN BOSES

Los documentos de boses, tienen la misma estructura que la de enemigos, pero además añade: "NFases" que indica el número de fases que tiene el boss (puede ir de 1 a 3 fases), "Acto" acto de la historia en la que te puedes enfrentar a él (los actos van de 1 a 3) y por último "Habilidades" que te indica una de las habilidades que tiene el boss.

```
{
  "Id_Enemigo": 51,
  "Nombre": "True Soul Gut",
  "Tipo": "Goblins",
  "Fuerza": 14,
  "Destreza": 12,
  "Constitucion": 14,
  "Inteligencia": 16,
  "Sabiduria": 18,
  "Carisma": 16,
  "Mapa_ID_Sitio": 13,
  "NFases": 1,
  "Acto": 1,
  "Habilidades": "Attack of Opportunity"
}
```

Boses	[{"Id_Enemigo":51,"Nombre":"True Soul Gut","Tipo":"Goblins","Fuerza":14,"Destreza":12,"Constitucion":14,"Inteligencia":16,"Sabiduria":18,"Carisma":16,"Mapa_ID_Sitio":13,"NFases":1,"Acto":1,"Habilidades":"Attack of Opportunity"}]
0	[{"Id_Enemigo":51,"Nombre":"True Soul Gut","Tipo":"Goblins","Fuerza":14,"Destreza":12,"Constitucion":14,"Inteligencia":16,"Sabiduria":18,"Carisma":16,"Mapa_ID_Sitio":13,"NFases":1,"Acto":1,"Habilidades":"Attack of Opportunity"}]
Id_Enemigo	51
Nombre	"True Soul Gut"
Tipo	"Goblins"
Fuerza	14
Destreza	12
Constitucion	14
Inteligencia	16
Sabiduria	18
Carisma	16
Mapa_ID_Sitio	13
NFases	1
Acto	1
Habilidades	"Attack of Opportunity"

COLECCIÓN NPCS

Los documentos están formados por: “Nombre_NPC” que indica el nombre del NPC, “Tipo” que indica el tipo de NPC, las características (“Fuerza”, “Destreza”, “Constitución”, “Inteligencia”, “Sabiduría” y “Carisma”), que son de tipo numérico y “Mapa_ID_Sitio” que indica la zona del mapa en la que aparece.

```
{"Nombre_NPC":"Myrnath","Tipo":"Elves","Fuerza":10,"Destreza":10,"Constitucion":10,"Inteligencia":10,"Sabiduria":10,"Carisma":10,"Mapa_ID_Sitio":1}
```

Npcs	[{"Nombre_NPC":"Myrnath","Tipo":"Elves","Fuerza":10,"Destreza":10,"Constitucion":10,"
0	{"Nombre_NPC":"Myrnath","Tipo":"Elves","Fuerza":10,"Destreza":10,"Constitucion":10,"
Nombre_NPC	"Myrnath"
Tipo	"Elves"
Fuerza	10
Destreza	10
Constitucion	10
Inteligencia	10
Sabiduria	10
Carisma	10
Mapa_ID_Sitio	1

2.4 JSON OBJETOS

En este documento se han metido los elementos que teníamos heredando al elemento Objeto en la primera práctica. Las colecciones creadas son Armas, Armaduras y Consumibles. En cada colección se ha metido información de los objetos que corresponden a ese tipo de objeto.

COLECCIÓN ARMAS

Los documentos están formados por: “Id_Objeto” de tipo numérico, “Nombre” que indica el nombre del arma, “Damage” que indica el daño que hace el arma, “Tipo” indica el tipo de arma (Staff, Pike, Axe, Sword), “Habilidad” indica la habilidad del arma, “Rareza” rareza del arma (Common, Uncommon, Rare, Very rare, Legendary) y “Valor” que indica el precio por el que se puede comerciar el arma.


```
{ "Id_Objeto":1,"Nombre":"Staff of the
Emperor","Damage":10,"Tipo":"Staff","Habilidad":"Piercing
Strike","Rareza":"Rare","Valor":190}
```

Armas	[{"Id_Objeto":1,"Nombre":"Staff of the Emperor","Damage":10,"Tipo":"Staff","Habilidad":
0	{"Id_Objeto":1,"Nombre":"Staff of the Emperor","Damage":10,"Tipo":"Staff","Habilidad":
Id_Objeto	1
Nombre	"Staff of the Emperor"
Damage	10
Tipo	"Staff"
Habilidad	"Piercing Strike"
Rareza	"Rare"
Valor	190

COLECCIÓN ARMADURAS

Los documentos están formados por: "Id_Objeto" de tipo numérico, "Nombre" que indica el nombre de la armadura, "Defensa" que indica cuanto te protege la armadura, "Pasiva" que indica la pasiva de la armadura, "Tipo" que indica que parte de la armadura es (Chestplate, Helmet, Gloves), "Rareza" rareza de la armadura (Common, Uncommon, Rare, Very rare, Legendary) y "Valor" que indica el precio por el que se puede comerciar esa armadura.

```
{ "Id_Objeto":21,"Nombre":"Simple Robe","Defensa":10,"Pasiva":
null,"Tipo":"Chestplate","Rareza":"Common","Valor":20}
```

Armaduras	[{"Id_Objeto":21,"Nombre":"Simple Robe","Defensa":10,"Pasiva":null,"Tipo":"Chestplate
0	{"Id_Objeto":21,"Nombre":"Simple Robe","Defensa":10,"Pasiva":null,"Tipo":"Chestplate"
Id_Objeto	21
Nombre	"Simple Robe"
Defensa	10
Pasiva	null
Tipo	"Chestplate"
Rareza	"Common"
Valor	20

