

# Nociones Teóricas Y Técnicas

(Proyecto 1008\_INMUNIDAD)

Antonio Ávila Delgado  
1º ASIR

## DESCRIPCIÓN

Este proyecto tiene una gran conexión con el mundo real en el que estamos afectados por la pandemia de la COVID-19. En este contexto, conseguir la inmunidad (a través de vacunas) es primordial. Dentro del proyecto, nos encontramos con las siguientes **premisas**:

- la inmunidad total se consigue con 4 dosis.
- a causa de problemas de producción farmacéutica, el gobierno ha decidido que las dosis pueden proceder de cualquier fabricante, hasta llegar a 4 por paciente.
- el presupuesto público ya no puede costear la vacunación gratuita, por tanto el importe se traslada a los pacientes.
- el gobierno ha prometido que el paciente pagará un precio barato por la inmunidad total, siendo este menos de 100 euros.
- debido a la complejidad de la situación, se nos pide trabajar en este proyecto con más de 2 colecciones, con datos de fabricante, con datos de pacientes, con historiales médicos, etc. En cualquier caso, todos los pacientes ya han recibido las 4 dosis.
- nos encontramos ante una compleja relación N:M (una vacuna puede ir a varios pacientes, un paciente puede recibir varias vacunas de distintos fabricantes).
- en las múltiples colecciones, se mezclan campos de distintos tipos (cadenas de caracteres, números [enteros y decimales], booleanos, fechas, entre otros. Mostramos algunos ejemplos:

La colección VACUNA contiene este tipo de documentos:

```
{ cod_vacuna : "vac001",
  nombre_vacuna: "Pfizer",
  precio_dosis: 22.50,
  nacionalidad: "EEUU"}
```

La colección PACIENTE contiene este tipo de documentos:

```
{ cod_paciente : "patient001",
  fecha_nacimiento: ISODate("1992-02-02T08:00:00Z"),
  nombre_paciente: "Francisco Romero",
  lugar_nacimiento: "Madrid",
  VIP: true}
```

La colección HISTORIAL contiene este tipo de documentos:

```
{
  cod_paciente : "patient001",
  enfermedades_previas: "Hipocondriaco",
  tratamiento_preexistente: ["Xylishield", "Lipitor", "Zocor"],
  tratamiento_dosis_mg: 3.75,
  sangre: { tipo: "A", factor_RH:"positivo"},
  detalle_vacuna:
    [
      {cod_vacuna : "vac001", "dosis_recibidas":2},
      {cod_vacuna : "vac008", "dosis_recibidas":1},
      {cod_vacuna : "vac009", "dosis_recibidas":1}
    ]
}
```

Dentro del proyecto, tenemos el siguiente **objetivo**: comprobar que el gobierno está cumpliendo su promesa de precios. Es algo complejo en naturaleza por sus múltiples vertientes, que son:

- sacar datos de las múltiples colecciones para averiguar el precio por dosis de las vacunas, por separado (según fabricante), por paciente.
- solo tendremos en cuenta a los pacientes con grupo sanguíneo "A" (que según los estudios, son los más expuestos al virus).
- eliminar a los que han pagado menos de 100 euros (ya que pagaron el precio barato prometido por el gobierno).
- y de los que han pagado más de 100, determinar quiénes han pagado caro, y dentro de ellos, quiénes MUY caro (más de 30% de sobreprecio).
- Finalmente, es necesario que el resultado salga ordenado de forma descendente, mostrando código de paciente, nombre, precio pagado y cómo de caro ha sido.

