

# Ejercicio

Crear una BD competicion

```
switched to db competicion
competicion>
```

Crear una colección llamada pruebas con los siguientes datos:

```
db.pruebas.insertMany([
  {nombre:"Carlos", mes:"Marzo", distKm:6, tiempoMin:42},
  {nombre:"Elena", mes:"Marzo", distKm:10, tiempoMin:60},
  {nombre:"Carlos", mes:"Marzo", distKm:2, tiempoMin:12},
  {nombre:"Elena", mes:"Marzo", distKm:10, tiempoMin:61},
  {nombre:"Carlos", mes:"Abril", distKm:5, tiempoMin:33},
  {nombre:"Elena", mes:"Abril", distKm:42, tiempoMin:285},
  {nombre:"Angel", mes:"Abril", distKm:5, tiempoMin:33}])
```

```
competicion> db.pruebas.insertMany([
  {nombre:"Carlos", mes:"Marzo", distKm:6, tiempoMin:42},
  {nombre:"Elena", mes:"Marzo", distKm:10, tiempoMin:60},
  {nombre:"Carlos", mes:"Marzo", distKm:2, tiempoMin:12},
  {nombre:"Elena", mes:"Marzo", distKm:10, tiempoMin:61},
  {nombre:"Carlos", mes:"Abril", distKm:5, tiempoMin:33},
  {nombre:"Elena", mes:"Abril", distKm:42, tiempoMin:285},
  {nombre:"Angel", mes:"Abril", distKm:5, tiempoMin:33}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("637288a7843c0dd002e3ac3c"),
    '1': ObjectId("637288a7843c0dd002e3ac3d"),
    '2': ObjectId("637288a7843c0dd002e3ac3e"),
    '3': ObjectId("637288a7843c0dd002e3ac3f"),
    '4': ObjectId("637288a7843c0dd002e3ac40"),
    '5': ObjectId("637288a7843c0dd002e3ac41"),
    '6': ObjectId("637288a7843c0dd002e3ac42")
  }
}
competicion>
```

Hacer las siguientes operaciones usando agregaciones:

- Calcular el número de pruebas que ha realizado cada corredor.

```
db.pruebas.aggregate([
  {$group:{"_id":"$nombre","cantidadPruebas":{$sum:1}}},
  {$sort:{"cantidadPruebas":1}}])
```

```
competicion> db.pruebas.aggregate([
  {$group:{"_id":"$nombre","cantidadPruebas":{$sum:1}}},
  {$sort:{"cantidadPruebas":1}}])
[
  { _id: 'Angel', cantidadPruebas: 1 },
  { _id: 'Carlos', cantidadPruebas: 3 },
  { _id: 'Elena', cantidadPruebas: 3 }
]
```

- Calcular el número de pruebas que ha realizado cada corredor por mes.

```
db.pruebas.aggregate([
  {$group:{"_id":{"nombre":"$nombre","mes":"$mes"},"cantidadPruebas":{$sum:1}}},
  {$sort:{"nombre":1,"mes":1}}])
```

```
competicion> db.pruebas.aggregate([
  {$group:{"_id":{"nombre":"$nombre","mes":"$mes"},"cantidadPruebas":{$sum:1}}},
  {$sort:{"nombre":1,"mes":1}}])
[
  { _id: { nombre: 'Carlos', mes: 'Abril' }, cantidadPruebas: 1 },
  { _id: { nombre: 'Elena', mes: 'Abril' }, cantidadPruebas: 1 },
  { _id: { nombre: 'Angel', mes: 'Abril' }, cantidadPruebas: 1 },
  { _id: { nombre: 'Elena', mes: 'Marzo' }, cantidadPruebas: 2 },
  { _id: { nombre: 'Carlos', mes: 'Marzo' }, cantidadPruebas: 2 }
]
competicion>
```

- Calcular el número total de Kms recorridos por cada corredor.

```
db.pruebas.aggregate([{$group:{"_id":"$nombre","cantidadKms":{"$sum":"$distKm"}},{ $sort:{"cantidadKms":1}}])
```

```
competicion> db.pruebas.aggregate([{$group:{"_id":"$nombre","cantidadKms":{"$sum":"$distKm"}},{ $sort:{"cantidadKms":1}}])
[
  { _id: 'Angel', cantidadKms: 5 },
  { _id: 'Carlos', cantidadKms: 13 },
  { _id: 'Elena', cantidadKms: 62 }
]
competicion>
```

- Calcular los kilómetros que corre cada uno de media al mes

```
db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre",mes:"$mes"},mediaKms:{$avg:'$distKm'}}]])
```