COLECCIÓN CONSUMIBLES

Los documentos están formados por: “Id_Objeto” de tipo numérico, “Nombre” que indica el nombre del consumible, “Tipo” que indica qué tipo de consumible es (Potions, Elixirs, Food), “Efecto” que indica el tipo de efecto que ejerce sobre el personaje (Healing, Boost, Neutralize), “Rareza” rareza del consumible (Common, Uncommon, Rare, Very rare, Legendary) y “Valor” que indica el precio por el que se puede comerciar ese consumible.

```
{"Id_Objeto":41,"Nombre":"Potion of Healing",
"Tipo":"Potions","Efecto":"Healing",
"Rareza":"Common","Valor":60}
```

Consumibles	[{"Id_Objeto":41,"Nombre":"Potion of Healing","Tipo":"Potions","Efecto":"Healing","Rareza":"Common","Valor":60}]
0	{"Id_Objeto":41,"Nombre":"Potion of Healing","Tipo":"Potions","Efecto":"Healing","Rareza":"Common","Valor":60}
Id_Objeto	41
Nombre	"Potion of Healing"
Tipo	"Potions"
Efecto	"Healing"
Rareza	"Common"
Valor	60

2.5 JSON HABILIDADES

En este documento se han creado las colecciones de las habilidades que van a usar los personajes de nuestro juego, estas dos colecciones en anteriores prácticas eran elementos que se relacionaban con Personaje.

COLECCIÓN HECHIZOS

Los documentos están formados por: “Nombre_Hechizo” que indica el nombre del hechizo, “Damage” el daño que hace el hechizo y “Curación” que indica la curación que realiza dicho hechizo.

```
{"Nombre_Hechizo":"Maleficio","Damage":6,"Curacion":0}
```

Hechizos	[{"Nombre_Hechizo":"Maleficio","Damage":6,"Curacion":0}, {"Nombre_Hechizo":"Ola atr
0	{"Nombre_Hechizo":"Maleficio","Damage":6,"Curacion":0}
Nombre_Hechizo	"Maleficio"
Damage	6
Curacion	0

COLECCIÓN FORMAS

Los documentos están formados por: "Animal" que indica el animal en el que se transforma y "Habilidad" que indica la habilidad que ofrece esa forma.

```
{"Animal":"Cat","Habilidad":"Speed"}
```

Formas	[{"Animal":"Cat","Habilidad":"Speed"}, {"Animal":"Badger","Habilidad":"Poison"}, {"Animal
0	{"Animal":"Cat","Habilidad":"Speed"}
Animal	"Cat"
Habilidad	"Speed"

3. CREACIÓN DE DOCUMENTOS

Para desarrollar los documentos JSON, se han seguido dos procesos. En primer lugar, se utilizó la página <https://www.freeformatter.com/xml-to-json-converter.html> para transformar los documentos que teníamos en XML a JSON. Posteriormente, se comparó el resultado con la estructura deseada, apoyándonos en las consultas más frecuentes. Una vez definida la estructura, se crearon los JSON vacíos y se completaron con los datos almacenados en CSV. En caso de necesitar más documentos en alguna colección específica, se solicitó a una IA que generará más datos siguiendo la estructura establecida.

4. DISEÑO DE OPERACIONES CRUD

Cabe destacar que las salidas están al completo debido a la cantidad de documentos utilizados.

4.1 CONSULTAS MÁS FRECUENTES

1. Nombre y contraseña

Hemos hecho la operación para cuando ya te has metido en el juego y le has dado a “recuérdame”, por lo que el juego automáticamente recibe tu ID (Cuenta de Steam o algo parecido) y te devuelve el Nick y Contraseña para meterte en tu cuenta del juego.

Consulta:

```
db.Usuarios.find({ ID_Usuario: 1 }, { Nick: 1, Password: 1, _id: 0 })
```

Salida:

```
> db.Usuarios.find({ ID_Usuario: 1 }, { Nick: 1, Password: 1, _id: 0 })
< {
  Nick: 'Zephyr',
  Password: 'GwYhELLZgM'
}
```

2. Poder ver las características de tu personaje

Hemos creado esta consulta para cuando un jugador necesite saber si puede realizar una acción dependiendo de una característica, ya sea una cualidad (Fuerza, Inteligencia) o simplemente el nivel.

Consulta Introduciendo el ID del personaje:

```
db.Personajes.find({ID_Personaje:"12"})
```

Salida:

```
> db.Personajes.find({ID_Personaje:12})
< {
  _id: ObjectId('66461814868ddc6aaf927686'),
  ID_Personaje: 12,
  Nombre: 'Javier',
  Raza: 'Drow',
  Clase: 'Warlock',
  Avatar: true,
  Nivel: 2,
  Fuerza: 8,
  Destreza: 17,
  Constitucion: 10,
  Inteligencia: 15,
  Sabiduria: 12,
  Carisma: 8,
  HechizosDisponibles: 3
}
```

Misma consulta, pero introduciendo el nombre del personaje:

```
db.Personajes.find({Nombre:"Shadowheart"})
```

Salida:

```
> db.Personajes.find({Nombre:"Shadowheart"})
< {
  _id: ObjectId('66461814868ddc6aaf92767e'),
  ID_Personaje: 4,
  Nombre: 'Shadowheart',
  Raza: 'Half-Elf',
  Clase: 'Cleric',
  Avatar: false,
  Nivel: 12,
  Fuerza: 13,
  Destreza: 13,
  Constitucion: 14,
  Inteligencia: 10,
  Sabiduria: 17,
  Carisma: 8,
  HechizosDisponibles: 23
}
```

3. Estadísticas de juego pasando el nombre de tu personaje

Esta consulta sirve para cuando un jugador quiera consultar las estadísticas de la partida que se esté jugando con un usuario concreto.

Consulta:

```
db.Partidas.find({Usuario_Nick:"Cyrus"})
```

Salida:

```
> db.Partidas.find({Usuario_Nick:"Cyrus"})
< {
  _id: ObjectId('66461715868ddc6aaf927663'),
  Escenario: 17,
  'Duración': 127,
  Progreso: 100,
  Usuario_Nick: 'Cyrus'
}
```

4. Obtener información sobre enemigos y bosses buscándolo por su nombre.

Hemos creado esta consulta para cuando el jugador quiera conocer con mayor detalle las características y otra información relevante de los enemigos (entre ellos, los bosses).

