

## Comandos consultas clase MongoDB

Estimados estudiantes en este documento encontrarán los comandos de MongoDB que se cubren en clase para su facilidad en la ejecución, favor abstenerse de hacer cambios o borrar los comandos.

```
mongo "mongodb://cluster0-shard-00-00-jxeqq.mongodb.net:27017,  
Cluster0-shard-00-01-jxeqq.mongodb.net:27017,  
cluster0-shard-00-02-jxeqq.mongodb.net:27017/aggregations?replicaSet=Cluster0-shar  
d-0" --authenticationDatabase admin --ssl -u m121 -p aggregations --norc
```

Field Path: "\$fieldName" (" \$codeestudiante")

System variable: "\$\$UPPERCASE" (" \$\$CURRENT")

User variable: "\$mivariable"

```
db.solarSystem.aggregate([  
    $match:{meanTemperature:{$gte:-40, $lte:40}}  
    ])
```

```
db.solarSystem.aggregate([  
    $match:{type:{$ne:'Star'}}  
    ])
```

```
db.solarSystem.find({type:{$ne:'Star'}})
```

```
db.solarSystem.count()
```

```
db.solarSystem.aggregate([  
    $match:{type:{$ne:'Star'}}  
    ],  
    {$count:'Planetas'}  
    ])
```

```
db.solarSystem.find({name:'Earth'},{_id:0})
```

```
db.solarSystem.aggregate([  
    $match:{atmosphericComposition:{$in:['O2/']}}  
    ])
```

```

db.solarSystem.aggregate([
  {
    $match: {atmosphericComposition: {$in: [/O2/]}}
  },
  {
    $project: {_id: 0, name: 1, tieneLunas: {$gt: ['$numberOfMoons', 0]}}
  }
])

```

```

db.solarSystem.aggregate([
  {
    $match: {atmosphericComposition: {$in: [/O2/]}}
  },
  {
    $project: {_id: 0, name: 1, tieneLunas: {$gt: ['$numberOfMoons', 0]}}
  }
])

```

/\*Seleccionar las películas que cumplan con los siguientes criterios:

Imdb.rating menor a 7,

Que no incluya los géneros de horror ni crimen

clasificación (rated) "PG" o "G"

Lenguaje incluya Japonés e Inglés\*/

```

db.movies.aggregate([
  {
    $match: {
      'imdb.rating': {$lt: 7},
      genres: {$nin: ['Crimen', 'Horror']},
      rated: {$in: ['PG', 'G']},
      languages: {$all: ['Japanese', 'English']}
    }
  })

```

```

db.solarSystem.aggregate([
  {
    $project: {name: 1, gravity: 1}
  }
])

```

```

db.solarSystem.aggregate([
  {
    $project: {name: 1, 'gravity.value': 1}
  }
])

```

```

db.solarSystem.aggregate([
  {
    $project:{_id:0, name:1, 'gravedad': '$gravity.value'}
  })

```

Calcular nuestro peso para cada elemento del sistema solar

```

db.solarSystem.aggregate([
  {
    $project:{_id:0, name:1, 'Mipeso':{$multiply:[{$divide:['$gravity.value',
9.8]], 85}}}
  })

```

```

db.movies.aggregate([
  {
    $match:{title:{$type:'string'}}
  },
  {
    $project:{
      titulo:{$split:['$title',' ']},
      _id:0
    },
  },
  {
    $match:{
      titulo:{$size:1}
    }
  }
]).itcount()

```

```

db.movies.aggregate([
  {
    $match:{
      cast:{$elemMatch:{$exists:true}},
      directors:{$elemMatch:{$exists:true}},
      writers:{$elemMatch:{$exists:true}}
    }
  },
  {
    $project:{
      _id:0,

