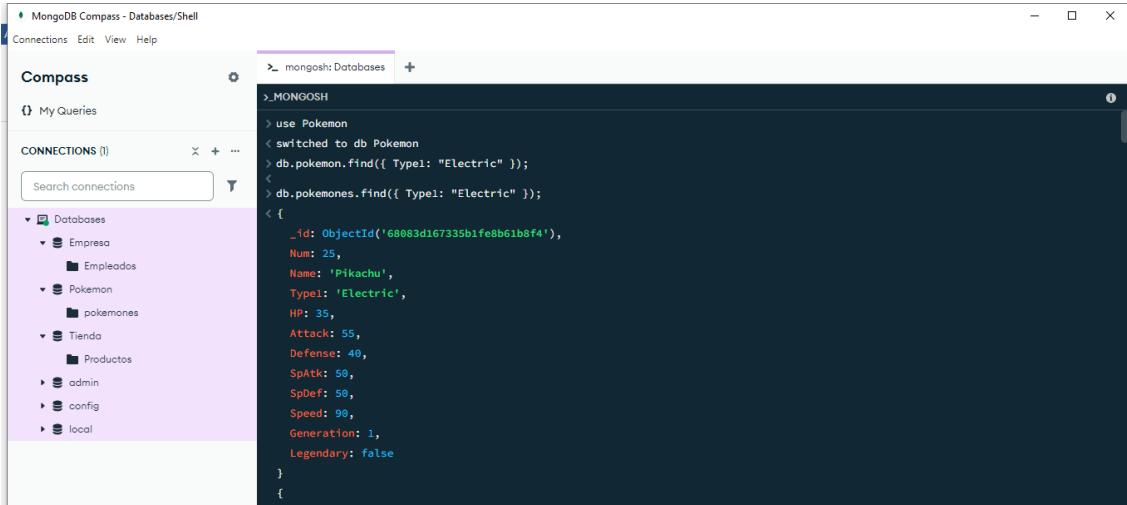


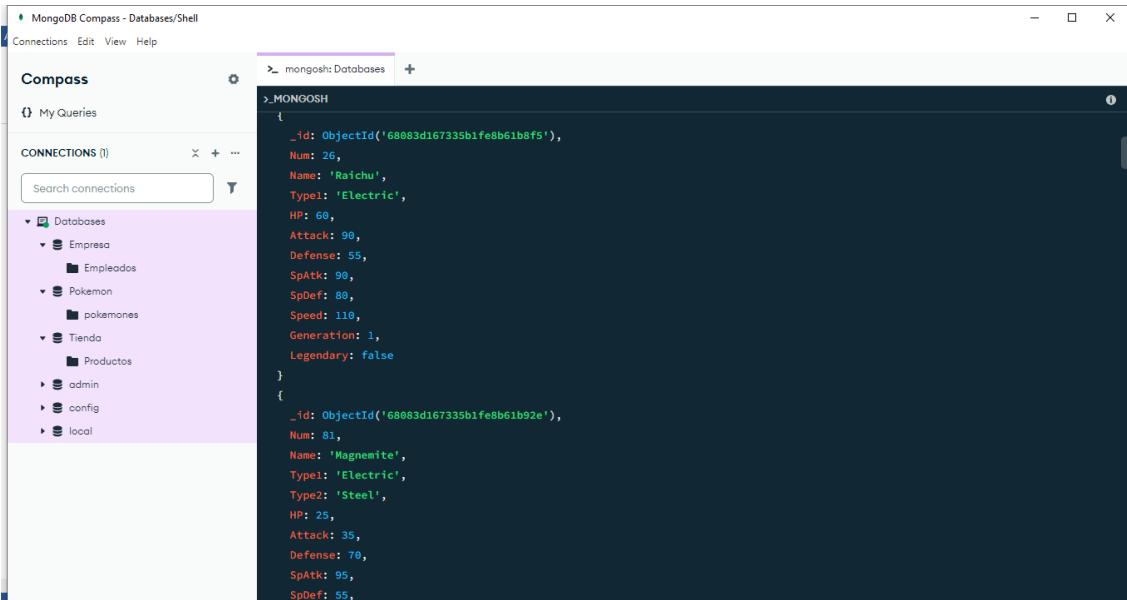
TALLER MONGODB con JSON

PARTE 1: CONSULTAS BASICAS

1. Encuentra todos los Pokémon de tipo “Electric”.



```
>_MONGOSH
> use Pokemon
< switched to db Pokemon
> db.pokemon.find({ Type: "Electric" });
<
> db.pokemones.find({ Type: "Electric" });
<
{
  _id: ObjectId('68083d167335b1fe8b61b8f4'),
  Num: 25,
  Name: 'Pikachu',
  Type1: 'Electric',
  HP: 35,
  Attack: 55,
  Defense: 40,
  SpAtk: 50,
  SpDef: 50,
  Speed: 90,
  Generation: 1,
  Legendary: false
}
{
```



```
>_MONGOSH
{
  _id: ObjectId('68083d167335b1fe8b61b8f5'),
  Num: 26,
  Name: 'Raichu',
  Type1: 'Electric',
  HP: 60,
  Attack: 90,
  Defense: 55,
  SpAtk: 90,
  SpDef: 88,
  Speed: 110,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('68083d167335b1fe8b61b92e'),
  Num: 81,
  Name: 'Magnemite',
  Type1: 'Electric',
  Type2: 'Steel',
  HP: 25,
  Attack: 35,
  Defense: 70,
  SpAtk: 95,
  SpDef: 55,
```

```
>_MONGOOSH
  Speed: 45,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('68083d167335b1fe8b61b92f'),
  Num: 82,
  Name: 'Magneton',
  Type1: 'Electric',
  Type2: 'Steel',
  HP: 50,
  Attack: 60,
  Defense: 95,
  SpAtk: 120,
  SpDef: 78,
  Speed: 78,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('68083d167335b1fe8b61b942'),
  Num: 100,
  Name: 'Voltorb',
  Type1: 'Electric',
  HP: 40,
  Attack: 38,
  Defense: 56,
  SpAtk: 55,
  SpDef: 55,
  Speed: 100,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('68083d167335b1fe8b61b943'),
  Num: 101,
  Name: 'Electrode',
  Type1: 'Electric',
  HP: 60,
  Attack: 50,
  Defense: 70,
  SpAtk: 88,
  SpDef: 88,
  Speed: 140,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('68083d167335b1fe8b61b95c'),
  Num: 125,
  Name: 'Electabuzz',
  Type1: 'Electric',
  HP: 65,
  Attack: 83,
  Defense: 57,
  SpAtk: 95,
  SpDef: 85,
  Speed: 105,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('68083d167335b1fe8b61b968'),
  Num: 135,
  Name: 'Jolteon',
  Type1: 'Electric',
  HP: 65,
  Attack: 65,
  Defense: 65,
  SpAtk: 100,
  SpDef: 85,
  Speed: 130,
  Generation: 1,
  Legendary: false
}
```

```
mongosh: Databases +  
>_MONGOSH  
spuel: 95,  
Speed: 130,  
Generation: 1,  
Legendary: false  
}  
{  
_id: ObjectId('68083d167335b1fe8b61b973'),  
Num: 145,  
Name: 'Zapdos',  
Type1: 'Electric',  
Type2: 'Flying',  
HP: 90,  
Attack: 90,  
Defense: 85,  
SpAtk: 125,  
SpDef: 98,  
Speed: 100,  
Generation: 1,  
Legendary: true  
}  
{  
_id: ObjectId('68083d167335b1fe8b61b998'),  
Num: 172,  
Name: 'Pichu',  
Type1: 'Electric',  
Type2: null,  
HP: 65,  
Attack: 40,  
Defense: 40,  
SpAtk: 65,  
SpDef: 45,  
Speed: 35,  
Generation: 1,  
Legendary: false  
}  
>_MONGOSH
```

```
mongosh: Databases +  
>_MONGOSH  
_id: ObjectId('68083d167335b1fe8b61b997'),  
Num: 179,  
Name: 'Mareep',  
Type1: 'Electric',  
Type2: null,  
HP: 55,  
Attack: 40,  
Defense: 40,  
SpAtk: 65,  
SpDef: 45,  
Speed: 35,  
Generation: 2,  
Legendary: false  
}  
{  
_id: ObjectId('68083d167335b1fe8b61b998'),  
Num: 180,  
Name: 'Flaaffy',  
Type1: 'Electric',  
Type2: null,  
HP: 70,  
Attack: 55,  
Defense: 55,  
SpAtk: 88,  
SpDef: 60,  
Speed: 45,  
Generation: 2,  
Legendary: false  
}  
>_MONGOSH
```

```
mongosh: Databases +  
>_MONGOSH  
_id: ObjectId('68083d167335b1fe8b61b997'),  
Num: 179,  
Name: 'Mareep',  
Type1: 'Electric',  
Type2: null,  
HP: 55,  
Attack: 40,  
Defense: 40,  
SpAtk: 65,  
SpDef: 45,  
Speed: 35,  
Generation: 2,  
Legendary: false  
}  
{  
_id: ObjectId('68083d167335b1fe8b61b998'),  
Num: 180,  
Name: 'Flaaffy',  
Type1: 'Electric',  
Type2: null,  
HP: 70,  
Attack: 55,  
Defense: 55,  
SpAtk: 88,  
SpDef: 60,  
Speed: 45,  
Generation: 2,  
Legendary: false  
}
```

```
>_MONGOSH
{
  "_id": ObjectId('68083d167335b1fe8b61b999'),
  "Num": 101,
  "Name": 'Ampharos',
  "Type1": 'Electric',
  "HP": 90,
  "Attack": 75,
  "Defense": 85,
  "SpAtk": 115,
  "SpDef": 98,
  "Speed": 55,
  "Generation": 2,
  "Legendary": false
}
{
  "_id": ObjectId('68083d167335b1fe8b61b99a'),
  "Num": 101,
  "Name": 'AmpharosMega Ampharos',
  "Type1": 'Electric',
  "Type2": 'Dragon',
  "HP": 99,
  "Attack": 95,
  "Defense": 105,
  "SpAtk": 165,
  "SpDef": 110,
```

```
>_MONGOSH
{
  "_id": ObjectId('68083d167335b1fe8b61b9d8'),
  "Num": 239,
  "Name": 'Elekid',
  "Type1": 'Electric',
  "HP": 45,
  "Attack": 63,
  "Defense": 37,
  "SpAtk": 65,
  "SpDef": 55,
  "Speed": 95,
  "Generation": 2,
  "Legendary": false
}
{
  "_id": ObjectId('68083d167335b1fe8b61b9dc'),
  "Num": 243,
  "Name": 'Raikou',
  "Type1": 'Electric',
  "HP": 99,
  "Attack": 85,
  "Defense": 75,
  "SpAtk": 115,
  "SpDef": 180,
  "Speed": 115,
```

```
>_MONGOSH
{
  "_id": ObjectId('68083d167335b1fe8b61ba27'),
  "Num": 309,
  "Name": 'Electrike',
  "Type1": 'Electric',
  "HP": 40,
  "Attack": 45,
  "Defense": 40,
  "SpAtk": 65,
  "SpDef": 48,
  "Speed": 65,
  "Generation": 3,
  "Legendary": false
}
{
  "_id": ObjectId('68083d167335b1fe8b61ba28'),
  "Num": 310,
  "Name": 'Manectric',
  "Type1": 'Electric',
  "HP": 70,
  "Attack": 75,
  "Defense": 60,
  "SpAtk": 105,
  "SpDef": 60,
```

```
mongosh: Databases/Shell
Connections Edit View Help
Compass
My Queries
CONNECTIONS () X + ...
Search connections
Databases
  Empresa
    Empleados
  Pokemon
    pokemones
  Tienda
    Productos
  admin
  config
  local

>_MONGOSH
{
  "_id": ObjectId("68083d167335b1fe8b61ba29"),
  "Num": 310,
  "Name": "ManectricMega Manectric",
  "Type1": "Electric",
  "HP": 70,
  "Attack": 75,
  "Defense": 80,
  "SpAtk": 135,
  "SpDef": 80,
  "Speed": 135,
  "Generation": 3,
  "Legendary": false
}
{
  "_id": ObjectId("68083d167335b1fe8b61ba2a"),
  "Num": 311,
  "Name": "Plusle",
  "Type1": "Electric",
  "HP": 60,
  "Attack": 50,
  "Defense": 40,
  "SpAtk": 85,
  "SpDef": 75,
  "Speed": 95,
  "Generation": 3,
```

