

Calcular el número de pruebas que ha realizado cada corredor.

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

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

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

Calcular el número total de Kms recorridos por cada corredor.

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

Calcular los kilómetros que corre cada uno de media al mes

```
competicion> db.pruebas.aggregate([{$group:{$_id:{nombre:"$nombre",mes:"$mes"},KmsTotales:{$sum:"$distKm"}}}])
[
  { _id: { nombre: 'Carlos', mes: 'Marzo' }, KmsTotales: 8 },
  { _id: { nombre: 'Elena', mes: 'Marzo' }, KmsTotales: 20 },
  { _id: { nombre: 'Elena', mes: 'Abril' }, KmsTotales: 42 },
  { _id: { nombre: 'Angel', mes: 'Abril' }, KmsTotales: 5 },
  { _id: { nombre: 'Carlos', mes: 'Abril' }, KmsTotales: 5 }
]
competicion>
```

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

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

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

Obtener la mayor y menor distancia recorrida por cada corredor.

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

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

```
competicion> db.pruebas.aggregate([{$project: {_id: "$nombre", distKm: "$distKm", distMillas: {$divide: ["$distKm", 1.60934]}}}])
[
  { _id: 'Carlos', distKm: 6, distMillas: 3.7282364198988405 },
  { _id: 'Elena', distKm: 10, distMillas: 6.213727366498068 },
  { _id: 'Carlos', distKm: 2, distMillas: 1.2427454732996135 },
  { _id: 'Elena', distKm: 10, distMillas: 6.213727366498068 },
  { _id: 'Carlos', distKm: 5, distMillas: 3.106863683249034 },
  { _id: 'Elena', distKm: 42, distMillas: 26.097654939291886 },
  { _id: 'Angel', distKm: 5, distMillas: 3.106863683249034 }
]
```

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

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

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

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

**Mostrar el corredor que tiene mayor media absoluta.**

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

```
competicion> db.pruebas.aggregate([{$group:{_id:{nombre:"$nombre",Mes:"$mes"},mediaCarreras:{$sum:1}},{$group:{_id:"$_id.nombre",promedioCarreras:{$avg:"$mediaCarreras"}},{$out:"media_persona_mes"}])
competicion> db.media_persona_mes.find()
[
  { _id: 'Carlos', promedioCarreras: 1.5 },
  { _id: 'Elena', promedioCarreras: 1.5 },
  { _id: 'Angel', promedioCarreras: 1 }
]
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"]}**

```
competicion> db.gustos.find()
[
  {
    _id: ObjectId("6373b9ebc815a07cddfc6dc4"),
    nombre: 'Carlos',
    aficiones: [ 'siesta', 'cine' ]
  },
  {
    _id: ObjectId("6373ba01c815a07cddfc6dc5"),
    nombre: 'Elena',
    aficiones: [ 'correr', 'cine' ]
  },
  {
    _id: ObjectId("6373ba0bc815a07cddfc6dc6"),
    nombre: 'Ana',
    aficiones: [ 'viajar', 'cine' ]
  },
  {
    _id: ObjectId("6373ba15c815a07cddfc6dc7"),
    nombre: 'Angel',
    aficiones: [ 'correr', 'montaña', 'cine' ]
  }
]
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

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