```

```

        cast:1,
        directors:1,
        writers:{
          $map:{
            input:'$writers',
            as:'escritor',
            in:{
              $arrayElemAt:[
                {
                  $split:['$$escritor',' (')],0]
                }
              ]
            }
          }
        },
        {
          $project:{

amorxlabor:{$gt:[{$size:{$setIntersection:["$cast","$directors","$writers"]}},0]
          }},
        {
          $match:{'amorxlabor':true}
        },
        {
          $count:'amorxlabor'
        }
      ]))

db.movies.aggregate([
  {
    $match:{
      cast:{$elemMatch:{$exists:true}},
      directors:{$elemMatch:{$exists:true}},
      writers:{$elemMatch:{$exists:true}}
    }
  },
  {
    $project:{
      _id:0,
      cast:1,
      directors:1,
      writers:{

```

```

        $map:{
          input:'$writers',
          as:'escritor',
          in:{
            $arrayElemAt:[
              ,
              $split:['$$escritor', ' (')],0]
            }
          }}
      }
    },
    {
      $project:{
        amorxlabor:{$setIntersection:['$cast','$directors','$writers']}
      },
      {
        $match:{
          amorxlabor:{$ne:[]}}
      }).itcount()
    }
  }
}

```

```
db.solarSystem.aggregate([{'$project':{'gravedad': '$gravity.value'}}]);
```

```

db.nycFacilities.aggregate([
  {
    $geoNear:{
      near:{
        type: "Point",
        coordinates: [-73.98769766092299, 40.757345233626594]
      },
      distanceField: "distanceFromMongoDB",
      spherical: true,
    }
  }
])

```

```

db.nycFacilities.aggregate([
  {
    $geoNear:{
      near:{
        type: "Point",
        coordinates: [-73.98769766092299, 40.757345233626594]
      }
    }
  }
])

```

```

    },
    distanceField: "distanceFromMongoDB",
    spherical: true,
    query: {type: 'Hospital'}
  }},
  {
    $limit: 5
  }
])

```

```

db.solarSystem.find({}, {_id: 0, name: 1, numberOfMoons: 1})
db.solarSystem.find({}, {_id: 0, name: 1, numberOfMoons: 1}).count()
db.solarSystem.find({}, {_id: 0, name: 1, numberOfMoons: 1}).skip(5)
db.solarSystem.find({}, {_id: 0, name: 1, numberOfMoons: 1}).limit(5)
db.solarSystem.find({}, {_id: 0, name: 1, numberOfMoons: 1}).sort({name: 1})

```

```

db.solarSystem.aggregate([
  $project: {
    _id: 0,
    name: 1,
    hasMagneticField: 1,
    numberOfMoons: 1
  },
  {
    $sort: {hasMagneticField: -1, numberOfMoons: -1}
  }
])

```

```

favoritos = [
  "Sandra Bullock",
  "Tom Hanks",
  "Julia Roberts",
  "Kevin Spacey",
  "George Clooney"]

```

```

Released en USA, Tomatoes.viewer.rating >=3

```

```

db.movies.aggregate([

```

```

    {
      $match:{
        'tomatoes.viewer.rating':{$gte:3},
        countries:'USA',
        cast:{
          $in:favoritos
        }
      },
    },
    {
      $project:{
        _id:0,
        title:1,
        'tomatoes.viewer.rating':1,
        cant_fav:{
          $size:{$setIntersection:['$cast',favoritos]}}
      }},
    {
      $sort:{cant_fav:-1,'tomatoes.viewer.rating':-1,title:-1}
    }
  ])

```

```

db.solarSystem.aggregate([
  $project:{
    _id: 0,
    name: 1,
    hasMagneticField:1,
    numberOfMoons: 1
  },
  {
    $sort:{hasMagneticField:-1, numberOfMoons:-1}
  }
])

```

```

db.movies.aggregate([
  {
    $group:{_id:'$year',
      num_peliculas:{$sum:1}}
  },

```

```

    {
      $sort:{num_peliculas:-1}
    }
  ])

```

```

db.movies.aggregate([
  {
    $group:{
      _id:{
        num_directores:{
          $cond:[{$isArray:'$directors'},{$size:'$directors'},0]
        }
      },
      num_peliculas:{$sum:1},
      promedio_metacritic:{$avg:'$metacritic'}
    }
  },
  {
    $sort:['_id.num_directores':-1]
  }
])