2. Muestra solo los nombres y el ataque de los Pokémon con más de 100 de ataque

```
mongosh: Databases/Shell
Connections Edit View Help
Compass
My Queries
CONNECTIONS () X + ...
Search connections
Databases
  Empresa
    Empleados
  Pokemon
    pokemones
  Tienda
    Productos
  admin
  config
  local

>_MONGOSH
> db.pokemones.find(
  { Attack: { $gt: 100 } },
  { _id: 0, Name: 1, Attack: 1 }
)
< [
  {
    "Name": "CharizardMega Charizard X",
    "Attack": 130
  },
  {
    "Name": "CharizardMega Charizard Y",
    "Attack": 104
  },
  {
    "Name": "BlastoiseMega Blastoise",
    "Attack": 103
  },
  {
    "Name": "BeedrillMega Beedrill",
    "Attack": 150
  },
  {
    "Name": "Nidoking",
    "Attack": 102
  },
]
```

MongoDB Compass - Databases/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS ()

Search connections

Databases

- Empresa
- Empleados
- Pokemon
- pokemones

- Tienda
- Productos

- admin
- config
- local

_MONGOSH

```
>_MONGOSH
{
  {
    Name: 'Primeape',
    Attack: 105
  }
  {
    Name: 'Arcanine',
    Attack: 110
  }
  {
    Name: 'Machamp',
    Attack: 130
  }
  {
    Name: 'Vicreebel',
    Attack: 105
  }
  {
    Name: 'Golem',
    Attack: 120
  }
  {
    Name: 'Dodrio',
    Attack: 110
  }
}
```

MongoDB Compass - Databases/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS ()

Search connections

Databases

- Empresa
- Empleados
- Pokemon
- pokemones

- Tienda
- Productos

- admin
- config
- local

_MONGOSH

```
>_MONGOSH
{
  {
    Name: 'Muk',
    Attack: 105
  }
  {
    Name: 'Krabby',
    Attack: 105
  }
  {
    Name: 'Kingler',
    Attack: 130
  }
  {
    Name: 'Hitmonlee',
    Attack: 120
  }
  {
    Name: 'Hitmonchan',
    Attack: 105
  }
  {
    Name: 'Rhydon',
    Attack: 130
  }
}
```

MongoDB Compass - Databases/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS ()

Search connections

Databases

- Empresa
- Empleados
- Pokemon
- pokemones

- Tienda
- Productos

- admin
- config
- local

_MONGOSH

```
>_MONGOSH
{
  {
    name: 'MrMime',
    Attack: 120
  }
  {
    Name: 'Hitmonchan',
    Attack: 105
  }
  {
    Name: 'Rhydon',
    Attack: 130
  }
  {
    Name: 'KangaskhanMega Kangaskhan',
    Attack: 125
  }
  {
    Name: 'Scyther',
    Attack: 110
  }
  {
    Name: 'Pinsir',
    Attack: 125
  }
}
Type "it" for more
```