En primer lugar usamos el operador **\$lookup** para intentar unir 2 colecciones de esta manera:

```
{
  $lookup: {
    from: "vacuna",
    localField: "detalle_vacuna.cod_vacuna",
    foreignField: "cod_vacuna",
    as: "vacuna_recibida"
  }
}
```

```
{
  "cod_paciente" : "pacient001",
  "detalle_vacuna" : [
    {
      "cod_vacuna" : "vac001",
      "dosis_recibidas" : 2
    },
    {
      "cod_vacuna" : "vac008",
      "dosis_recibidas" : 1
    },
    {
      "cod_vacuna" : "vac009",
      "dosis_recibidas" : 1
    }
  ],
  "vacuna_recibida" : [
    {
      "nombre_vacuna" : "Pfizer",
      "precio_dosis" : 22.5,
      "nacionalidad" : "EEUU"
    },
    {
      "nombre_vacuna" : "Bayer",
      "precio_dosis" : 48.5,
      "nacionalidad" : "Alemana"
    },
    {
      "nombre_vacuna" : "Sinopharm",
      "precio_dosis" : 50.5,
      "nacionalidad" : "China"
    }
  ]
}
```

Una vez conseguido, buscamos el objetivo más ambicioso de unir más de 2 colecciones:

```
{ $lookup: {
  from: "vacuna",
  localField: "detalle_vacuna.cod_vacuna",
  foreignField: "cod_vacuna",
  as: "vacuna_recibida"
}
},
{ $lookup: {
  from: "paciente",
  localField: "cod_paciente",
  foreignField: "cod_paciente",
  as: "paciente_vacunado"
} }
}
```

```
{
  "cod_paciente" : "patient001",
  "detalle_vacuna" : [
    {
      "cod_vacuna" : "vac001",
      "dosis_recibidas" : 2
    },
    {
      "cod_vacuna" : "vac008",
      "dosis_recibidas" : 1
    },
    {
      "cod_vacuna" : "vac009",
      "dosis_recibidas" : 1
    }
  ],
  "vacuna_recibida" : [
    {
      "nombre_vacuna" : "Pfizer",
      "precio_dosis" : 22.5,
      "nacionalidad" : "EEUU"
    },
    {
      "nombre_vacuna" : "Bayer",
      "precio_dosis" : 48.5,
      "nacionalidad" : "Alemana"
    },
    {
      "nombre_vacuna" : "Sinopharm",
      "precio_dosis" : 50.5,
      "nacionalidad" : "China"
    }
  ],
  "paciente_vacunado" : [
    {
      "fecha_nacimiento" : ISODate("1992-02-02T08:00:00Z"),
      "nombre_paciente" : "Francisco Romero",
      "lugar_nacimiento" : "Madrid"
    }
  ]
}
```

Después, utilizamos el operador de etapa **\$unwind** para recibir la información de dosis por vacunas-paciente, ya que una de las colecciones tiene un conjunto (array):

```
{
  $unwind: "$detalle_vacuna"
}
```

, y después **\$group**, para agrupar todos los resultado y finalmente el operador **\$sum** para hacer suma agregada:

```
{
  $group: {
    _id: "$paciente_vacunado.nombre_paciente",
    "total_dosis_recibidas": { $sum: "$detalle_vacuna.dosis_recibidas" }
  }
}
```

y así nos aseguramos que es cierto lo que nos indicaron cuando nos entregaron todas las colecciones, a saber, que han recibido 4 dosis todos los pacientes:

```
{ "_id" : [ "Manuel Escudero" ], "total_dosis_recibidas" : 4 }
{ "_id" : [ "Rodrigo Bernabeu" ], "total_dosis_recibidas" : 4 }
{ "_id" : [ "Laura Vaz" ], "total_dosis_recibidas" : 4 }
{ "_id" : [ "Francisco Romero" ], "total_dosis_recibidas" : 4 }
{ "_id" : [ "Felipe Ramos" ], "total_dosis_recibidas" : 4 }
{ "_id" : [ "Irina Putin" ], "total_dosis_recibidas" : 4 }
{ "_id" : [ "Soledad Sanz" ], "total_dosis_recibidas" : 4 }
{ "_id" : [ "Sonia Lara" ], "total_dosis_recibidas" : 4 }
{ "_id" : [ "Martina Klein" ], "total_dosis_recibidas" : 4 }
```

Con ello, estamos ya en disposición de utilizar el operador **\$set**, para crear un campo nuevo llamado “precio\_cada\_dosis”

```
{
  $set: {
    precio_cada_dosis: {$arrayElemAt: ["$vacuna_recibida",0]}
  }
}
```

que después nos hará falta para hacer cálculos sobre los precios

```
"coste_por_vacuna": {$multiply:
  ["$detalle_vacuna.dosis_recibidas", "$precio_cada_dosis.precio_dosis"]}
```

