

C3 - Bases de dades SQL (MySQL)

TA17 - MongoDB

Ejercicio 1 - TA17-PT1 (Creación e inserciones):

```
//crear o mover a la base de datos  
use TA17-PT1
```

```
//creamos las colecciones  
db.createCollection("FABRICANTES");  
db.createCollection("ARTICULOS");
```

```
//provamos de crear los documentos de fabricantes y llenamos de datos  
try {
```

```
    db.FABRICANTES.insertMany( [  
        {nombre: "Sony"},  
        {nombre: "Xiaomi"},  
        {nombre: "Samsung"},  
        {nombre: "Toshiba"},  
        {nombre: "LG"}  
    ]);
```

```
} catch (e) {  
    print (e);  
}
```

```
//provamos de crear los documentos de articulos, llenamos de datos y referenciamos la clave foranea  
try {
```

```
    db.ARTICULOS.insertMany( [  
        {nombre: "Sony Xperia Z", precio: 120, fabricante: ObjectId("6536a1ee5a29557e5a976b5a")},  
        {nombre: "Sony Lambda", precio: 141, fabricante: ObjectId("6536a1ee5a29557e5a976b5a")},  
        {nombre: "Mi 3 plus", precio: 187, fabricante: ObjectId("6536a1ee5a29557e5a976b5b")},  
        {nombre: "Mi 4 Compact", precio: 622, fabricante: ObjectId("6536a1ee5a29557e5a976b5b")},  
        {nombre: "LG Plus Note", precio: 430, fabricante: ObjectId("6536a1ee5a29557e5a976b5e")},  
        {nombre: "LG Mariloli", precio: 188, fabricante: ObjectId("6536a1ee5a29557e5a976b5e")}  
    ]);
```

```
} catch (e) {  
    print (e);  
}
```

Ejercicio 1 - TA17-PT1 (Consultas):

//hacemos busqueda de los articulos con precio mayor o igual a 200

```
db.ARTICULOS.find({precio:{$gte:200}});
```

//Captura:

```
>_MONGOSH
> db.ARTICULOS.find({precio:{$gte:200}});
< {
  _id: ObjectId("6536a4ab5a29557e5a976b62"),
  nombre: 'Mi 4 Compact',
  precio: 622,
  fabricante: ObjectId("6536a1ee5a29557e5a976b5b")
}
{
  _id: ObjectId("6536a4ab5a29557e5a976b63"),
  nombre: 'LG Plus Note',
  precio: 430,
  fabricante: ObjectId("6536a1ee5a29557e5a976b5e")
}
TA17-PT1> |
```

//hacemos busqueda de articulos ordenados por precios descendentes

```
db.ARTICULOS.find().sort({precio:-1});
```

//Captura:

```
>_MONGOSH
> db.ARTICULOS.find().sort({precio:-1});
< {
  _id: ObjectId("6536a4ab5a29557e5a976b62"),
  nombre: 'Mi 4 Compact',
  precio: 622,
  fabricante: ObjectId("6536a1ee5a29557e5a976b5b")
}
{
  _id: ObjectId("6536a4ab5a29557e5a976b63"),
  nombre: 'LG Plus Note',
  precio: 430,
  fabricante: ObjectId("6536a1ee5a29557e5a976b5e")
}
{
  _id: ObjectId("6536a4ab5a29557e5a976b64"),
  nombre: 'LG Mariloli',
  precio: 188,
  fabricante: ObjectId("6536a1ee5a29557e5a976b5e")
}
{
  _id: ObjectId("6536a4ab5a29557e5a976b61"),
  nombre: 'Mi 3 plus',
  precio: 187,
  fabricante: ObjectId("6536a1ee5a29557e5a976b5b")
}
{
  _id: ObjectId("6536a4ab5a29557e5a976b60"),
  nombre: 'Sony Lambda',
  precio: 141,
  fabricante: ObjectId("6536a1ee5a29557e5a976b5a")
}
{
  _id: ObjectId("6536a4ab5a29557e5a976b5f"),
  nombre: 'Sony Xperia Z',
  precio: 120,
  fabricante: ObjectId("6536a1ee5a29557e5a976b5a")
}
TA17-PT1>
```