3. Encuentra los Pokémon cuya defensa esté entre 80 y 100 (inclusive)

The screenshots show the MongoDB Compass interface with three separate queries running in the shell:

```
>_MONGOSH
> db.pokemones.find({
  Defense: { $gte: 80, $lte: 100 }
});
< [
  {
    _id: ObjectId('68083d167335b1fe8b61b8d8'),
    Num: 3,
    Name: 'Venusaur',
    Type1: 'Grass',
    Type2: 'Poison',
    HP: 88,
    Attack: 82,
    Defense: 83,
    SpAtk: 100,
    SpDef: 100,
    Speed: 88,
    Generation: 1,
    Legendary: false
  },
  {
    _id: ObjectId('68083d167335b1fe8b61b8e0'),
    Num: 8,
    Name: 'Wartortle',
    Type1: 'Water',
    HP: 59,
    Attack: 63,
    Defense: 88,
    SpAtk: 65,
    SpDef: 88,
    Speed: 58,
    Generation: 1,
    Legendary: false
  },
  {
    _id: ObjectId('68083d167335b1fe8b61b8e1'),
    Num: 9,
    Name: 'Blastoise',
    Type1: 'Water',
    HP: 79,
    Attack: 83,
    Defense: 100,
    SpAtk: 85,
    SpDef: 105,
    Speed: 78,
    Defense: 100,
    SpAtk: 85,
    SpDef: 105,
    Speed: 78,
    Generation: 1,
    Legendary: false
  },
  {
    _id: ObjectId('68083d167335b1fe8b61b8ed'),
    Num: 18,
    Name: 'PidgeotMega Pidgeot',
    Type1: 'Normal',
    Type2: 'Flying',
    HP: 83,
    Attack: 88,
    Defense: 80,
    SpAtk: 135,
    SpDef: 88,
    Speed: 121,
    Generation: 1,
    Legendary: false
  }
]
```

```
>_MONGOSH
{
  {
    _id: ObjectId('68083d167335b1fe8b61b8f6'),
    Num: 27,
    Name: 'Sandshrew',
    Type1: 'Ground',
    HP: 58,
    Attack: 75,
    Defense: 85,
    SpAtk: 28,
    SpDef: 38,
    Speed: 40,
    Generation: 1,
    Legendary: false
  }
  {
    _id: ObjectId('68083d167335b1fe8b61b8fa'),
    Num: 31,
    Name: 'Nidoqueen',
    Type1: 'Poison',
    Type2: 'Ground',
    HP: 98,
    Attack: 92,
    Defense: 87,
    SpAtk: 75,
    SpDef: 85,
    Speed: 76,
    Generation: 1,
    Legendary: false
  }
  {
    _id: ObjectId('68083d167335b1fe8b61b8fa'),
    Num: 31,
    Name: 'Nidoqueen',
    Type1: 'Poison',
    Type2: 'Ground',
    HP: 98,
    Attack: 92,
    Defense: 87,
    SpAtk: 75,
    SpDef: 85,
    Speed: 76,
    Generation: 1,
    Legendary: false
  }
  {
    _id: ObjectId('68083d167335b1fe8b61b908'),
    Num: 45,
    Name: 'Vileplume',
    Type1: 'Grass',
    Type2: 'Poison',
    HP: 75,
    Attack: 80,
    Defense: 85,
    SpAtk: 110,
    SpDef: 90,
    Speed: 50,
    Generation: 1,
    Legendary: false
  }
  {
    _id: ObjectId('68083d167335b1fe8b61b90a'),
    Num: 47,
    Name: 'Parasect',
    Type1: 'Bug',
    Type2: 'Grass',
    HP: 60,
    Attack: 95,
    Defense: 80,
    SpAtk: 60,
    SpDef: 80,
    Speed: 30,
    Generation: 1,
    Legendary: false
  }
  {
```

```
>_MONGOSH
{
  "_id": ObjectId('68083d167335b1fe8b61b916'),
  "Num": 59,
  "Name": "Arcanine",
  "Type1": "Fire",
  "HP": 90,
  "Attack": 110,
  "Defense": 80,
  "SpAtk": 100,
  "SpDef": 88,
  "Speed": 95,
  "Generation": 1,
  "Legendary": false
}
{
  "_id": ObjectId('68083d167335b1fe8b61b919'),
  "Num": 62,
  "Name": "Poliwrath",
  "Type1": "Water",
  "Type2": "Fighting",
  "HP": 90,
  "Attack": 95,
  "Defense": 95,
  "SpAtk": 70,
  "SpDef": 90,
  "Generation": 1,
  "Legendary": false
}
{
  "_id": ObjectId('68083d167335b1fe8b61b920'),
  "Num": 68,
  "Name": "Machamp",
  "Type1": "Fighting",
  "HP": 90,
  "Attack": 130,
  "Defense": 80,
  "SpAtk": 65,
  "SpDef": 85,
  "Speed": 55,
  "Generation": 1,
  "Legendary": false
}
{
  "_id": ObjectId('68083d167335b1fe8b61b926'),
  "Num": 74,
  "Name": "Geodude",
  "Type1": "Rock",
  "Type2": "Ground",
  "HP": 40,
  "Attack": 80,
  "Defense": 100,
  "SpAtk": 30,
  "SpDef": 30,
  "Speed": 20,
  "Generation": 1,
  "Legendary": false
}
{
  "_id": ObjectId('68083d167335b1fe8b61b92f'),
  "Num": 82,
  "Name": "Magneton",
  "Type1": "Electric",
  "Type2": "Steel",
  "HP": 50,
  "Attack": 60,
  "Defense": 95,
  "SpAtk": 120,
  "SpDef": 70,
  "Generation": 1,
  "Legendary": false
}
```

```
>_MONGOSH
  {
    _id: ObjectId('68083d167335b1fe8b61b937'),
    Num: 87,
    Name: 'Dewgong',
    Type1: 'Water',
    Type2: 'Ice',
    HP: 90,
    Attack: 70,
    Defense: 80,
    SpAtk: 70,
    SpDef: 95,
    Speed: 70,
    Generation: 1,
    Legendary: false
  }
}

>_MONGOSH
  {
    _id: ObjectId('68083d167335b1fe8b61b937'),
    Num: 98,
    Name: 'Shellder',
    Type1: 'Water',
    HP: 30,
    Attack: 65,
    Defense: 100,
    SpAtk: 45,
    SpDef: 25,
    Speed: 40,
    Generation: 1,
    Legendary: false
  }
{
  _id: ObjectId('68083d167335b1fe8b61b93c'),
  Num: 94,
  Name: 'GengarMega Gengar',
  Type1: 'Ghost',
  Type2: 'Poison',
  HP: 68,
  Attack: 65,
  Defense: 80,
  SpAtk: 170,
  SpDef: 95,
  Speed: 130,
}

>_MONGOSH
  {
    _id: ObjectId('68083d167335b1fe8b61b940'),
    Num: 98,
    Name: 'Krabby',
    Type1: 'Water',
    HP: 30,
    Attack: 105,
    Defense: 90,
    SpAtk: 25,
    SpDef: 25,
    Speed: 50,
    Generation: 1,
    Legendary: false
  }
{
  _id: ObjectId('68083d167335b1fe8b61b944'),
  Num: 102,
  Name: 'Exeggute',
  Type1: 'Grass',
  Type2: 'Psychic',
  HP: 68,
  Attack: 40,
  Defense: 80,
  SpAtk: 60,
  SpDef: 45,
```

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists databases: Empresa, Pokemon, and Tienda. The main panel displays the contents of the '_MONGOSH' collection. Two documents are shown:

```
_id: ObjectId('68083d167335b1fe8b61b945'),  
Num: 103,  
Name: 'Exeggutor',  
Type1: 'Grass',  
Type2: 'Psychic',  
HP: 95,  
Attack: 95,  
Defense: 85,  
SpAtk: 125,  
SpDef: 65,  
Speed: 55,  
Generation: 1,  
Legendary: false}  
{  
_id: ObjectId('68083d167335b1fe8b61b946'),  
Num: 104,  
Name: 'Cubone',  
Type1: 'Ground',  
HP: 50,  
Attack: 50,  
Defense: 95,  
SpAtk: 40,  
SpDef: 50,  
Speed: 40}
```

PARTE 2: AGREGACIONES

4. Muestra el promedio de ataque de los Pokémon por tipo (Type1)

The screenshot shows the MongoDB Compass interface. The 'Connections' sidebar lists databases: Empresa, Pokemon, and Tienda. The main panel shows the aggregation command in the mongo shell:

```
>_MONGOSH  
|> db.pokemones.aggregate([  
  {  
    $group: {  
      _id: "$Type1",  
      promedioAtaque: { $avg: "$Attack" }  
    }  
  },  
  {  
    $sort: { promedioAtaque: -1 } // Opcional, para ver los más fuertes primero  
  }  
]);  
< {  
  _id: 'Dragon',  
  promedioAtaque: 112.125  
}  
{  
  _id: 'Fighting',  
  promedioAtaque: 96.77777777777777  
}  
{  
  _id: 'Ground',  
  promedioAtaque: 95.75  
}
```

```

>_MONGOASH
{
  "_id: 'Rock',
  promedioAtaque: 92.86363636363636
}
{
  "_id: 'Steel',
  promedioAtaque: 92.70370370370371
}
{
  "_id: 'Dark',
  promedioAtaque: 88.38709677419355
}
{
  "_id: 'Fire',
  promedioAtaque: 84.76923076923077
}
{
  "_id: 'Flying',
  promedioAtaque: 78.75
}
{
  "_id: 'Poison',
  promedioAtaque: 74.67857142857143
}
}

>_MONGOSH
{
  "_id: 'Water',
  promedioAtaque: 74.15178571428571
}
{
  "_id: 'Ghost',
  promedioAtaque: 73.78125
}
{
  "_id: 'Normal',
  promedioAtaque: 73.46938775510205
}
{
  "_id: 'Grass',
  promedioAtaque: 73.21428571428571
}
{
  "_id: 'Ice',
  promedioAtaque: 72.75
}
{
  "_id: 'Psychic',
  promedioAtaque: 71.45614035087719
}
{
  "_id: 'Bug',
  promedioAtaque: 70.97101449275362
}
{
  "_id: 'Electric',
  promedioAtaque: 69.09090909090909
}
{
  "_id: 'Fairy',
  promedioAtaque: 61.529411764785884
}

