

# Nociones Teóricas Y Técnicas

(Proyecto 1004\_VACUNA)

Antonio Ávila Delgado  
1º ASIR

## **DESCRIPCIÓN DEL PROYECTO**

En este proyecto tiene una gran conexión con el mundo real en el que estamos afectados por la pandemia de la COVID-19. Una de las grandes necesidades es encontrar una vacuna lo antes posible, y en ello MongoDB juega un importante papel por su capacidad de manejar gran cantidad de datos. En concreto, vamos a centrarnos en uno de los aspectos más importante de la vacuna: los ensayos clínicos con humanos, que como sabemos, para que tengan éxito, tienen que participar un número altísimo de voluntarios, generando gran cantidad de datos. De ahí la importancia de Mongo DB.

Para conseguir todo ello, vamos a poblar con documentos una colección llamada VACUNA con información sobre los miembros de los ensayos clínicos, mezclando campos de distintos tipos (cadenas de caracteres, números [enteros y decimales], booleanos, fechas, entre otros. En concreto tendremos:

- Nombre
- Apellido
- Género
- Fecha de nacimiento
- Localidad
- Raza
- Enfermedades previas
- Tratamientos preexistentes
- Dosis de los tratamientos preexistentes (expresadas en mg)
- Tipo de ensayo
- Uso de placebo
- Pertenencia a un estudio clínico anterior

Después haremos distintas operaciones de búsqueda/consultas sobre dichos datos.

## NOCIONES TEORICO-TÉCNICAS

En primer lugar tenemos que resolver un desafío. Durante el ensayo clínico de la vacuna uno de los miembros del ensayo clínico desarrolló complicaciones. Es una persona nacida en 1955, por lo que tenemos que buscar todas las otras personas nacidas en dicho año, para ver si tienen un patrón en común que explique esas complicaciones médicas. Para ello usaremos **\$regex** y el **^** de esta manera:

```
db.vacuna.find({ fecha_nacimiento: { $regex: /^1955/ } }).pretty()
```

Se encuentran varias personas con síntomas similares, por lo que hay que extender la búsqueda a todas las personas de esa década, es decir, los nacidos en los años 50. Por ahora, al ser muchos, solo necesitaremos sus nombres y apellidos (por lo que usaremos una “projection”). Para ello usaremos **\$regex** y el **.** y la “projection” de esta manera:

```
db.vacuna.find({ fecha_nacimiento: { $regex: /195./ }, { _id: 0, nombre: 1, apellido: 1 } }).pretty()
```

Parece ser que el grupo de personas mayores está teniendo más desafíos con la vacuna, así que vamos a extender la búsqueda para saber el número de miembros del ensayo mayores de 20 años (es decir, nacidos el siglo pasado). Usaremos **\$regex**, dos **.** y **count**

```
db.vacuna.find({ fecha_nacimiento: { $regex: /19../ } }).count()
```

Hemos notado, gracias a las búsquedas, que en las personas mayores que tienen factores de riesgo con la tensión arterial están más afectados, así que vamos a buscar todos los miembros del ensayo clínico que tienen problemas de hipertensión (da igual que se escriba con mayúscula o minúscula).

Usaremos **\$regex** y **\$options** (en concreto **i**).

```
db.vacuna.find({ enfermedades_previas: { $regex: /hipertensión/, $options: 'i' } }).pretty()
```

En paralelo, tenemos que resolver el problema de los hipocondriacos (los que se creen enfermos, pero no lo están), ya que pueden sesgar los resultados del ensayo clínico. Por tanto necesitamos el número de miembros del ensayo clínico que son hipocondriacos (da igual que se escriba con mayúscula o minúscula). Usaremos **\$regex** y **count**

```
db.vacuna.find({ enfermedades_previas: { $regex: /hipocondriaco/ i } }).count()
```

Otro factor a tener en cuenta para el éxito de la vacuna es que no haya personas que sea alérgica a ella. Por tanto, necesitamos los datos de las personas que tienen alergias, ya que son potenciales candidatos a desarrollar una alergia a la vacuna y hay que hacerles un seguimiento especial. Como no sabemos cómo los médicos introdujeron los datos, vamos a buscar todas las variantes posibles usando **/alerg\*/i**

```
db.vacuna.find({ enfermedades_previas: { $regex: /alerg*/i } }).pretty()
```

También es muy interesante saber las personas que nunca han estado enfermas en su vida, pero en este caso solo necesitamos nombre, apellidos y fecha de nacimiento (“projection”)

Usaremos **\$regex** y **\$options** (en concreto **mix**) y también **.** y **\***

```
db.vacuna.find({ enfermedades_previas: { $regex: /nunc. *enferm./, $options: 'mix' } }, { _id: false, nombre: 1, apellido: 1, fecha_nacimiento: 1, enfermedades_previas: 1 }).pretty()
```

En cualquier caso, en búsquedas anteriores nos centrábamos en personas con problemas con tensión arterial, pero no solo eran importantes los que tenían hipertensión, sino también otros trastornos como la hipotensión. Por ello vamos a buscar a todos los que tengan problemas con la tensión, ya sea hipertensión o hipotensión y en cualquier línea de la descripción.