Ejercicio 2 - TA17-PT2 (Creación e inserciones):

```
//crear o mover a la base de datos
use TA17-PT2
```

```
//creamos las colecciones
db.createCollection("EMPLEADOS");
db.createCollection("DEPARTAMENTOS");
```

```
//probamos de crear los documentos de departamentos y llenamos de datos
try {
    db.DEPARTAMENTOS.insertMany( [
        {nombre: "Departamento 1", presupuesto: 20000},
        {nombre: "Departamento 2", presupuesto: 46000},
        {nombre: "Departamento 3", presupuesto: 50000},
        {nombre: "Departamento 4", presupuesto: 10},
        {nombre: "Departamento 5", presupuesto: 9000}
    ] );
} catch (e) {
    print (e);
}
```

```
//probamos de crear los documentos de empleados, llenamos de datos y referenciamos
la clave foranea
try {
    db.EMPLEADOS.insertMany( [
        {_id: "11111111X", nombre: "Sergi", apellidos: "Rodriguez Utge", departamento:
ObjectId("6536aa065a29557e5a976b65")},
        {_id: "22222222X", nombre: "Marc", apellidos: "Gonzalez Verges",
departamento: ObjectId("6536aa065a29557e5a976b69")},
        {_id: "33333333X", nombre: "Josep", apellidos: "Matas Prados", departamento:
ObjectId("6536aa065a29557e5a976b65")},
        {_id: "44444444X", nombre: "Julia", apellidos: "Pla Agramunt", departamento:
ObjectId("6536aa065a29557e5a976b69")},
        {_id: "55555555X", nombre: "David", apellidos: "Fernandez Gonzalez",
departamento: ObjectId("6536aa065a29557e5a976b65")}
    ] );
} catch (e) {
    print (e);
}
```

Ejercicio 2 - TA17-PT2 (Consultas):

//hacemos busqueda de los empleados con departamento "Departamento 1"

```
const id_departamento = db.DEPARTAMENTOS.findOne({nombre: "Departamento 1"})._id;  
db.EMPLEADOS.find({departamento: id_departamento});
```

//Captura:

```
>_MONGOSH  
  
> const id_departamento = db.DEPARTAMENTOS.findOne({nombre: "Departamento 1"})._id;  
db.EMPLEADOS.find({departamento: id_departamento});  
  
< {  
  _id: '1111111X',  
  nombre: 'Sergi',  
  apellidos: 'Rodriguez Utge',  
  departamento: ObjectId("6536aa065a29557e5a976b65")  
}  
{  
  _id: '3333333X',  
  nombre: 'Josep',  
  apellidos: 'Matas Prados',  
  departamento: ObjectId("6536aa065a29557e5a976b65")  
}  
{  
  _id: '5555555X',  
  nombre: 'David',  
  apellidos: 'Fernandez Gonzalez',  
  departamento: ObjectId("6536aa065a29557e5a976b65")  
}  
}
```

TA17-PT2 >

//hacemos busqueda de articulos ordenados por su nombre de forma alfabetica

```
db.EMPLEADOS.find().sort({nombre:1});
```

//Captura:

```
>_MONGOSH  
  
> db.EMPLEADOS.find().sort({nombre:1});  
  
< {  
  _id: '5555555X',  
  nombre: 'David',  
  apellidos: 'Fernandez Gonzalez',  
  departamento: ObjectId("6536aa065a29557e5a976b65")  
}  
{  
  _id: '3333333X',  
  nombre: 'Josep',  
  apellidos: 'Matas Prados',  
  departamento: ObjectId("6536aa065a29557e5a976b65")  
}  
{  
  _id: '4444444X',  
  nombre: 'Julia',  
  apellidos: 'Pla Agramunt',  
  departamento: ObjectId("6536aa065a29557e5a976b69")  
}  
{  
  _id: '2222222X',  
  nombre: 'Marc',  
  apellidos: 'Gonzalez Verges',  
  departamento: ObjectId("6536aa065a29557e5a976b69")  
}  
{  
  _id: '1111111X',  
  nombre: 'Sergi',  
  apellidos: 'Rodriguez Utge',  
  departamento: ObjectId("6536aa065a29557e5a976b65")  
}  
}
```