Los enemigos se buscan dentro de la colección de enemigos. **Consulta:**

```
db.Enemigos.find({Nombre:"Andorn"})
```

Salida:

```
> db.Enemigos.find({Nombre:"Andorn"})
< {
  _id: ObjectId('6646195b868ddc6aaf9276be'),
  Id_Enemigo: 35,
  Nombre: 'Andorn',
  Tipo: 'Humans',
  Fuerza: 12,
  Destreza: 14,
  Constitucion: 10,
  Inteligencia: 11,
  Sabiduria: 10,
  Carisma: 14,
  Mapa_ID_Sitio: 16
}
```

Para bosses, se busca en la colección de bosses. **Consulta:**

```
db.Boses.find({Nombre:"True Soul Nere"})
```

Salida:

```
> db.Boses.find({Nombre:"True Soul Nere"})
< {
  _id: ObjectId('664619f5868ddc6aaf9276ea'),
  Id_Enemigo: 59,
  Nombre: 'True Soul Nere',
  Tipo: 'Lolth-Sworn Drow',
  Fuerza: 17,
  Destreza: 15,
  Constitucion: 15,
  Inteligencia: 18,
  Sabiduria: 13,
  Carisma: 18,
  NFases: 2,
  Acto: 1,
  Habilidades: 'Attack of Opportunity'
}
```

4.2 CONSULTAS DE FILTRADO

1. Consulta de Enemigos en un Mapa determinado y que te devuelva su información:

Esta consulta facilita al jugador la información de los enemigos en un lugar concreto.

Consulta:

```
db.Enemigos.find({ Mapa_ID_Sitio: 1 }, { Nombre: 1, Tipo: 1, Fuerza: 1, Destreza: 1,
Constitucion: 1, Inteligencia: 1, Sabiduria: 1, Carisma: 1, _id: 0 })
```

Salida:

```
< {
  Nombre: 'Intellect Devourer',
  Tipo: 'Aberrations',
  Fuerza: 6,
  Destreza: 14,
  Constitucion: 13,
  Inteligencia: 12,
  Sabiduria: 11,
  Carisma: 10
}
{
  Nombre: 'Quelenna',
  Tipo: 'Elves',
  Fuerza: 9,
  Destreza: 11,
  Constitucion: 10,
  Inteligencia: 14,
  Sabiduria: 12,
  Carisma: 10
}
{
  Nombre: 'Lesser Hellsboar',
  Tipo: 'Fiend',
  Fuerza: 12,
  Destreza: 11,
  Constitucion: 12,
  Inteligencia: 6,
  Sabiduria: 8,
  Carisma: 5
}
```

2. Partidas con una duración superior a 60 horas:

Una buena forma de conocer cuántos jugadores le han dedicado bastante tiempo al juego (este dato le puede interesar a los desarrolladores del juego).

Consulta:

```
db.Partidas.find({Duración:{$gt: 60}},{Duración: 1, Progreso: 1, Usuario_Nick: 1, _id:0})
```

Salida:

```
> db.Partidas.find({Duración:{$gt: 60}},{Duración: 1, Progreso: 1, Usuario_Nick: 1, _id:0})
< {
  'Duración': 70,
  Progreso: 79,
  Usuario_Nick: 'Thorne'
}
{
  'Duración': 95,
  Progreso: 97,
  Usuario_Nick: 'Ember'
}
{
  'Duración': 75,
  Progreso: 73,
  Usuario_Nick: 'Sylas'
}
{
  'Duración': 90,
  Progreso: 87,
  Usuario_Nick: 'Elara'
}
{
  'Duración': 100,
  Progreso: 99,
  Usuario_Nick: 'Thorne'
}
{
  'Duración': 65,
  Progreso: 62,
  Usuario_Nick: 'Darian'
}
```

3. NPCs con Inteligencia superior a 12

Se ha creado esta consulta como ejemplo, ya que puede haber ocasiones en las que ningún personaje de tu grupo tenga una cualidad (en este caso Inteligencia) suficientemente alta para hacer una acción, como poder leer unas runas, mover una roca pesada (si fuera Fuerza la cualidad que buscamos). De esta manera sabemos quién y donde se encuentra para poder hablar con el npc y que nos ayude en la aventura.

Consulta:

```
db.NPCs.find({Inteligencia:{$gt:12}},{Nombre_NPC: 1, Mapa_ID_Sitio: 1, _id: 0})
```


Salida:

```
> db.NPCs.find({Inteligencia:{$gt:12}},{Nombre_NPC: 1, Mapa_ID_Sitio: 1, _id: 0})
< {
  Nombre_NPC: 'Injured Mind Flayer',
  Mapa_ID_Sitio: 2
}
{
  Nombre_NPC: 'Kagha',
  Mapa_ID_Sitio: 6
}
{
  Nombre_NPC: 'Zevlor',
  Mapa_ID_Sitio: 7
}
{
  Nombre_NPC: 'Lump the Enlightened',
  Mapa_ID_Sitio: 8
}
{
  Nombre_NPC: 'Auntie Ethel',
  Mapa_ID_Sitio: 10
}
```

4. Consulta que devuelva los hechizos que realizan curación

Esta consulta sirve al jugador para cuando necesita un hechizo de un tipo en concreto en combate y no los tiene equipados.

Consulta:

```
db.Hechizos.find({Curacion:{$gt:0}},{Nombre_Hechizo: 1, _id: 0})
```

Salida:

```
> db.Hechizos.find({Curacion:{$gt:0}},{Nombre_Hechizo: 1, _id: 0})
< {
  Nombre_Hechizo: 'Palabra de curación'
}
{
  Nombre_Hechizo: 'Curar Heridas'
}
{
  Nombre_Hechizo: 'Luz Sanadora'
}
{
  Nombre_Hechizo: 'Bálsamo de Paz'
}
{
  Nombre_Hechizo: 'Curación empoderada'
}
{
  Nombre_Hechizo: 'Curación rápida'
}
{
  Nombre_Hechizo: 'Mano Sanadora'
}
{
  Nombre_Hechizo: 'Toque Médico'
}
```

5. Consulta que te devuelve las formas bestia cuya habilidad es Poison

Esta consulta nos sirve para cuando queremos una forma bestia con una habilidad en concreto, en este caso que envenene.