```

5. Encuentra el Pokémon con más HP de cada tipo

MongoDB Compass - Databases/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS ()

Search connections

Databases

- Empresa
 - Empleados
- Pokemon
 - pokemones
- Tienda
 - Productos
- admin
- config
- local

MongoDB Compass - Databases/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS ()

Search connections

Databases

- Empresa
 - Empleados
- Pokemon
 - pokemones
- Tienda
 - Productos
- admin
- config
- local

MongoDB Compass - Databases/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS ()

Search connections

Databases

- Empresa
 - Empleados
- Pokemon
 - pokemones
- Tienda
 - Productos
- admin
- config
- local

_MONGOOSH

```
> db.pokemones.aggregate([
  {
    $sort: { HP: -1 }
  },
  {
    $group: {
      _id: "$Type1",
      PokemonConMasHP: { $first: "$Name" },
      HP: { $first: "$HP" }
    }
  });
< [
  {
    _id: 'Fighting',
    PokemonConMasHP: 'Hariyama',
    HP: 144
  },
  {
    _id: 'Psychic',
    PokemonConMasHP: 'Wobbuffet',
    HP: 190
  },
  {
    _id: 'Normal',
    PokemonConMasHP: 'Blissey',
    HP: 255
  },
  {
    _id: 'Dragon',
    PokemonConMasHP: 'KyuremBlack Kyurem',
    HP: 125
  },
  {
    _id: 'Poison',
    PokemonConMasHP: 'Muk',
    HP: 105
  },
  {
    _id: 'Ground',
    PokemonConMasHP: 'Rhyperior',
    HP: 115
  },
  {
    _id: 'Steel',
    PokemonConMasHP: 'Dialga',
    HP: 100
  }
]
```

_MONGOSH

```
> MONGOSH
{
  _id: 'Bug',
  PokemonConMasHP: 'Yanmega',
  HP: 86
},
{
  _id: 'Grass',
  PokemonConMasHP: 'Gogoat',
  HP: 123
},
{
  _id: 'Fire',
  PokemonConMasHP: 'Entei',
  HP: 115
},
{
  _id: 'Dark',
  PokemonConMasHP: 'Yveltal',
  HP: 126
},
{
  _id: 'Fairy',
  PokemonConMasHP: 'Xerneas',
  HP: 126
}
```

```

>_MONGOOSH
{
    "_id": "Electric",
    "PokemonConMasHP": "Ampharos",
    "HP": 99
}
{
    "_id": "Water",
    "PokemonConMasHP": "Wailord",
    "HP": 176
}
{
    "_id": "Ghost",
    "PokemonConMasHP": "Dribblet",
    "HP": 150
}
{
    "_id": "Rock",
    "PokemonConMasHP": "Aurorus",
    "HP": 123
}
{
    "_id": "Ice",
    "PokemonConMasHP": "Walrein",
    "HP": 110
}
{
    "_id": "Flying",
    "PokemonConMasHP": "Noivern",
    "HP": 85
}

```

6. Muestra los 5 Pokémon más rápidos

```

>_MONGOOSH
> db.pokemones.find(
    {},
    { _id: 0, Name: 1, Speed: 1 }
).sort({ Speed: -1 }).limit(5);
< [
    {
        Name: 'DeoxysSpeed Forme',
        Speed: 180
    },
    {
        Name: 'Ninjask',
        Speed: 160
    },
    {
        Name: 'AlakazamMega Alakazam',
        Speed: 150
    },
    {
        Name: 'DeoxysAttack Forme',
        Speed: 150
    },
    {
        Name: 'DeoxysNormal Forme',
        Speed: 150
    }
]

```

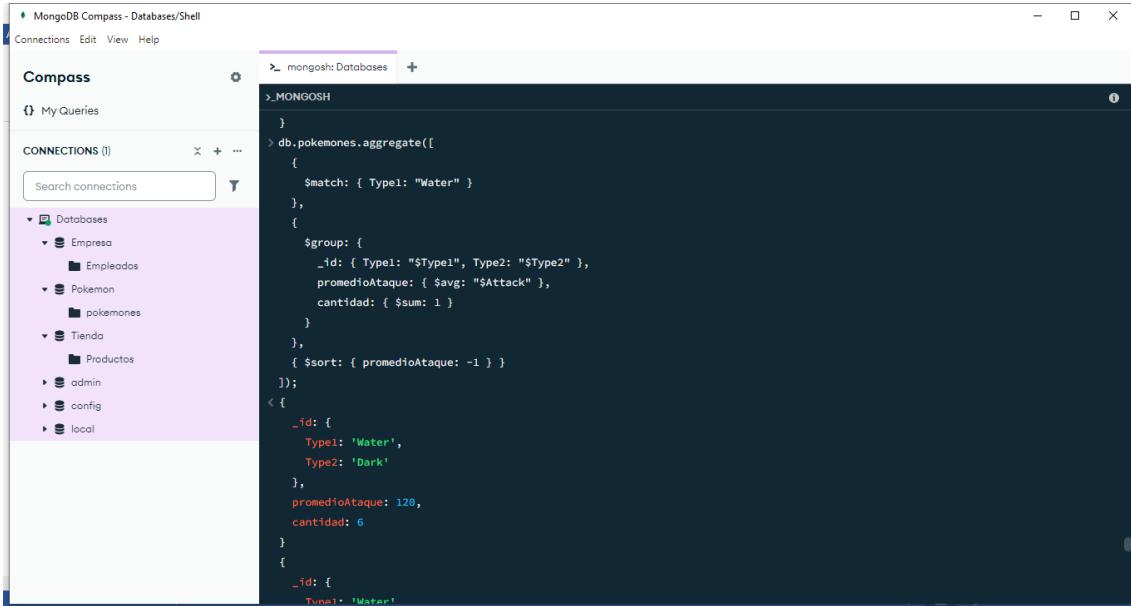
PARTE 3: COMBINACIONES DE \$MATCH, \$GROUP Y \$SORT

7. Muestra el promedio de ataque de los Pokémon tipo “Water”, ordenado de mayor a menor

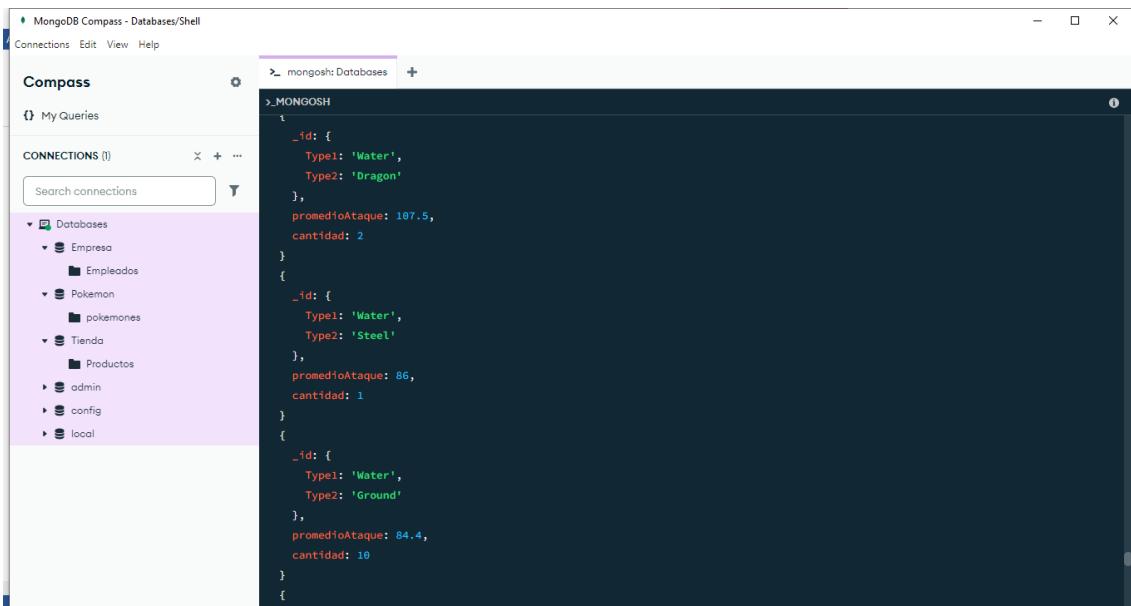


```
> db.pokemones.aggregate([
  { $match: { Type1: "Water" } },
  {
    $group: {
      _id: "$Type1",
      promedioAtaque: { $avg: "$Attack" }
    }
  },
  { $sort: { promedioAtaque: -1 } } // Aunque haya un solo resultado
]);
< [
  {
    _id: 'Water',
    promedioAtaque: 74.15178571428571
  }
]
```