Usaremos **\$regex** y **\$options** (en concreto **m**) y no hará falta **.** y **\***

```
db.vacuna.find({ enfermedades_previas: { $regex: /tensión/, $options: 'm' } }).pretty()
```

Y los niños, no nos olvidemos de los niños. Y a efectos de este ensayo clínico, se considera niño a todos los que tengan 10 años o menos. Por tanto, necesitamos a todos los miembros del ensayo clínico que son niños y niñas, solo nombre, apellidos y género (con “projection”)

```
db.vacuna.find({ fecha_nacimiento: { $regex: /201./ } }, { _id: 0, nombre: 1, apellido: 1, sexo: 1 }).pretty()
```

Pero ahora viene la pregunta importante: ¿quiénes están siendo vacunados realmente?

Para ello usamos esta consulta nos devuelva quienes reciben la vacuna realmente, usando **\$eq**

```
db.vacuna.find({ placebo: { $eq: "N" } }).pretty()
```

Otro factor importante es hacer un seguimiento a las medicaciones preexistentes que estuvieran tomando los miembros del ensayo clínico, por si pudieran afectar negativamente a la potencia de la vacuna. Nos interesan especialmente las dosis altas. Por tanto vamos a hacer una consulta que nos devuelva todas las medicaciones cuyas dosis fuera mayor o igual que 100 mg, usando **\$gte**

```
db.vacuna.find({tratamiento_preexistente_dosis_mg: { $gte: 100 }}).pretty()
```

En los archivos \*.js también se incluyen búsquedas con \$gt, \$lte, \$lt

También surge la duda sobre si los participante ya son veteranos haciendo ensayos clínicos. Por eso vamos a hacer la consulta que nos devuelva el número de personas que no han participado antes en un ensayo, usando **\$ne**

```
db.vacuna.find({miembro_ensayos_anteriores: { $ne: "S" }}).count()
```

También queremos analizar los polos opuestos de las dosis de medicamentos, por ello vamos a realizar la consulta que nos devuelva quienes están tomando o bien las dosis más altas (200 mg) y o bien las más bajas (1.25 mg) de cualquier medicamento, usando **\$in**

```
db.vacuna.find({tratamiento_preexistente_dosis_mg: { $in: [ 200, 1.25 ] }}).pretty()
```

Y ahora queremos saber quienes no están tomando ni las dosis más altas (200 mg), ni las más bajas (1.25 mg) de cualquier medicamento, usando **\$nin**

```
db.vacuna.find({tratamiento_preexistente_dosis_mg: { $nin: [ 200, 1.25 ] }}).pretty()
```

Ha saltado la alarma, hay otro caso de complicaciones de salud con la vacuna para los pacientes que tomaban el medicamento Excedrin en dosis superiores a 50 mg. Para ello vamos a hacer la siguiente consulta usando **\$and**

```
db.vacuna.find({$and: [{tratamiento_preexistente_dosis_mg: { $gte: 50 }}, {tratamiento_preexistente: { $eq: "Excedrin" } }]}).pretty()
```

Por seguridad vamos a hacer la consulta nos devuelva los datos de los miembros el ensayo que toman el medicamento Excedrin (da igual la dosis) y cualquier otro medicamento en dosis superiores a 200 mg, usando **\$or**

```
db.vacuna.find({$or: [{ tratamiento_preexistente_dosis_mg: {$gte: 200}}, { tratamiento_preexistente: {$eq: "Excedrin" } } ]}).pretty()
```

También tenemos que investigar, en el ensayo clónico, si el género o el país de origen influye en la eficacia de la vacuna. Se ha comprobado que los varones de Uganda no suelen tener ningún problema con la vacuna, así que vamos a saber quiénes son, usando **\$nor**

```
db.vacuna.find({$nor: [{ localidad: {$eq: Uganda}}, { sexo: {$eq: "M" } } ]}).pretty()
```

Y para finalizar, los medicamentos (da igual la marca o formulación) han demostrado en estudios anteriores que, al ser dosis bajas, no afectan a la eficacia de la vacuna, así que vamos a hacer una lista de los mismos, para saber cuáles son, usando **\$not**

```
db.vacuna.find( { tratamiento_preexistente_dosis_mg: {$not: { $gte: 4 } } } ).pretty()
```

Con todos estos análisis de datos, gracias a MongoDB, estamos seguros que los ensayos clínicos serán un éxito, lo cual allanará el camino hacia el lanzamiento, a nivel mundial, de la vacuna de la COVID-19.