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre",mes:"$mes"},mediaKms:{$avg:'$distKm'}}]])
[
  { _id: { nombre: 'Carlos', mes: 'Abril' }, mediaKms: 5 },
  { _id: { nombre: 'Elena', mes: 'Abril' }, mediaKms: 42 },
  { _id: { nombre: 'Angel', mes: 'Abril' }, mediaKms: 5 },
  { _id: { nombre: 'Elena', mes: 'Marzo' }, mediaKms: 10 },
  { _id: { nombre: 'Carlos', mes: 'Marzo' }, mediaKms: 4 }
]
competicion>
```

- Visualizar las distintas distancias que ha corrido cada corredor en las pruebas.

```
db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre"},distancias:{$addToSet:'$distKm'}}]])
```

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre"},distancias:{$addToSet:'$distKm'}}]])
[
  { _id: { nombre: 'Elena' }, distancias: [ 10, 42 ] },
  { _id: { nombre: 'Carlos' }, distancias: [ 5, 6, 2 ] },
  { _id: { nombre: 'Angel' }, distancias: [ 5 ] }
]
competicion>
```

- Calcular el número medio de pruebas por corredor al mes (se cuenta el número de pruebas por persona y mes y a continuación se hace la media de este dato)  
Idea: - Sabemos contar el número de pruebas por mes. - Sabemos hacer la media de unos valores jusemos 2 etapas!

```
db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre",mes:"$mes"},pruebas:{$sum:1}}},{ $group:{_id:"$ _id.nombre",mediaPruebas:{$avg:"$pruebas"}}}])
```

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre",mes:"$mes"},... pruebas:{$sum:1}}},{ $group:{_id:"$ _id.nombre",mediaPruebas:{$avg:"$pruebas"}}}])
[
  { _id: 'Carlos', mediaPruebas: 1.5 },
  { _id: 'Angel', mediaPruebas: 1 },
  { _id: 'Elena', mediaPruebas: 1.5 }
]
competicion>
```

- Visualizar en cada mes qué distancias se han hecho en alguna sesión. (Si una distancia se ha corrido varias veces en ese mes debe aparecer varias veces).

```
db.pruebas.aggregate([{$group:{_id:{mes:"$mes"}, distancias:{$push:'$distKm'}}}])
```

```
competicion> db.pruebas.aggregate([{$group:{_id:{mes:"$mes"}, distancias:{$push:'$distKm'}}}])
[
  { _id: { mes: 'Abril' }, distancias: [ 5, 42, 5 ] },
  { _id: { mes: 'Marzo' }, distancias: [ 6, 10, 2, 10 ] }
]
competicion>
```

- Obtener la mayor y menor distancia recorrida por cada corredor.

```
db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre"}, mayorDistancia:{$max:'$distKm'}, menorDistancia:{$min:'$distKm'}}}])
```

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre"}, mayorDistancia:{$max:'$distKm'}, menorDistancia:{$min:'$distKm'}}}])
[
  { _id: { nombre: 'Carlos' }, mayorDistancia: 6, menorDistancia: 2 },
  { _id: { nombre: 'Angel' }, mayorDistancia: 5, menorDistancia: 5 },
  { _id: { nombre: 'Elena' }, mayorDistancia: 42, menorDistancia: 10 }
]
competicion>
```

- Queremos disponer de los datos de distancias recorridas en millas, sabiendo que una milla = 1,60934 km

```
db.pruebas.aggregate([{$project:{_id:0,'nombre':1,distMillas:{$multiply:['$distKm',1.60934]}}}])
```

```
competicion> db.pruebas.aggregate([{$project:{_id:0, 'nombre':1,distMillas:{$multiply:['$distKm',1.60934]}}}])
[
  { nombre: 'Carlos', distMillas: 9.65604 },
  { nombre: 'Elena', distMillas: 16.0934 },
  { nombre: 'Carlos', distMillas: 3.21868 },
  { nombre: 'Elena', distMillas: 16.0934 },
  { nombre: 'Carlos', distMillas: 8.0467 },
  { nombre: 'Elena', distMillas: 67.59228 },
  { nombre: 'Angel', distMillas: 8.0467 }
]
competicion>
```

- Obtener la media en kilómetros mensuales de cada corredor, pero solo para aquellos valores medios que sean superiores a 5km.

```
db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre", mes: "$mes"},media:{$avg:'$distKm'}}},{ $match: {media:{$gt:5}}}]
```

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre", mes: "$mes"},media:{$avg:'$distKm'}}},{ $match: {media:{$gt:5}}}]
[
  { _id: { nombre: 'Elena', mes: 'Abril' }, media: 42 },
  { _id: { nombre: 'Elena', mes: 'Marzo' }, media: 10 }
]
competicion>
```

- Calcular la media de kilómetros por corredor y mes, ordenado por mes.