Ordenado



```
>_MONGOSH
}
> db.pokemones.aggregate([
  {
    $match: { Type1: "Water" }
  },
  {
    $group: {
      _id: { Type1: "$Type1", Type2: "$Type2" },
      promedioAtaque: { $avg: "$Attack" },
      cantidad: { $sum: 1 }
    }
  },
  { $sort: { promedioAtaque: -1 } }
]);
< [
  {
    _id: {
      Type1: 'Water',
      Type2: 'Dark'
    },
    promedioAtaque: 120,
    cantidad: 6
  }
]
```



```
>_MONGOSH
{
  _id: {
    Type1: 'Water',
    Type2: 'Dragon'
  },
  promedioAtaque: 107.5,
  cantidad: 2
}
{
  _id: {
    Type1: 'Water',
    Type2: 'Steel'
  },
  promedioAtaque: 86,
  cantidad: 1
}
{
  _id: {
    Type1: 'Water',
    Type2: 'Ground'
  },
  promedioAtaque: 84.4,
  cantidad: 10
}
```

```
mongosh> _MONGOOSH
{
  "_id": {
    "Type1: "Water",
    "Type2: "Ice"
  },
  "promedioAtaque: 83.33333333333333,
  cantidad: 3
}

{
  "_id": {
    "Type1: "Water",
    "Type2: "Rock"
  },
  "promedioAtaque: 82.75,
  cantidad: 4
}

{
  "_id": {
    "Type1: "Water",
    "Type2: "Fighting"
  },
  "promedioAtaque: 79.66666666666667,
  cantidad: 3
}
```

```
mongosh> _MONGOOSH
{
  "_id": {
    "Type1: "Water",
    "Type2: "Psychic"
  },
  "promedioAtaque: 73,
  cantidad: 5
}

{
  "_id": {
    "Type1: "Water"
  },
  "promedioAtaque: 71.83050847457628,
  cantidad: 59
}

{
  "_id": {
    "Type1: "Water",
    "Type2: "Poison"
  },
  "promedioAtaque: 68.33333333333333,
  cantidad: 3
}
```

```
mongosh> _MONGOOSH
{
  "_id": {
    "Type1: "Water",
    "Type2: "Flying"
  },
  "promedioAtaque: 56.57142857142857,
  cantidad: 7
}

{
  "_id": {
    "Type1: "Water",
    "Type2: "Ghost"
  },
  "promedioAtaque: 58,
  cantidad: 2
}

{
  "_id": {
    "Type1: "Water",
    "Type2: "Grass"
  },
  "promedioAtaque: 59,
  cantidad: 3
}
```

The screenshot shows the MongoDB Compass interface. The left sidebar displays a tree view of connections and databases, with 'Empresas' selected. The main pane shows a list of documents under the 'Database' collection. Each document has an '_id' field containing a ObjectId, a 'Type1' field, a 'Type2' field, a 'promedioAtaque' field, and a 'cantidad' field.

_id	Type1	Type2	promedioAtaque	cantidad
{...}	Water	Grass	50	3
{...}	Water	Electric	48	2
{...}	Water	Fairy	35	2

8. Encuentra el Pokémon con más ataque por generación y ordénalos de mayor a menor.

The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays a tree view of databases: Databases (selected), Empresa, Pokemon, Tienda, admin, config, and local. The main area is titled '_MONGOSH' and contains the following MongoDB shell command:

```
_MONGOSH
}
> db.pokemones.aggregate([
  { $sort: { Attack: -1 } },
  {
    $group: {
      _id: "$Generation",
      PokemonConMasAtaque: { $first: "$Name" },
      Ataque: { $first: "$Attack" }
    }
  },
  { $sort: { Ataque: -1 } }
]);
< {
  _id: 1,
  PokemonConMasAtaque: 'MewtwoMega Mewtwo X',
  Ataque: 190
}
{
  _id: 2,
  PokemonConMasAtaque: 'HeracrossMega Heracross',
  Ataque: 185
}
t
```

```

>_MONGOOSH
{
  "_id": 2,
  "PokemonConMasAtaque": "HeracrossMega Heracross",
  "Ataque": 185
}
{
  "_id": 3,
  "PokemonConMasAtaque": "RayquazaMega Rayquaza",
  "Ataque": 180
}
{
  "_id": 4,
  "PokemonConMasAtaque": "GarchompMega Garchomp",
  "Ataque": 170
}
{
  "_id": 5,
  "PokemonConMasAtaque": "KyuremBlack Kyurem",
  "Ataque": 170
}
{
  "_id": 6,
  "PokemonConMasAtaque": "DiancieMega Diancie",
  "Ataque": 160
}

```

PARTE 4: INDEXACIÓN Y RENDIMIENTO

9. Crea un índice en el campo Type1

```

> db.pokemons.createIndex({ Type1: 1 })
< Type1_1
> db.pokemons.createIndex({ Type1: 1 });
< Type1_1

```

The screenshot shows the MongoDB Compass interface with the 'pokemon' collection selected. The document list displays two entries: Bulbasaur and Ivysaur.

_id	Num	Name	Type1	Type2	HP	Attack	Defense	SpA	SpD	Speed	Generation	Legendary
ObjectId('68083d167335b1fe8b61b8d6')	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	false
ObjectId('68083d167335b1fe8b61b8d7')	2	Ivysaur	Grass	Poison	60	62	63	80	80	65	1	false

MongoDB Compass - Databases/Pokemon.pokemones

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS [1]

Search connections

Databases

- Empresa
- Empleados
- Pokemon

 - pokemones

- Tienda
- Productos
- admin
- config
- local

mongosh:Databases pokemones +

Databases > Pokemon > pokemones

Documents 800 Aggregations Schema Indexes Validation

Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

25 1 - 25 of 800

`_id: ObjectId('68003d167335b1fe8b61b8d8')`
Num : 3
Name : "Venusaur"
Type1 : "Grass"
Type2 : "Poison"
HP : 88
Attack : 82
Defense : 83
SpAtk : 100
SpDef : 100
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8d9')`
Num : 3
Name : "VenusaurMega Venusaur"
Type1 : "Grass"
Type2 : "Poison"
HP : 88
Attack : 100
Defense : 123
SpAtk : 122
SpDef : 128
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8d9')`
Num : 3
Name : "VenusaurMega Venusaur"
Type1 : "Grass"
Type2 : "Poison"
HP : 88
Attack : 100
Defense : 123
SpAtk : 122
SpDef : 128
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8da')`
Num : 4
Name : "Charmander"
Type1 : "Fire"
HP : 39
Attack : 52
Defense : 43
SpAtk : 60
SpDef : 50

`_id: ObjectId('68003d167335b1fe8b61b8db')`
Num : 5
Name : "Charmeleon"
Type1 : "Fire"
HP : 58
Attack : 64
Defense : 58
SpAtk : 80
SpDef : 65
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8dc')`
Num : 6
Name : "Charizard"
Type1 : "Fire"
Type2 : "Flying"
HP : 78
Attack : 84
Defense : 78
SpAtk : 109
SpDef : 85

MongoDB Compass - Databases/Pokemon.pokemones

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS [1]

Search connections

Databases

- Empresa
- Empleados
- Pokemon

 - pokemones

- Tienda
- Productos
- admin
- config
- local

mongosh:Databases pokemones +

Databases > Pokemon > pokemones

Documents 800 Aggregations Schema Indexes Validation

Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

25 1 - 25 of 800

`_id: ObjectId('68003d167335b1fe8b61b8d8')`
Num : 3
Name : "Venusaur"
Type1 : "Grass"
Type2 : "Poison"
HP : 88
Attack : 82
Defense : 83
SpAtk : 100
SpDef : 100
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8d9')`
Num : 3
Name : "VenusaurMega Venusaur"
Type1 : "Grass"
Type2 : "Poison"
HP : 88
Attack : 100
Defense : 123
SpAtk : 122
SpDef : 128
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8d9')`
Num : 3
Name : "VenusaurMega Venusaur"
Type1 : "Grass"
Type2 : "Poison"
HP : 88
Attack : 100
Defense : 123
SpAtk : 122
SpDef : 128
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8da')`
Num : 4
Name : "Charmander"
Type1 : "Fire"
HP : 39
Attack : 52
Defense : 43
SpAtk : 60
SpDef : 50