Consulta:

```
db.Formas.find({ Habilidad: "Poison" },{_id: 0})
```

Salida:

```
> db.Formas.find({ Habilidad: "Poison" },{_id: 0})
< {
  Animal: 'Badger',
  Habilidad: 'Poison'
}
{
  Animal: 'Spider',
  Habilidad: 'Poison'
}
{
  Animal: 'Dilophosaurus',
  Habilidad: 'Poison'
}
{
  Animal: 'Sabre-Toothed Tiger',
  Habilidad: 'Poison'
}
```

4.3 CONSULTAS DE ORDENACIÓN

En ese caso no se tenían consultas de ordenación por lo que se han realizado consultas desde cero.

1. Ordenar las armas de tipo espada alfabéticamente por nombre:

Esta consulta es útil para cuando el jugador necesite buscar rápidamente un arma en concreto dentro de su inventario.

Consulta:

```
db.Armas.find({ Tipo: "Sword" }).sort({ Nombre: 1 })
```

Salida:

```
> db.Armas.find({ Tipo: "Sword" }).sort({ Nombre: 1})
< {
  _id: ObjectId('66461ab3868ddc6aaf927735'),
  Id_Objeto: 16,
  Nombre: 'Greatsword',
  Damage: 12,
  Tipo: 'Sword',
  Habilidad: 'Tenacity',
  Rareza: 'Common',
  Valor: 110
}
{
  _id: ObjectId('66461ab3868ddc6aaf927739'),
  Id_Objeto: 20,
  Nombre: 'Silver Sword of the Astral Plane',
  Damage: 21,
  Tipo: 'Sword',
  Habilidad: 'Tenacity',
  Rareza: 'Legendary',
  Valor: 1300
}
{
  _id: ObjectId('66461ab3868ddc6aaf927737'),
  Id_Objeto: 18,
  Nombre: 'Soulbreaker Greatsword',
  Damage: 17,
  Tipo: 'Sword',
  Habilidad: null,
  Rareza: 'Rare',
  Valor: 380
}
}
```

Si queremos que solo se muestren los nombres para que se vea más claro:

```
db.Armas.find({ Tipo: "Sword" }, { Nombre: 1, _id: 0 }).sort({ Nombre: 1 })
```

Salida:

```
> db.Armas.find(
  { Tipo: "Sword" },
  { Nombre: 1, _id: 0 }
).sort({ Nombre: 1 })
< {
  Nombre: 'Greatsword'
}
{
  Nombre: 'Silver Sword of the Astral Plane'
}
{
  Nombre: 'Soulbreaker Greatsword'
}
{
  Nombre: 'Sword of Chaos'
}
{
  Nombre: 'Sword of Justice'
}
}
```

2. Ordenar las partidas con duración superior a 60 horas de forma descendente por duración:

Hemos creado esta consulta para cuando un jugador quiera seleccionar una partida u otra según lo avanzada que esté, y así pueda verlas ordenadas y le sea más fácil elegir la que prefiera.

Consulta:

```
db.Partidas.find({ Duración: { $gt: 60 } }).sort({ Duración: -1 })
```

Salida:

```
> db.Partidas.find({ Duración: { $gt: 60 } }).sort({ Duración: -1 })
< {
  _id: ObjectId('66461715868ddc6aaf927651'),
  Escenario: 17,
  'Duración': 145,
  Progreso: 100,
  Usuario_Nick: 'Kira'
}
{
  _id: ObjectId('66461715868ddc6aaf927650'),
  Escenario: 15,
  'Duración': 130,
  Progreso: 100,
  Usuario_Nick: 'Jack'
}
{
  _id: ObjectId('66461715868ddc6aaf92765a'),
  Escenario: 18,
  'Duración': 128,
  Progreso: 100,
  Usuario_Nick: 'Tristan'
}
```

Si queremos que se muestren solo las duraciones de las partidas para ver la ordenación más clara:

```
db.Partidas.find({ Duración: { $gt: 60 } }, { Duración: 1, _id: 0 }).sort({ Duración: -1 })
```

Salida:

```
> db.Partidas.find({ Duración: { $gt: 60 } },{Duración: 1, _id: 0}).sort({ Duración: -1 })
< {
  'Duración': 145
}
{
  'Duración': 130
}
{
  'Duración': 128
}
{
  'Duración': 128
}
{
  'Duración': 127
}
{
  'Duración': 127
}
{
  'Duración': 115
}
{
  'Duración': 113
}
{
  'Duración': 113
}
{
  'Duración': 112
}
{
  'Duración': 112
}
```

3. Enemigos cuyo nombre empiece por la letra “A”, ordenados por fuerza de menor a mayor:

Se ha creado esta consulta para los casos que el jugador no se acuerde del todo como se llamaba un enemigo, pero se acuerda de como empezaba el nombre, y se acuerda que era bastante fuerte, por lo que será de los primeros de la lista que crea la consulta.