```
db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre",mes:"$mes"},
mediaKms:{$avg:'$distKm'}}},{ $sort: {'_id.mes':1}}])
```

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre",mes:"$mes"},mediaKms:{$avg:'$distKm'}}},{
{$sort: {'_id.mes':1}}])
[
  { _id: { nombre: 'Carlos', mes: 'Abril' }, mediaKms: 5 },
  { _id: { nombre: 'Elena', mes: 'Abril' }, mediaKms: 42 },
  { _id: { nombre: 'Angel', mes: 'Abril' }, mediaKms: 5 },
  { _id: { nombre: 'Carlos', mes: 'Marzo' }, mediaKms: 4 },
  { _id: { nombre: 'Elena', mes: 'Marzo' }, mediaKms: 10 }
]
competicion>
```

- Mostrar el corredor que tiene mayor media absoluta.

```
db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre"},mediaKm: {$avg:'$distKm'}}},{ $sort:
{medimedia:-1}},{$limit:1}])
```

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre"},mediaKm: {$avg:'$distKm'}}},{
{$sort: {medimedia:-1}},{$limit:1}])
[ { _id: { nombre: 'Elena' }, mediaKm: 20.666666666666668 } ]
competicion>
```

- Calcula la media de pruebas realizadas por cada persona mensualmente. Idea: Redirige la salida de una agrupación creando una nueva colección. No muestra nada en la salida, pero puedes comprobarlo haciendo un find sobre la nueva colección de salida.

```
db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre", mes: "$mes"},
pruebas:{$sum:1}}], {$group:{_id:"$_id.nombre", mediaPruebas:{$avg:'$pruebas'}}},{ $out:
"pruebas_persona_mes"})
```

```
competicion> show collections
pruebas
pruebas_persona_mes
competicion> db.pruebas_persona_mes.find()
[
  { _id: 'Elena', mediaPruebas: 1.5 },
  { _id: 'Angel', mediaPruebas: 1 },
  { _id: 'Carlos', mediaPruebas: 1.5 }
]
competicion>
```

- Creamos una nueva colección en la BD llamada gustos, con los siguientes datos:  
 {nombre:"Carlos", aficiones:["siesta","cine"]},{nombre:"Elena", aficiones:["correr","cine"]}  
 {nombre:"Ana", aficiones:["viajar","cine"]},{nombre:"Angel", aficiones:["correr","montaña",  
 "cine"]}

```
db.gustos.insertMany([
  {nombre:"Carlos", aficiones:["siesta","cine"]},
  {nombre:"Elena", aficiones:["correr","cine"]},
  {nombre:"Ana", aficiones:["viajar","cine"]},
  {nombre:"Angel", aficiones:["correr","montaña","cine"]}
])
```

```
competicion> db.gustos.insertMany([
  {nombre:"Carlos", aficiones:["siesta","cine"]},
  {nombre:"Elena", aficiones:["correr","cine"]},
  {nombre:"Ana", aficiones:["viajar","cine"]},
  {nombre:"Angel", aficiones:["correr","montaña","cine"]}
])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6373a342bb6e8efdc962320d"),
    '1': ObjectId("6373a342bb6e8efdc962320e"),
    '2': ObjectId("6373a342bb6e8efdc962320f"),
    '3': ObjectId("6373a342bb6e8efdc9623210")
  }
}
competicion>
```

- Queremos saber el número de personas con el que cuenta cada afición. Idea: Cuando tenemos documentos que contienen un array y queremos agrupar por valores del array, a veces conviene eliminar los arrays y convertirlos en múltiples documentos. En realidad estamos "normalizando" (primera forma normal). Y agrupar sobre esos documentos.

```
db.gustos.aggregate([
  {$unwind:'$aficiones'},
  {$group:{_id:'$aficiones', numPersonas:{$sum:1}}}
])
```

```
competicion> db.gustos.aggregate([
  {$unwind:'$aficiones'},
  {$group:{_id:'$aficiones', numPersonas:{$sum:1}}}
])
[
  { _id: 'viajar', numPersonas: 1 },
  { _id: 'correr', numPersonas: 2 },
  { _id: 'cine', numPersonas: 4 },
  { _id: 'siesta', numPersonas: 1 },
  { _id: 'montaña', numPersonas: 1 }
]
competicion>
```