Como se nos pide que los resultados aparezcan en un cierto orden, necesitamos el operador **\$sort**, que toma un parámetro negativo para conseguir un orden descendente.

```
{ $sort: { "total_vacunas": -1 } } ]
```

Al pedírsenos que solo consideremos, grupo sanguíneo “A” y precios superiores a 100, usamos el operador **\$match** en los lugares correspondientes.

```
{ $match : { "sangre.tipo" : { $eq: "A" } } },
{ $match: { "total_vacunas": { $gt: 100 } } }
```

Y como al final se nos pide definir una condición, para saber quiénes han pagado caro y muy caro, usamos el operador **\$cond**

```
{
  $cond: { if: { $gte: [ "$total_vacunas", 130 ] }, then: "muy caro", else: "caro" }
}
```

Todo ello, usando adecuadamente el operador **\$project**, **\$eq** and **\$gt** en los momentos adecuados.

Teniendo en cuenta todos esos operadores, construimos esta consulta que cumple con los criterios del objetivo marcado:

**db.historial.aggregate(**

```
[
{ $match : { "sangre.tipo" : { $eq: "A" } } },
{
  $unwind: "$detalle_vacuna"
},
{
  $lookup: {
    from: "vacuna",
    localField: "detalle_vacuna.cod_vacuna",
    foreignField: "cod_vacuna",
    as: "vacuna_recibida"
  }
},
{
  $lookup: {
    from: "paciente",
    localField: "cod_paciente",
    foreignField: "cod_paciente",
    as: "paciente_vacunado"
  }
},
{
```



```

    $set: {
      precio_cada_dosis: {$arrayElemAt: ["$vacuna_recibida",0]}
    }
  },

  {
    $project: {
      _id: 0,
      "cod_paciente": 1,
      "detalle_vacuna.cod_vacuna": 1,
      "detalle_vacuna.dosis_recibidas": 1,
      "vacuna_recibida.precio_dosis": 1,
      "paciente_vacunado.nombre_paciente": 1,
      "coste_por_vacuna": {$multiply:
        ["$detalle_vacuna.dosis_recibidas", "$precio_cada_dosis.precio_dosis"]}
    }
  },
{
  $group: {
    _id: {
      ref: "$cod_paciente",
      nombre: "$paciente_vacunado.nombre_paciente",
    },

```

```

    "total_vacunas": { $sum: "$coste_por_vacuna" }
  }
},
{
  $sort: {
    "total_vacunas": -1
  }
},
{
  $match: {
    "total_vacunas": { $gt: 100 }
  }
},
{
  $project:
  {
    "persona inmunizada": "$_id",
    "_id": 0,

    "coste total vacunas": "$total_vacunas",
    "precio pagado":
    {
      $cond: { if: { $gte: [ "$total_vacunas", 130 ] }, then: "muy caro", else: "caro" }
    }
  }
}

```

```

    }
  }
}

]
).pretty()

```

Y el resultado es el que se nos ha pedido:

```

{
  "persona inmunizada" : {
    "ref" : "pacient001",
    "nombre" : [
      "Francisco Romero"
    ]
  },
  "coste total vacunas" : 144,
  "precio pagado" : "muy caro"
}
{
  "persona inmunizada" : {
    "ref" : "pacient003",
    "nombre" : [
      "Laura Vaz"
    ]
  },
  "coste total vacunas" : 138.75,
  "precio pagado" : "muy caro"
}
{
  "persona inmunizada" : {
    "ref" : "pacient006",
    "nombre" : [
      "Rodrigo Bernabeu"
    ]
  },
  "coste total vacunas" : 137.25,
  "precio pagado" : "muy caro"
}
{
  "persona inmunizada" : {
    "ref" : "pacient009",
    "nombre" : [
      "Irina Putin"
    ]
  },
  "coste total vacunas" : 108,
  "precio pagado" : "caro"
}

```