```

```

db.movies.aggregate([
  {
    $match:{metacritic:{$gte:0}},
  },
  {
    $group:{
      _id:null,
      promedio_metacritic:{$avg:'$metacritic'},
    }
  }
])

```

```

db.icecream_data.aggregate([
  {
    $project:{
      _id:0,
      max_temp:{
        $reduce:{
          input:'$trends',

```



```

        initialValue:-Infinity,
        in:{
$cond:[{$gt:['$$this.avg_high_tmp','$$value']},'$$this.avg_high_tmp','$$value']
        }
    }
}
    }
}
    }
    ]
)

```

```

db.icecream_data.aggregate([
    {
        $project:{
            _id:0,
            max_temp:{$max:'$trends.avg_high_tmp'}
        }
    }
])

```

```

db.icecream_data.aggregate([
    {
        $project:{
            _id:0,
            max_temp:{$max:'$trends.avg_high_tmp'},
            prom_cpi:{$avg:'$trends.icecream_cpi'},
            desviacion_std:{$stdDevPop:'$trends.icecream_cpi'}
        }
    }
])

```

```

db.movies.aggregate([
    {
        $match:{
            awards: /Won \d{1,2} Oscars?/
        }
    },
    {
        $group: {
            _id: null,

```

```

    rating_alto: { $max: "$imdb.rating" },
    rating_bajo: { $min: "$imdb.rating" },
    promedio_rating: { $avg: "$imdb.rating" },
    desviacion: { $stdDevSamp: "$imdb.rating" }
  }
}
])

```

```

db.movies.aggregate([
  {
    $match:{
      'imdb.rating':{$gt:0},
      year:{$gte:2010,$lte:2015},
      runtime:{$gte:90}
    }
  },
  {
    $unwind:'$genres'
  },
  {
    $group:{
      _id:{
        año:'$year',
        genero:'$genres'
      }
    }
  }
])

```

```

db.movies.aggregate([
  {
    $match:{
      'imdb.rating':{$gt:0},
      year:{$gte:2010,$lte:2015},
      runtime:{$gte:90}
    }
  },
  {
    $unwind:'$genres'
  },

```

```

{
  $group:{
    _id:{
      year:'$year',
      genre:'$genres'
    },
    promedio_rating:{$avg:'$imdb.rating'}
  }
},
{
  $sort:{
    '_id.year':-1,'promedio_rating':-1}
  },
  {
    $group:{
      _id:'$_id.year',
      genre:{$first:'$_id.genre'},
      promedio_rating:{$first:'$promedio_rating'}
    }
  }
})

```

```

db.air_alliances.aggregate([
  {
    $lookup:{
      from:'air_airlines',
      localField:'airlines',
      foreignField:'name',
      as:'aerolineas'
    }
  }
])

```

```

db.air_routes.aggregate([
  {
    $match: {
      airplane: /747|380/
    }
  },
  {

```

```
$lookup:{  
  from: "air_alliances",  
  localField:"airline.name",  
  foreignField: "airlines",  
  as: "alianzas"  
}}
```

```
])
```

Se trabajará con la colección companies.json

```
mongoimport --db startups --collection companies --drop --file  
C:/Users/apao/Downloads/companies.json
```

Mongoimport → bin

Enlace para versiones Mongodb 4.4 o +, para descargar el mongoimport primero y luego poder usar el comando

[https://www.mongodb.com/try/download/database-tools?tck=docs\\_databasetools](https://www.mongodb.com/try/download/database-tools?tck=docs_databasetools)

Sino se reconoce el comando desde el prompt del símbolo del sistema probar el siguiente comando

```
C:\Program Files\MongoDB\Server\4.2\bin>mongoimport -d startups -c companies  
--type json --drop --file "C:/...../companies.json"
```

```
db.companies.createIndex({description:"text",overview:"text"})  
db.companies.aggregate([  
  {  
    $match:{$text:{$search:"network"}}  
  }])
```