`_id: ObjectId('68003d167335b1fe8b61b8db')`
Num : 5
Name : "Charmeleon"
Type1 : "Fire"
HP : 58
Attack : 64
Defense : 58
SpAtk : 80
SpDef : 65
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8dc')`
Num : 6
Name : "Charizard"
Type1 : "Fire"
Type2 : "Flying"
HP : 78
Attack : 84
Defense : 78
SpAtk : 109
SpDef : 85

MongoDB Compass - Databases/Pokemon.pokemones

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS [1]

Search connections

Databases

- Empresa
- Empleados
- Pokemon

 - pokemones

- Tienda
- Productos
- admin
- config
- local

mongosh:Databases pokemones +

Databases > Pokemon > pokemones

Documents 800 Aggregations Schema Indexes Validation

Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

25 1 - 25 of 800

`_id: ObjectId('68003d167335b1fe8b61b8d8')`
Num : 3
Name : "Venusaur"
Type1 : "Grass"
Type2 : "Poison"
HP : 88
Attack : 82
Defense : 83
SpAtk : 100
SpDef : 100
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8d9')`
Num : 3
Name : "VenusaurMega Venusaur"
Type1 : "Grass"
Type2 : "Poison"
HP : 88
Attack : 100
Defense : 123
SpAtk : 122
SpDef : 128
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8d9')`
Num : 3
Name : "VenusaurMega Venusaur"
Type1 : "Grass"
Type2 : "Poison"
HP : 88
Attack : 100
Defense : 123
SpAtk : 122
SpDef : 128
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8da')`
Num : 4
Name : "Charmander"
Type1 : "Fire"
HP : 39
Attack : 52
Defense : 43
SpAtk : 60
SpDef : 50

`_id: ObjectId('68003d167335b1fe8b61b8db')`
Num : 5
Name : "Charmeleon"
Type1 : "Fire"
HP : 58
Attack : 64
Defense : 58
SpAtk : 80
SpDef : 65
Speed : 88
Generation : 1
Legendary : false

`_id: ObjectId('68003d167335b1fe8b61b8dc')`
Num : 6
Name : "Charizard"
Type1 : "Fire"
Type2 : "Flying"
HP : 78
Attack : 84
Defense : 78
SpAtk : 109
SpDef : 85

MongoDB Compass - Databases/Pokemon.pokemones

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (1)

Search connections

Databases > Pokemon > pokemones

Documents 800 Aggregations Schema Indexes Validation

Type a query: { field: 'value' } or [Generate query](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

1 25 of 800

`_id: ObjectId('68083d167335b1fe8b61b8dc')`
Num : 6
Name : "Charizard"
Type1 : "Fire"
Type2 : "Flying"
HP : 78
Attack : 84
Defense : 78
SpAtk : 108
SpDef : 85
Speed : 100
Generation : 1
Legendary : false

`_id: ObjectId('68083d167335b1fe8b61b8dd')`
Num : 6
Name : "CharizardMega Charizard X"
Type1 : "Fire"
Type2 : "Dragon"
HP : 78
Attack : 130
Defense : 111
SpAtk : 130

MongoDB Compass - Databases/Pokemon.pokemones

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (1)

Search connections

Databases > Pokemon > pokemones

Documents 800 Aggregations Schema Indexes Validation

Type a query: { field: 'value' } or [Generate query](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

1 25 of 800

`_id: ObjectId('68083d167335b1fe8b61b8dd')`
Num : 6
Name : "CharizardMega Charizard X"
Type1 : "Fire"
Type2 : "Dragon"
HP : 78
Attack : 130
Defense : 111
SpAtk : 130
SpDef : 85
Speed : 100
Generation : 1
Legendary : false

`_id: ObjectId('68083d167335b1fe8b61b8de')`
Num : 6
Name : "CharizardMega Charizard Y"
Type1 : "Fire"
Type2 : "Flying"
HP : 78
Attack : 104
Defense : 78

MongoDB Compass - Databases/Pokemon.pokemones

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (1)

Search connections

Databases > Pokemon > pokemones

Documents 800 Aggregations Schema Indexes Validation

Type a query: { field: 'value' } or [Generate query](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

1 25 of 800

`_id: ObjectId('68083d167335b1fe8b61b8de')`
Num : 6
Name : "CharizardMega Charizard Y"
Type1 : "Fire"
Type2 : "Flying"
HP : 78
Attack : 104
Defense : 78
SpAtk : 159
SpDef : 115
Speed : 100
Generation : 1
Legendary : false

`_id: ObjectId('68083d167335b1fe8b61b8df')`
Num : 7
Name : "Squirtle"
Type1 : "Water"
HP : 44
Attack : 48
Defense : 65
SpAtk : 50
SpDef : 60

The screenshot shows the MongoDB Compass interface. On the left, the connection tree shows databases like Empresa, Pokemon, and Tienda. The main area is focused on the 'pokemon' collection within the 'Pokemon' database. It lists 800 documents. A document preview shows the following fields:

```
_id: ObjectId('68083d167335b1fe8b61b8df')
Num: 7
Name: "Squirtle"
Type1: "Water"
HP: 44
Attack: 48
Defense: 65
SpAtk: 50
SpDef: 64
Speed: 43
Generation: 1
Legendary: false
```

Below this, another document is shown:

```
_id: ObjectId('68083d167335b1fe8b61b8e0')
Num: 8
Name: "Wartortle"
Type1: "Water"
HP: 59
Attack: 63
Defense: 88
SpAtk: 65
SpDef: 88
Speed: 58
Generation: 1
```

10. Usa explain() para analizar el rendimiento de una búsqueda

The screenshot shows the MongoDB Compass interface with the 'Shell' tab selected. In the shell, the following command is entered and executed:

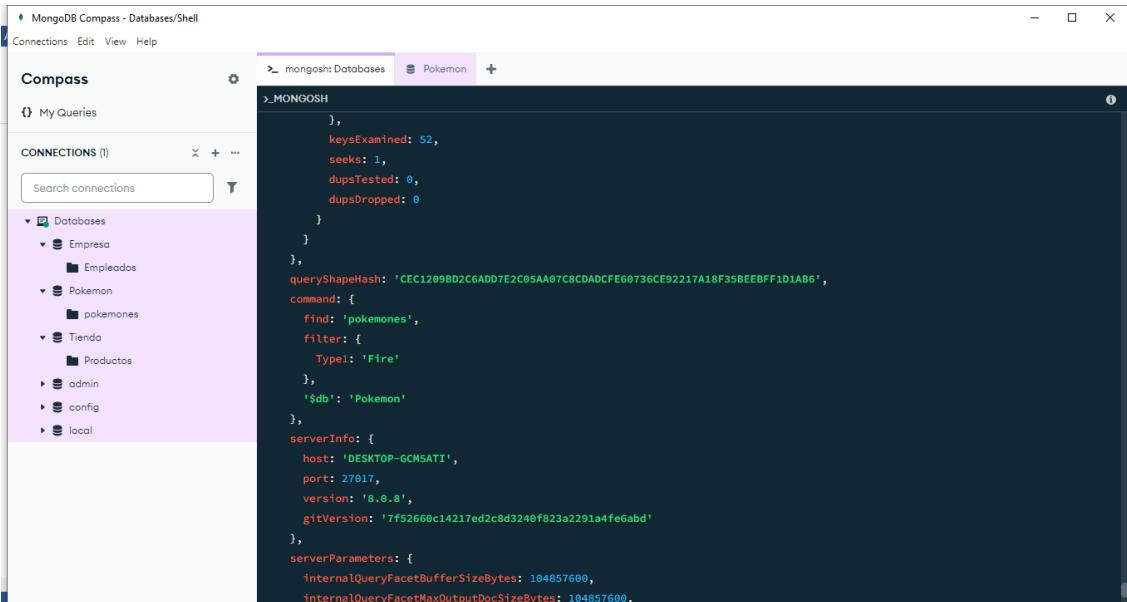
```
>_MONGOOSH
>`type -a`_
>db.pokemones.find({ Type1: "Fire" }).explain("executionStats")
< {
```