Consulta:

```
db.Enemigos.find({Nombre: { $regex: /^A/ }}).sort({Fuerza: 1})
```

Salida:

```
> db.Enemigos.find({Nombre: { $regex: /^A/ }}).sort({Fuerza: 1})
< {
  _id: ObjectId('6646195b868ddc6aaf9276a2'),
  Id_Enemigo: 6,
  Nombre: 'Addled Frog',
  Tipo: 'Beasts',
  Fuerza: 3,
  Destreza: 16,
  Constitucion: 10,
  Inteligencia: 3,
  Sabiduria: 8,
  Carisma: 8,
  Mapa_ID_Sitio: 10
}
{
  _id: ObjectId('6646195b868ddc6aaf9276be'),
  Id_Enemigo: 35,
  Nombre: 'Andorn',
  Tipo: 'Humans',
  Fuerza: 12,
  Destreza: 14,
  Constitucion: 10,
  Inteligencia: 11,
  Sabiduria: 10,
  Carisma: 14,
  Mapa_ID_Sitio: 16
}
```

Misma consulta, pero mostrando solo la fuerza:

```
db.Enemigos.find({Nombre: { $regex: /^A/ }},{Fuerza: 1, _id: 0}).sort({Fuerza: 1})
```

Salida:

```
> db.Enemigos.find({Nombre: { $regex: /^A/ }},{Fuerza: 1, _id: 0}).sort({Fuerza: 1})
< {
  Fuerza: 3
}
{
  Fuerza: 12
}
{
  Fuerza: 18
}
{
  Fuerza: 22
}
{
  Fuerza: 22
}
{
  Fuerza: 25
}
```

4. Orden de armaduras de tipo chestplate según su defensa de mayor a menor:

Esta consulta es útil para cuando el jugador necesite buscar rápidamente una armadura en concreto dentro de su inventario. Se ordenan de mayor a menor para que pueda elegir antes la armadura que más protege.

Consulta:

```
db.Armaduras.find({Tipo: "Chestplate"}).sort({Defensa: -1})
```

Salida:

```
> db.Armaduras.find({Tipo: "Chestplate"}).sort({Defensa: -1})
< {
  _id: ObjectId('66461ad4868ddc6aaf92773e'),
  Id_Objeto: 25,
  Nombre: 'Helldusk Armour',
  Defensa: 21,
  Pasiva: 'Defiers Rejuvenation',
  Tipo: 'Chestplate',
  Rareza: 'Legendary',
  Valor: 8000
}
{
  _id: ObjectId('66461ad4868ddc6aaf92773a'),
  Id_Objeto: 21,
  Nombre: 'Simple Robe',
  Defensa: 10,
  Pasiva: null,
  Tipo: 'Chestplate',
  Rareza: 'Common',
  Valor: 20
}
```

Misma consulta, pero mostrando solo la defensa:

```
db.Armaduras.find({Tipo: "Chestplate"},{Defensa: 1, _id: 0}).sort({Defensa: -1})
```

Salida:

```
> db.Armaduras.find({Tipo: "Chestplate"},{Defensa: 1, _id: 0}).sort({Defensa: -1})
< {
  Defensa: 21
}
{
  Defensa: 10
}
{
  Defensa: 10
}
{
  Defensa: 10
}
{
  Defensa: 10
}
{
  Defensa: 10
}
```

4.4 CONSULTAS DE AGRUPACIÓN

1. Información sobre los enemigos y bosses:

Esta consulta la hemos creado ya que el jugador puede necesitar la información de un enemigo en específico para poder saber sus debilidades o fuerzas, lo mismo con los bosses. De esta manera podrá prepararse mejor para las peleas contra estos enemigos.

Consulta:

```
db.Enemigos.aggregate([{$lookup:{from:"Bosses",localField:"Id_Enemigo",
foreignField: "Id_Enemigo",
as:"boss_data"}},{ $unwind:{path:"$boss_data",preserveNullAndEmptyArrays:
true}}]).toArray()
```

Salida:

```
> db.Enemigos.aggregate([{$lookup:{from:"Bosses",localField:"Id_Enemigo",foreignField: "Id_Enemigo",
< [
  {
    _id: ObjectId('6646195b868ddc6aaf92769d'),
    Id_Enemigo: 1,
    Nombre: 'Intellect Devourer',
    Tipo: 'Aberrations',
    Fuerza: 6,
    Destreza: 14,
    Constitucion: 13,
    Inteligencia: 12,
    Sabiduria: 11,
    Carisma: 10,
    Mapa_ID_Sitio: 1
  },
  {
    _id: ObjectId('6646195b868ddc6aaf92769e'),
    Id_Enemigo: 2,
    Nombre: 'Spectator',
    Tipo: 'Aberrations',
    Fuerza: 8,
    Destreza: 14,
    Constitucion: 14,
    Inteligencia: 13,
    Sabiduria: 14,
    Carisma: 11,
    Mapa_ID_Sitio: 11
  },
  {
    _id: ObjectId('6646195b868ddc6aaf92769f'),
    Id_Enemigo: 3,
    Nombre: 'Mind Flayer',
    Tipo: 'Aberrations',
    Fuerza: 11,
    Destreza: 12,
    Constitucion: 12,
    Inteligencia: 19,
    Sabiduria: 17,
    Carisma: 17,
    Mapa_ID_Sitio: 2
  },
]
```


2. Armas y armaduras de tipo legendario, mostrando primero las armaduras y luego las armas:

El jugador puede querer conocer las armas y armaduras más poderosas del juego y cuáles son sus atributos.

Consulta:

```
db.Armaduras.aggregate([{$match:{Rareza:"Legendary"}},{ $lookup:{from:"Armas",localField:"Rareza",foreignField:"Rareza",as:"ArmasLegen"}},{ $unwind:"$ArmasLegen"},{$project:{_id:0,Nombre:"$Nombre",Defensa:"$Defensa",Pasiva:"$Pasiva",Tipo:"$Tipo",Rareza:"$Rareza",Valor:"$Valor",TipoObjeto:{$literal:"Armadura"}}},{ $sort:{TipoObjeto:1}},{ $unionWith:{coll:"Armas",pipeline:[{$match:{Rareza:"Legendary"}},{ $project:{_id:0,Nombre:"$Nombre",Damage:"$Damage",Tipo:"$Tipo",Rareza:"$Rareza",Valor:"$Valor",TipoObjeto:{$literal:"Arma"}}},{ $sort:{TipoObjeto:1}}]}])
```

