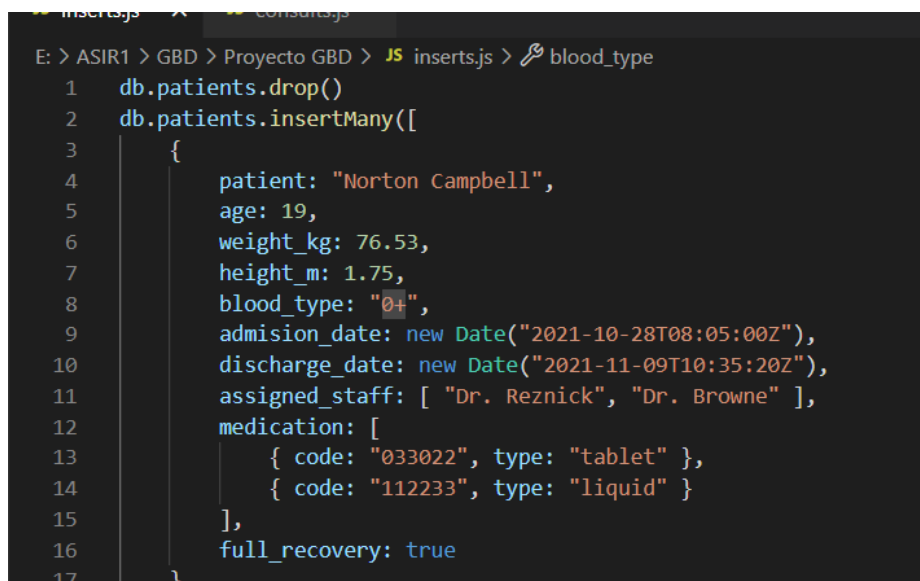


## PROYECTO FINAL DE TRIMESTRE

### MARÍA DEL CARMEN PAREJA FERREIRA

En este proyecto, se presenta una base de datos creada a partir de la información de pacientes y ex-pacientes de un hospital ficticio.

La base de datos, redactada en inglés, cuenta con 20 registros, dentro de los cuales encontrará campos de diversos tipos con información sobre: nombre, características físicas (peso y altura), grupo sanguíneo, fecha de admisión, fecha de alta (si procede), médico asignado, medicación (código del medicamento y tipo) y estado de recuperación.



```

E: > ASIR1 > GBD > Proyecto GBD > JS inserts.js > blood_type
1  db.patients.drop()
2  db.patients.insertMany([
3    {
4      patient: "Norton Campbell",
5      age: 19,
6      weight_kg: 76.53,
7      height_m: 1.75,
8      blood_type: "0+",
9      admission_date: new Date("2021-10-28T08:05:00Z"),
10     discharge_date: new Date("2021-11-09T10:35:20Z"),
11     assigned_staff: [ "Dr. Reznick", "Dr. Browne" ],
12     medication: [
13       { code: "033022", type: "tablet" },
14       { code: "112233", type: "liquid" }
15     ],
16     full_recovery: true
17   },

```

Por su parte, para las consultas se ha hecho un uso total de todos los operadores vistos en clase;

- |         |         |               |
|---------|---------|---------------|
| ✓ \$eq  | ✓ \$ne  | ✓ \$not       |
| ✓ \$gt  | ✓ \$nin | ✓ \$regex     |
| ✓ \$gte | ✓ \$and | ✓ \$all       |
| ✓ \$lt  | ✓ \$or  | ✓ \$ElemMatch |
| ✓ \$lte | ✓ \$nor | ✓ \$exists    |

además de los nuevos operadores que dejaré brevemente explicados:

- **\$type**: obtendrá los documentos cuyo campo tenga un valor del tipo especificado (string, array, null...).

```
{ field: { $type: <BSON type> } }
```

- **\$project:** el operador pertenece al método *aggregation* y permite especificar la inclusión de campos, la supresión del campo `_id`, la adición de nuevos campos y el restablecimiento de los valores de los campos existentes.

```
{ $project: { <specification(s)> } }
```

- **\$subtract:** el operador pertenece al método *aggregation* y permite restar dos números, dos fechas para devolver la diferencia en milisegundos, o una fecha y un número en milisegundos para devolver la fecha resultante...

```
{ $subtract: [ <expression1>, <expression2> ] }
```

- **\$divide:** el operador pertenece al método *aggregation* y permite dividir un número por otro, devolviendo posteriormente el resultado.

```
{ $divide: [ <expression1>, <expression2> ] }
```

- **\$multiply:** el operador pertenece al método *aggregation* y permite multiplicar un número por otro, devolviendo posteriormente el resultado.

```
{ $multiply: [ <expression1>, <expression2> ] }
```

- **\$round:** el operador pertenece al método *aggregation* y permite redondear un número a un entero al decimal especificado.

```
{ $round: [ <number>, <place> ] }
```