The output shows the execution plan for a query that finds all 'Fire' type pokemons. The plan includes details such as explainVersion, queryPlanner, parsedQuery, indexFilterSet, queryHash, planCacheShapeHash, planCacheKey, optimizationTimeMillis, maxIndexedOrSolutionsReached, maxIndexedAndSolutionsReached, maxScansToExplodeReached, prunedSimilarIndexes, winningPlan, isCached, stage, and inputStage.

```
>_MONGOSH
keyPattern: {
  Type1: 1
},
IndexName: 'Type1_1',
isMultiKey: false,
multiKeyPaths: [
  Type1: []
],
isUnique: false,
isSparse: false,
isPartial: false,
indexVersion: 2,
direction: 'forward',
indexBounds: {
  Type1: [
    ["Fire", "Fire"]
  ]
},
rejectedPlans: []
},
executionStats: {
  executionSuccess: true,
  nReturned: 52,
```

```
>_MONGOSH
executionTimeMillis: 6,
totalKeysExamined: 52,
totalDocsExamined: 52,
executionStages: {
  isCached: false,
  stage: 'FETCH',
  nReturned: 52,
  executionTimeMillisEstimate: 0,
  works: 53,
  advanced: 52,
  needTime: 0,
  needYield: 0,
  saveState: 0,
  restoreState: 0,
  isEOF: 1,
  docsExamined: 52,
  alreadyHasObj: 0,
  inputStage: {
    stage: 'IXSCAN',
    nReturned: 52,
    executionTimeMillisEstimate: 0,
    works: 53,
    advanced: 52,
    needTime: 0,
    needYield: 0
  }
},
```

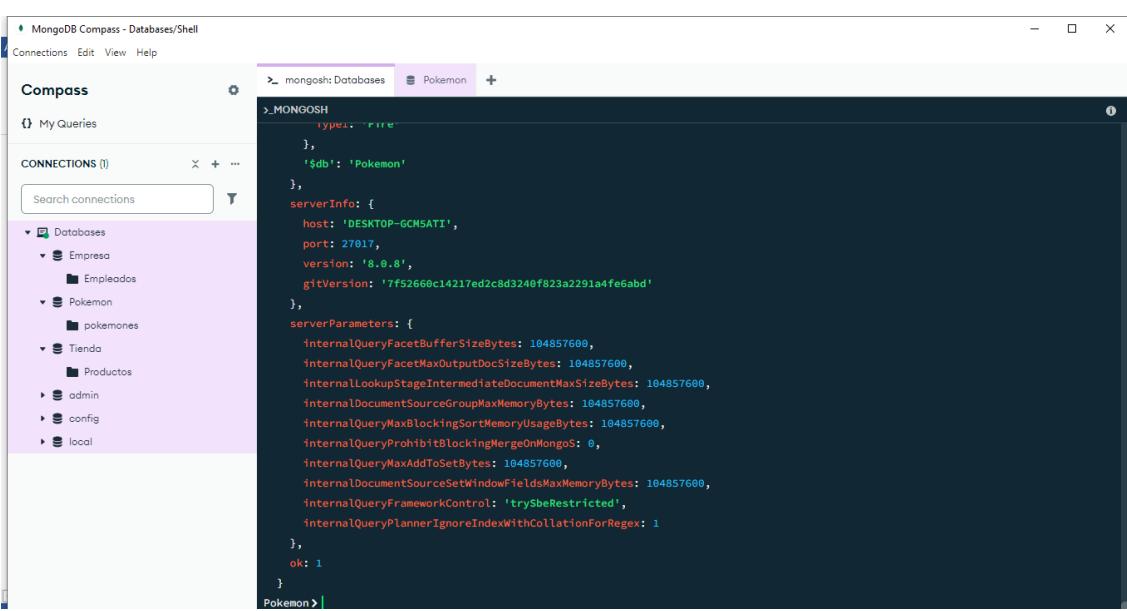
```
>_MONGOSH
needTime: 0,
needYield: 0,
saveState: 0,
restoreState: 0,
isEOF: 1,
keyPattern: {
  Type1: 1
},
IndexName: 'Type1_1',
isMultiKey: false,
multiKeyPaths: [
  Type1: []
],
isUnique: false,
isSparse: false,
isPartial: false,
indexVersion: 2,
direction: 'forward',
indexBounds: {
  Type1: [
    ["Fire", "Fire"]
  ]
},
keysExamined: 52,
seeks: 1,
```



```

>_MONGOSH
    },
    keysExamined: 52,
    seeks: 1,
    dupsTested: 0,
    dupsDropped: 0
  }
},
queryShapeHash: 'CEC1209BD2C6ADD7E2C05AA07C8CDADC60736CE92217A18F35BEEBFF1D1AB6',
command: {
  find: 'pokemones',
  filter: {
    Type1: 'Fire'
  },
  '$db': 'Pokemon'
},
serverInfo: {
  host: 'DESKTOP-GCH5ATI',
  port: 27017,
  version: '8.0.8',
  gitVersion: '7f52660c14217ed2c8d3240f823a2291a4fe6abd'
},
serverParameters: {
  internalQueryFacetBufferSizeBytes: 104857600,
  internalQueryFacetMaxOutputDocSizeBytes: 104857600
}

```

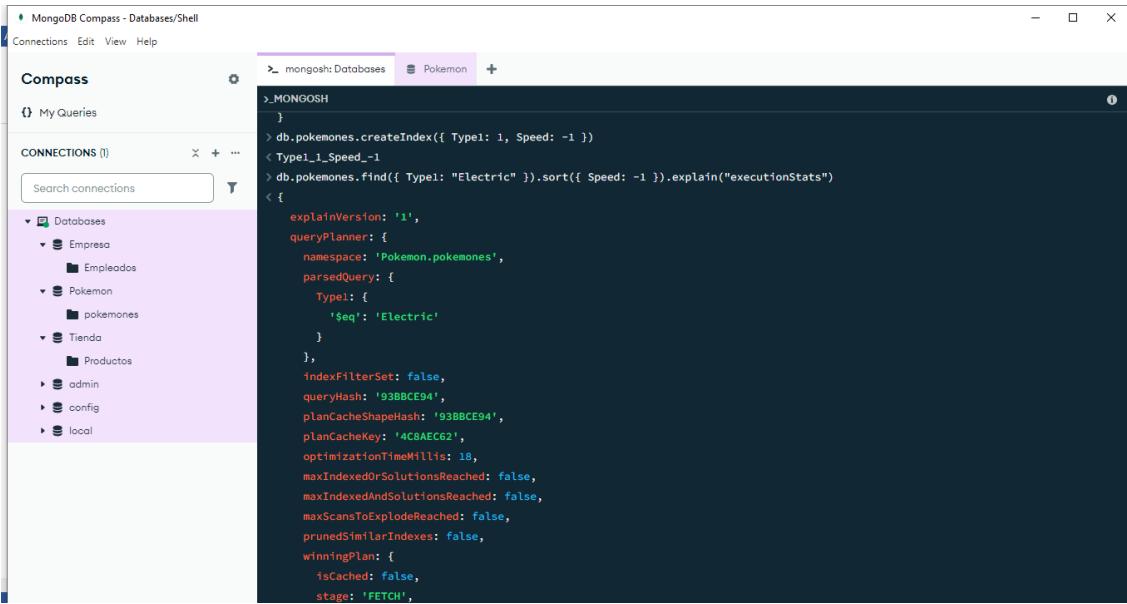



```

>_MONGOSH
  type: 'fire'
},
'$db': 'Pokemon'
},
serverInfo: {
  host: 'DESKTOP-GCH5ATI',
  port: 27017,
  version: '8.0.8',
  gitVersion: '7f52660c14217ed2c8d3240f823a2291a4fe6abd'
},
serverParameters: {
  internalQueryFacetBufferSizeBytes: 104857600,
  internalQueryFacetMaxOutputDocSizeBytes: 104857600,
  internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
  internalDocumentSourceGroupMaxMemoryBytes: 104857600,
  internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
  internalQueryProhibitBlockingMergeOnMongoS: 0,
  internalQueryMaxAddToSetBytes: 104857600,
  internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
  internalQueryFrameworkControl: 'trySbeRestricted',
  internalQueryPlannerIgnoreIndexWithCollationForRegex: 1
},
ok: 1
}