Salida:

```
{
  Nombre: 'Helldusk Armour',
  Defensa: 21,
  Pasiva: 'Defiers Rejuvenation',
  Tipo: 'Chestplate',
  Rareza: 'Legendary',
  Valor: 8000,
  TipoObjeto: 'Armadura'
}
{
  Nombre: 'Mask of the Shapeshifter',
  Defensa: 10,
  Pasiva: 'Cats Grace',
  Tipo: 'Helmet',
  Rareza: 'Legendary',
  Valor: 400,
  TipoObjeto: 'Armadura'
}
{
  Nombre: 'Mask of the Shapeshifter',
  Defensa: 10,
  Pasiva: 'Cats Grace',
  Tipo: 'Helmet',
  Rareza: 'Legendary',
  Valor: 400,
  TipoObjeto: 'Armadura'
}
{
  Nombre: 'Mourning Frost',
  Damage: 13,
  Tipo: 'Staff',
  Rareza: 'Legendary',
  Valor: 640,
  TipoObjeto: 'Arma'
}
{
  Nombre: 'Silver Sword of the Astral Plane',
  Damage: 21,
  Tipo: 'Sword',
  Rareza: 'Legendary',
  Valor: 1300,
  TipoObjeto: 'Arma'
}
```

3. Consulta en la que se muestra el nombre y el valor de las armas y consumibles con un valor superior a 100.

Esta consulta la hemos creado para cuando un jugador quiera vender objetos a partir de un valor puesto, en este caso superior a 100. De esta manera se hace más rápido y no se tiene que ir uno a uno mirando el valor.

Consulta:

```
db.Armas.aggregate([{$match:{Valor:{$gt:100}}},{ $project: {_id:0,Nombre:"$Nombre",Valor:"$Valor"}},{ $unionWith:{coll:"Consumibles",pipeline:[{$match:{Valor:{$gt:100}}},{ $project: {_id:0,Nombre:"$Nombre",Valor:"$Valor"}}]}])
```

Salida:

```
{
  Nombre: 'Soulbreaker Greatsword',
  Valor: 380
}
{
  Nombre: 'Sword of Chaos',
  Valor: 960
}
{
  Nombre: 'Silver Sword of the Astral Plane',
  Valor: 1300
}
{
  Nombre: 'Potion of Angelic Slumber',
  Valor: 200
}
{
  Nombre: 'Elixir of the Colossus',
  Valor: 200
}
```

5. CREACIÓN DE ÍNDICES Y ESTUDIO DE RENDIMIENTO

Para el estudio del uso de documentos se ha utilizado al final de la consulta `.explain("executionStats")` pero empleando únicamente esto la información que ofrece es demasiado detallada, por lo que se ha decidido comparar en las consultas los milisegundos que tarda en hacerse y la cantidad de documentos que son examinados.

Para ello se ha añadido lo siguiente a cada consulta:

```
var stats = db.miColeccion.find().explain("executionStats");

print("Tiempo de ejecución (ms): " + stats.executionStats.executionTimeMillis +

      ", Documentos examinados: " + stats.executionStats.totalDocsExamined +

      ", Claves examinadas: " + stats.executionStats.totalKeysExamined);
```

5.1 COLECCIÓN USUARIOS:

Para la consulta:

```
db.Usuarios.find({ ID_Usuario: 1 }, { Nick: 1, Password: 1, _id: 0 })
```

Sería óptimo crear un índice en `ID_Usuario` ya que encontrará de manera más eficiente el usuario con ese ID.

Creación del índice:

```
db.Usuarios.createIndex({ ID_Usuario: 1 })
```

Consulta para estudiar el rendimiento:

```
var stats = db.Usuarios.find({ ID_Usuario: 1 }, { Nick: 1, Password: 1, _id: 0 }).explain("executionStats");

print("Tiempo de ejecución (ms): " + stats.executionStats.executionTimeMillis + ",
Documentos examinados: " + stats.executionStats.totalDocsExamined);
```

RENDIMIENTO:

Sin índice consultó un total de 69 documentos, mientras que **con el índice** creado consultó solo 1.

5.2 COLECCIÓN PARTIDAS:

Para la consulta de ordenar las partidas con duración superior a 60 horas de forma descendente por duración:

```
db.Partidas.find({ Duración: { $gt: 60 } },{Duración: 1, _id: 0}).sort({ Duración: -1 })
```

Es óptimo crear un índice en duración, ya que es el dato por el que estás filtrando.

Creación del índice:

```
db.Partidas.createIndex({ Duración: 1 })
```

Consulta para estudiar el rendimiento:

```
var stats = db.Partidas.find({ Duración: { $gt: 60 } },{Duración: 1, _id: 0}).sort({ Duración: -1 }).explain("executionStats");
```

```
print("Tiempo de ejecución (ms): " + stats.executionStats.executionTimeMillis + ",  
Documentos examinados: " + stats.executionStats.totalDocsExamined);
```

RENDIMIENTO:**Sin índice:**

Tiempo de ejecución (ms): 0, Documentos examinados: 69

Con índice:

En este caso el índice cambia en el acceso a claves que se han realizado. Documentos examinados: 0.

Tiempo de ejecución (ms): 0, TotalKeysExamined: 34

5.3 COLECCIÓN PERSONAJES:

Para la consulta de información del personaje:

```
db.Personajes.find({ID_Personaje:"12"})
```

En esta es óptimo crear un índice en ID_Personaje, ya que es el dato por el que estás filtrando.

Creación del índice:

```
db.Personajes.createIndex({ID_Personaje: 1})
```

Consulta para estudiar el rendimiento:

```
var stats = db.Personajes.find({ID_Personaje: "12"}).explain("executionStats");  
  
print("Tiempo de ejecución (ms): " + stats.executionStats.executionTimeMillis +  
  
      ", Documentos examinados: " + stats.executionStats.totalDocsExamined +  
  
      ", Claves examinadas: " + stats.executionStats.totalKeysExamined);
```

RENDIMIENTO:

Sin índice:

Tiempo de ejecución (ms): 0, Documentos examinados: 16

Con índice:

Tiempo de ejecución (ms): 0, Documentos examinados: 0, Claves examinadas: 0

Si el filtrado se realiza por nombre:

```
db.Personajes.find({Nombre:"Shadowheart"})
```

Es óptimo crear el índice en el nombre.

