

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

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

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

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

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

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

4. Calcular los kilómetros que corre cada uno de media al mes.

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

5. Visualizar las distintas(DISTINCT) distancias que ha corrido cada corredor en las pruebas.

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

6. 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 ¡usemos 2 etapas!

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

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

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

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

```
competicion> db.pruebas.aggregate({$project:{nombre:"$nombre",distKm:"$distKm",distanciaMillas:{$divide:["$distKm",1.60934]}}})
[
  {
    _id: ObjectId("637286674c7cfc5cc6344510"),
    nombre: 'Carlos',
    distKm: 6,
    distanciaMillas: 3.7282364198988405
  },
  {
    _id: ObjectId("637286854c7cfc5cc6344511"),
    nombre: 'Elena',
    distKm: 10,
    distanciaMillas: 6.213727366498068
  },
  {
    _id: ObjectId("637286904c7cfc5cc6344512"),
    nombre: 'Carlos',
    distKm: 2,
    distanciaMillas: 1.2427454732996135
  },
  {
    _id: ObjectId("6372869d4c7cfc5cc6344513"),
    nombre: 'Elena',
    distKm: 10,
    distanciaMillas: 6.213727366498068
  },
  {
    _id: ObjectId("637286a94c7cfc5cc6344514"),
    nombre: 'Carlos',
    distKm: 5,
    distanciaMillas: 3.106863683249034
  },
  {
    _id: ObjectId("637286b34c7cfc5cc6344515"),
    nombre: 'Elena',
    distKm: 42,
    distanciaMillas: 26.097654939291886
  },
  {
    _id: ObjectId("637286bf4c7cfc5cc6344516"),
    nombre: 'Angel',
    distKm: 5,
    distanciaMillas: 3.106863683249034
  }
]
competicion>
```

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

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

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

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

11. Mostrar el corredor que tiene mayor media absoluta.

```
competicion> db.puebas.aggregate([{$group:{_id:{nombre:"$nombre"},distanciaCorrida:{avg:"$distKm"}},{$limit:1},{$sort:{ "_id.distanciaCorrida":-1}}]
[ { _id: { nombre: 'Elena', distanciaCorrida: 42 } } ]
competicion>
```

12. 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.

```
competicion> db.puebas.aggregate([{$group:{_id:{nombre:"$nombre",mes:"$mes"},vecesCorrenMes:{sum:1}}},{$group:{_id:"$ _id.nombre",mediaCorren:{avg:"$vecesCorrenMes"}]])
[
  { _id: 'Angel', mediaCorren: 1 },
  { _id: 'Carlos', mediaCorren: 1.5 },
  { _id: 'Elena', mediaCorren: 1.5 }
]
competicion> db.puebas.aggregate([{$group:{_id:{nombre:"$nombre",mes:"$mes"},vecesCorrenMes:{sum:1}}},{$group:{_id:"$ _id.nombre",mediaCorren:{avg:"$vecesCorrenMes"}},{$out:"CalcularMediaMes"}])
competicion> show collections
CalcularMediaMes
pruebas
competicion> db.CalcularMediaMes.find()
[
  { _id: 'Elena', mediaCorren: 1.5 },
  { _id: 'Angel', mediaCorren: 1 },
  { _id: 'Carlos', mediaCorren: 1.5 }
]
competicion>
```

13. Creamos una nueva colección en la BD llamada gustos, con los siguientes datos

```
competicion> db.gustos.find()
[
  {
    _id: ObjectId("6374ce807115465c54cd598d"),
    nombre: 'Carlos',
    aficiones: [ 'siesta', 'cine' ]
  },
  {
    _id: ObjectId("6374ce8d7115465c54cd598e"),
    nombre: 'Elena',
    aficiones: [ 'correr', 'cine' ]
  },
  {
    _id: ObjectId("6374ce997115465c54cd598f"),
    nombre: 'Ana',
    aficiones: [ 'viajar', 'cine' ]
  },
  {
    _id: ObjectId("6374cea67115465c54cd5990"),
    nombre: 'Angel',
    aficiones: [ 'correr', 'montaña', 'cine' ]
  }
]
competicion>
```

14. 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.

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