```

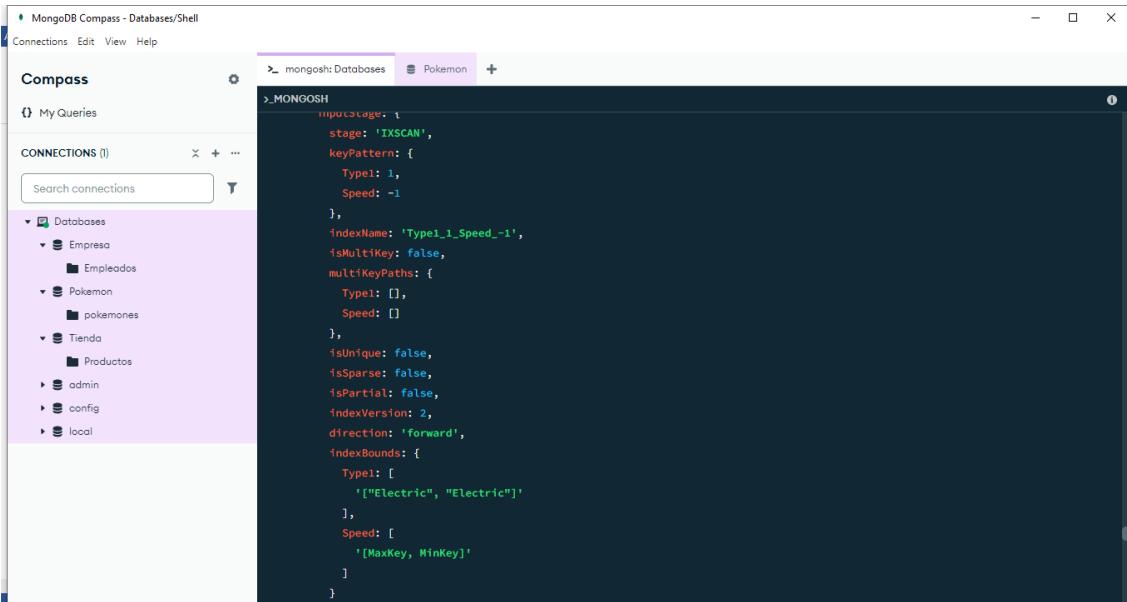
11. Crea un índice compuesto en Type1 y Speed, y analiza una búsqueda



The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays the connection list and database structure. The main panel shows the mongo shell command line. The command issued is:

```
_MONGOOSH
}
> db.pokemones.createIndex({ Type1: 1, Speed: -1 })
< Type1_1_Speed_-1
> db.pokemones.find({ Type1: "Electric" }).sort({ Speed: -1 }).explain("executionStats")
< {
  explainVersion: '1',
  queryPlanner: {
    namespace: 'Pokemon.pokemones',
    parsedQuery: {
      Type1: {
        '$eq': 'Electric'
      }
    },
    indexFilterSet: false,
    queryHash: '93BBCCE94',
    planCacheShapeHash: '93BBCCE94',
    planCacheKey: '4C8AEC62',
    optimizationTimeMillis: 18,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'FETCH'
    }
  }
}
```

- Ver los más rápidos de cada tipo



The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays the connection list and database structure. The main panel shows the mongo shell command line. The command issued is:

```
_MONGOSH
inputStage: {
  stage: 'IXSCAN',
  keyPattern: {
    Type1: 1,
    Speed: -1
  },
  indexName: 'Type1_1_Speed_-1',
  isMultiKey: false,
  multiKeyPaths: {
    Type1: [],
    Speed: []
  },
  isUnique: false,
  isSparse: false,
  isPartial: false,
  indexVersion: 2,
  direction: 'forward',
  indexBounds: {
    Type1: [
      '[{"Electric", "Electric"}]'
    ],
    Speed: [
      '[MaxKey, MinKey]'
    ]
  }
}
```

```
>_MONGOSH
    },
    rejectedPlans: [
    {
        isCached: false,
        stage: 'SORT',
        sortPattern: {
            Speed: -1
        },
        memLimit: 104857600,
        type: 'simple',
        inputStage: {
            stage: 'FETCH',
            inputStage: {
                stage: 'IXSCAN',
                keyPattern: {
                    TypeI: 1
                },
                indexName: 'TypeI_1',
                isMultiKey: false,
                multiKeyPaths: {
                    TypeI: []
                },
                isUnique: false,
                isSparse: false,
                isPartial: false,
                indexVersion: 2,
                direction: 'forward',
                indexBounds: {
                    TypeI: [
                        "[\"Electric\", \"Electric\"]"
                    ]
                }
            }
        }
    ],
    executionStats: {
        executionSuccess: true,
        nReturned: 44,
        executionTimeMillis: 19,
        totalKeysExamined: 44,
        totalDocsExamined: 44,
        executionStages: {
            isCached: false,
            stage: 'FETCH',
            nReturned: 44,
            executionTimeMillisEstimate: 10,
            works: 46,
            advanced: 44,
            needTime: 0,
            needYield: 0,
            saveState: 0,
            restoreState: 0,
            isEOF: 1,
            docsExamined: 44,
            alreadyHasObj: 0,
            inputStage: {
                stage: 'IXSCAN',
                nReturned: 44,
                executionTimeMillisEstimate: 10,
                works: 45,
                advanced: 44,
                needTime: 0,
                needYield: 0,
                saveState: 0,
                restoreState: 0,
                isEOF: 1,
                keyPattern: {
                    TypeI: 1
                }
            }
        }
    }
}
```

```
>_MONGOSH
    },
    isUnique: false,
    isSparse: false,
    isPartial: false,
    indexVersion: 2,
    direction: 'forward',
    indexBounds: {
        TypeI: [
            "[\"Electric\", \"Electric\"]"
        ]
    }
},
executionStats: {
    executionSuccess: true,
    nReturned: 44,
    executionTimeMillis: 19,
    totalKeysExamined: 44,
    totalDocsExamined: 44,
    executionStages: {
        isCached: false,
        stage: 'FETCH',
        nReturned: 44,
        executionTimeMillisEstimate: 10,
        works: 46,
        advanced: 44,
        needTime: 0,
        needYield: 0,
        saveState: 0,
        restoreState: 0,
        isEOF: 1,
        docsExamined: 44,
        alreadyHasObj: 0,
        inputStage: {
            stage: 'IXSCAN',
            nReturned: 44,
            executionTimeMillisEstimate: 10,
            works: 45,
            advanced: 44,
            needTime: 0,
            needYield: 0,
            saveState: 0,
            restoreState: 0,
            isEOF: 1,
            keyPattern: {
                TypeI: 1
            }
        }
    }
}
```

```
>_MONGOSH
    },
    stage: 'FETCH',
    nReturned: 44,
    executionTimeMillisEstimate: 10,
    works: 46,
    advanced: 44,
    needTime: 0,
    needYield: 0,
    saveState: 0,
    restoreState: 0,
    isEOF: 1,
    docsExamined: 44,
    alreadyHasObj: 0,
    inputStage: {
        stage: 'IXSCAN',
        nReturned: 44,
        executionTimeMillisEstimate: 10,
        works: 45,
        advanced: 44,
        needTime: 0,
        needYield: 0,
        saveState: 0,
        restoreState: 0,
        isEOF: 1,
        keyPattern: {
            TypeI: 1
        }
    }
}
```

```
_MONGOOSH
    isEOF: 1,
    keyPattern: {
        Type: 1,
        Speed: -1
    },
    indexName: 'Type1_Speed_-1',
    isMultiKey: false,
    multiKeyPaths: [
        {
            Type: [],
            Speed: []
        },
        {
            isUnique: false,
            isSparse: false,
            isPartial: false,
            indexVersion: 2,
            direction: 'forward',
            indexBounds: {
                Type: [
                    ['[Electric', 'Electric"]'
                ],
                Speed: [
                    '[MaxKey, MinKey]'
                ]
            }
        }
    ],
    _id: 1
}
```

```
_MONGOOSH
    j,
    keysExamined: 44,
    seeks: 1,
    dupesTested: 0,
    dupesDropped: 0
}
},
queryShapeHash: '05A13AD19C66174D9E22D478845CF37B8DB2A43B4CD105CD71ADD2F217323797',
command: {
    find: 'pokemones',
    filter: {
        Type: 'Electric'
    },
    sort: {
        Speed: -1
    },
    '$db': 'Pokemon'
},
serverInfo: {
    host: 'DESKTOP-GCHMSATI',
    port: 27017,
    version: '8.0.8',
    gitVersion: '7f52660c14217ed2c8d3240f823a2291a4fe6abd'
},
```

```
_MONGOSH
    speed: -1
},
'$db': 'Pokemon'
},
serverInfo: {
    host: 'DESKTOP-GCHMSATI',
    port: 27017,
    version: '8.0.8',
    gitVersion: '7f52660c14217ed2c8d3240f823a2291a4fe6abd'
},
serverParameters: {
    internalQueryFaceBufferSizeBytes: 104857600,
    internalQueryFaceMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeRestricted',
    internalQueryPlannerIgnoreIndexWithCollationForRegex: 1
},
ok: 1
}
```