Creación del índice:

```
db.Personajes.createIndex({Nombre: 1})
```

Consulta para estudiar el rendimiento:

```
var stats = db.Personajes.find({Nombre:"Shadowheart"}).explain("executionStats");  
  
print("Tiempo de ejecución (ms): " + stats.executionStats.executionTimeMillis +  
  
      ", Documentos examinados: " + stats.executionStats.totalDocsExamined +  
  
      ", Claves examinadas: " + stats.executionStats.totalKeysExamined);
```

RENDIMIENTO:**Sin índice:**

Tiempo de ejecución (ms): 0, Documentos examinados: 16, Claves examinadas: 0

Con índice:

Tiempo de ejecución (ms): 0, Documentos examinados: 1, Claves examinadas: 1

5.4 COLECCIÓN BOSES Y ENEMIGOS:**Para la consulta de información sobre los enemigos y bosses:**

```
db.Enemigos.aggregate([{$lookup:{from:"Boses",localField:"Id_Enemigo",foreignField:  
"Id_Enemigo",  
as:"boss_data"}},{ $unwind:{path:"$boss_data",preserveNullAndEmptyArrays:  
true}}]).toArray();
```

Es óptimo crear un índice en cada colección para Id_Enemigo, ya que es el dato por el que se está filtrando.

Creación de los índices:

```
db.Enemigos.createIndex({ Id_Enemigo: 1 })
```

```
db.Boses.createIndex({ Id_Enemigo: 1 })
```

Consulta para estudiar el rendimiento:

```
var stats =  
db.Enemigos.aggregate([{$lookup:{from:"Boses",localField:"Id_Enemigo",foreignField:  
"Id_Enemigo",as:"boss_data"}},{ $unwind:{path:"$boss_data",preserveNullAndEmptyA  
rrays:true}}]).explain("executionStats"); printjson(stats); if (stats.executionStats) {  
print("Tiempo de ejecución (ms): " + stats.executionStats.executionTimeMillis + ",  
Documentos examinados: " + stats.executionStats.totalDocsExamined); } else {  
print("executionStats no está definido."); }
```

RENDIMIENTO:

Escaneo de la Colección Enemigos:

Sin Índices: Se realizó un escaneo completo de la colección Enemigos (COLLSCAN) para buscar coincidencias, lo que resultó en la examinación de todos los 69 documentos de la colección.

Con Índices: También se realizó un escaneo completo de la colección Enemigos (COLLSCAN) para buscar coincidencias, pero esta vez no se examinaron los documentos de la colección (totalDocsExamined es 0). Sin embargo, no se notó un cambio significativo en el número de documentos devueltos o en el tiempo de ejecución.

Operación \$lookup:

Sin Índices: La operación de \$lookup escaneó la colección Boses 69 veces (collectionScans es 69), lo que resultó en la examinación de un total de 2622 documentos de la colección.

Con Índices: La operación de \$lookup sigue escaneando la colección Boses 69 veces, pero esta vez no se examinan los documentos de la colección (totalDocsExamined es 0). Esto sugiere que la operación de \$lookup se está beneficiando del índice en la colección Boses, pero no hay un cambio significativo en el número de documentos examinados o en el tiempo de ejecución.

5.5 COLECCIÓN SITIOS:

Para consulta de Enemigos en un Mapa determinado y que te devuelva su información:

```
db.Enemigos.find({ Mapa_ID_Sitio: 1 }, { Nombre: 1, Tipo: 1, Fuerza: 1, Destreza: 1,
Constitucion: 1, Inteligencia: 1, Sabiduria: 1, Carisma: 1, _id: 0 })
```

Es óptimo crear un índice en Mapa_ID_Sitio ya que es el dato por el que filtras.

Creación del índice:

```
db.Enemigos.createIndex({ Mapa_ID_Sitio: 1 })
```

Consulta para estudiar el rendimiento:

```
var stats = db.Enemigos.find({ Mapa_ID_Sitio: 1 }, { Nombre: 1, Tipo: 1, Fuerza: 1,
Destreza: 1, Constitucion: 1, Inteligencia: 1, Sabiduria: 1, Carisma: 1, _id: 0
}).explain("executionStats");
```

```
print("Tiempo de ejecución (ms): " + stats.executionStats.executionTimeMillis +
```

```
    ", Documentos examinados: " + stats.executionStats.totalDocsExamined +
```

```
    ", Claves examinadas: " + stats.executionStats.totalKeysExamined);
```

RENDIMIENTO:

Sin Índice:

Tiempo de ejecución (ms): 0, Documentos examinados: 69, Claves examinadas: 0

Con Índice:

Tiempo de ejecución (ms): 0, Documentos examinados: 5, Claves examinadas: 5

5.6 COLECCIÓN NPCS:

Para la consulta NPCs con Inteligencia superior a 12:

```
db.NPCs.find({Inteligencia:{$gt:12}},{Nombre_NPC: 1, Mapa_ID_Sitio: 1, _id: 0})
```

Es óptimo crear un índice en Inteligencia ya que es lo que se usa como filtro.

Creación del índice:

```
db.NPCs.createIndex({ Inteligencia: 1 })
```

Consulta para estudiar el rendimiento:

```
var stats = db.NPCs.find({Inteligencia:{$gt:12}},{Nombre_NPC: 1, Mapa_ID_Sitio: 1, _id: 0}).explain("executionStats");
```

```
print("Tiempo de ejecución (ms): " + stats.executionStats.executionTimeMillis +
```

```
    ", Documentos examinados: " + stats.executionStats.totalDocsExamined +
```

```
    ", Claves examinadas: " + stats.executionStats.totalKeysExamined);
```

RENDIMIENTO:

Sin Índice:

Tiempo de ejecución (ms): 0, Documentos examinados: 30, Claves examinadas: 0

Con Índice:

Tiempo de ejecución (ms): 0, Documentos examinados: 5, Claves examinadas: 5

5.7 COLECCIÓN ARMAS Y CONSUMIBLES

Para la consulta en la que se muestra el nombre y el valor de las armas y consumibles con un valor superior a 100.

```
db.Armas.aggregate([{$match:{Valor:{$gt:100}}},{ $project:{_id:0,Nombre:"$Nombre", Valor:"$Valor"}},{ $unionWith:{coll:"Consumibles",pipeline:[{$match:{Valor:{$gt:100}}},{ $project:{_id:0,Nombre:"$Nombre",Valor:"$Valor"}}]}]);
```

Este índice en el campo "Tipo" permitirá a MongoDB encontrar más eficientemente las armaduras del tipo "Chestplate". Además, dado que la consulta también implica una operación de ordenación por el campo "Defensa", MongoDB puede aprovechar este índice para optimizar la etapa de ordenación si el índice cubre tanto el campo "Tipo" como el campo "Defensa".

Creación de índices:

```
db.Armas.createIndex({ Valor: 1 })
```

```
db.Consumibles.createIndex({ Valor: 1 })
```

Consulta para estudiar el rendimiento:

```
db.Armas.aggregate([{$match:{Valor:{$gt:100}}},{ $project:{_id:0,Nombre:"$Nombre",
Valor:"$Valor"}},{ $unionWith:{coll:"Consumibles",pipeline:[{$match:{Valor:{$gt:100}}},
{$project:{_id:0,Nombre:"$Nombre",Valor:"$Valor"}}]}]).explain("executionStats");
```

RENDIMIENTO:**Sin Índice:**

Para la colección "Armas":

La consulta examinó un total de 20 documentos.

El tiempo de ejecución fue de 0 milisegundos.

Para la colección "Consumibles":

La consulta examinó un total de 20 documentos.

El tiempo de ejecución fue de 0 milisegundos.

La consulta en conjunto devolvió un total de 16 documentos.

Con Índice:

Para la colección "Armas":

La consulta examinó un total de 14 claves (índice).

La consulta examinó un total de 14 documentos.

El tiempo de ejecución fue de 0 milisegundos.

Para la colección "Consumibles":

La consulta examinó un total de 2 claves (índice).

La consulta examinó un total de 2 documentos.

El tiempo de ejecución fue de 0 milisegundos.

La consulta en conjunto devolvió un total de 16 documentos.

5.8 COLECCIÓN ARMADURAS:

Para la consulta de orden de armaduras de tipo chestplate según su defensa de mayor a menor:

```
db.Armaduras.find({Tipo: "Chestplate"},{Defensa: 1, _id: 0}).sort({Defensa: -1})
```

Es óptimo crear un índice en tipo, ya que es el dato por el que filtras.

Creación del índice:

```
db.Armaduras.createIndex({ Tipo: 1 })
```

Consulta para estudiar el rendimiento:

```
var stats = db.Armaduras.find({Tipo: "Chestplate"},{Defensa: 1, _id: 0}).sort({Defensa: -1}).explain("executionStats");
```

```
print("Tiempo de ejecución (ms): " + stats.executionStats.executionTimeMillis +
```

```
    ", Documentos examinados: " + stats.executionStats.totalDocsExamined +
```

```
    ", Claves examinadas: " + stats.executionStats.totalKeysExamined);
```

RENDIMIENTO:**Sin Índice:**

Tiempo de ejecución (ms): 0, Documentos examinados: 20, Claves examinadas: 0

Con Índice:

Tiempo de ejecución (ms): 0, Documentos examinados: 5, Claves examinadas: 5

5.9 COLECCIÓN HECHIZOS:**Para la consulta que devuelva los hechizos que realizan curación:**

```
db.Hechizos.find({Curacion:{$gt:0}},{Nombre_Hechizo: 1, _id: 0})
```

Como estás filtrando los documentos basados en el campo "Curación", agregar un índice en este campo puede mejorar significativamente el rendimiento de la consulta al permitir que MongoDB acceda rápidamente a los documentos que cumplen con el criterio de búsqueda.

Creación del índice:

```
db.Hechizos.createIndex({ Curacion: 1 })
```

Consulta para estudiar el rendimiento:

```
var stats = db.Hechizos.find({Curacion:{$gt:0}},{Nombre_Hechizo: 1, _id: 0}).explain("executionStats");
```

```
print("Tiempo de ejecución (ms): " + stats.executionStats.executionTimeMillis +
```

```
    ", Documentos examinados: " + stats.executionStats.totalDocsExamined +
```

```
    ", Claves examinadas: " + stats.executionStats.totalKeysExamined);
```

RENDIMIENTO:**Sin Índice:**

Tiempo de ejecución (ms): 0, Documentos examinados: 17, Claves examinadas: 0

Con Índice:

Tiempo de ejecución (ms): 0, Documentos examinados: 8, Claves examinadas: 8

5.10 COLECCIÓN FORMAS:

Para la consulta que te devuelve las formas bestia cuya habilidad es Poison

```
db.Formas.find({ Habilidad: "Poison" }, {_id: 0})
```

Es óptimo crear un índice en Habilidad. Este índice permitirá a MongoDB encontrar rápidamente los documentos que coincidan con el valor específico de Habilidad cuando realices consultas como la que mencionaste.

Creación del índice:

```
db.Formas.createIndex({ Habilidad: 1 })
```

Consulta para estudiar el rendimiento:

```
var stats = db.Formas.find({ Habilidad: "Poison" }, {_id: 0}).explain("executionStats");
```

```
print("Tiempo de ejecución (ms): " + stats.executionStats.executionTimeMillis +
```

```
    ", Documentos examinados: " + stats.executionStats.totalDocsExamined +
```

```
    ", Claves examinadas: " + stats.executionStats.totalKeysExamined);
```

RENDIMIENTO:

Sin Índice:

Tiempo de ejecución (ms): 0, Documentos examinados: 10, Claves examinadas: 0

Con Índice:

Tiempo de ejecución (ms): 0, Documentos examinados: 4, Claves examinadas: 4