TA17-PT2 > |

Ejercicio 3 - TA17-PT3 (Creación e inserciones):

```
//crear o mover a la base de datos  
use TA17-PT3
```

```
//creamos las colecciones  
db.createCollection("ALMACENES");  
db.createCollection("CAJAS");
```

```
//creamos los documentos de almacenes y llenamos de datos  
db.ALMACENES.insertMany( [  
    {lugar: "Tarragona", capacidad: 2},  
    {lugar: "Castellon", capacidad: 3},  
    {lugar: "Zaragoza", capacidad: 1},  
    {lugar: "Madrid", capacidad: 4},  
    {lugar: "Barcelona", capacidad: 5}  
]);
```

```
//creamos los documentos de cajas, llenamos de datos y referenciamos la clave foranea  
db.CAJAS.insertMany( [  
    {contenido: "Play Station 5", valor: 690, almacen:  
ObjectId("6536b0fa5a29557e5a976b6a")},  
    {contenido: "Portatil Asus XP 15", valor: 1500, almacen:  
ObjectId("6536b0fa5a29557e5a976b6e")},  
    {contenido: "Manta termica", valor: 15, almacen:  
ObjectId("6536b0fa5a29557e5a976b6e")},  
    {contenido: "Taladro", valor: 38, almacen: ObjectId("6536b0fa5a29557e5a976b6a")},  
    {contenido: "Dentadura de Halloween", valor: 8, almacen:  
ObjectId("6536b0fa5a29557e5a976b6c")}  
]);
```

Ejercicio 3 - TA17-PT3 (Consultas):

//buscamos las cajas en el almacen ubicado en "Tarragona"

```
const id_almacen = db.ALMACENES.findOne({lugar: "Tarragona"})._id;  
db.CAJAS.find({almacen: id_almacen});
```

//Captura:

```
>_MONGOSH  
  
> const id_almacen = db.ALMACENES.findOne({lugar: "Tarragona"})._id;  
db.CAJAS.find({almacen: id_almacen});  
  
< {  
  _id: ObjectId("6536dd4dcdb217ae322686d6"),  
  contenido: 'Play Station 5',  
  valor: 690,  
  almacen: ObjectId("6536b0fa5a29557e5a976b6a")  
}  
{  
  _id: ObjectId("6536dd4dcdb217ae322686d9"),  
  contenido: 'Taladro',  
  valor: 38,  
  almacen: ObjectId("6536b0fa5a29557e5a976b6a")  
}  
TA17-PT3>
```

//buscamos las cajas en el almacen ubicado en "Tarragona" con valor mayor a 50

```
db.CAJAS.find({almacen: id_almacen, valor:{$gt:50}});
```

//Captura:

```
>_MONGOSH  
  
> db.CAJAS.find({almacen: id_almacen, valor:{$gt:50}});  
  
< {  
  _id: ObjectId("6536dd4dcdb217ae322686d6"),  
  contenido: 'Play Station 5',  
  valor: 690,  
  almacen: ObjectId("6536b0fa5a29557e5a976b6a")  
}  
TA17-PT3> |
```

Ejercicio 4 - TA17-PT4 (Creación e inserciones):

```
//crear o mover a la base de datos
use TA17-PT4
```

```
//creamos la coleccion de Peliculas con un validador para que tenga como requisito el nombre
```

```
db.createCollection("PELICULAS", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      title: "PELICULAS Object Validation",
      required: ["nombre"],
      properties: {
        nombre: {
          bsonType: "string",
          description: "'nombre' must be a string and is required"
        }
      }
    }
  }
});
```

```
//creamos la coleccion de Salas con un validador para que tenga como requisito el nombre
```

```
db.createCollection("SALAS", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      title: "SALAS Object Validation",
      required: ["nombre"],
      properties: {
        nombre: {
          bsonType: "string",
          description: "'nombre' must be a string and is required"
        }
      }
    }
  }
});
```

```
//insertamos datos de una pelicula pero nos saltará un error
```

```
try {
  db.SALAS.insertOne( {nombre:null, calificacion_edad: 6} );
} catch (e) {
  print (e);
}
```

```
//insertamos datos de peliculas y creamos el documento de calificacion_edad al poner los datos
```

```
db.PELICULAS.insertMany( [
  {nombre: "Citizen Kane", calificacion_edad: "PG"},
  {nombre: "Singin in the Rain", calificacion_edad: "G"},
  {nombre: "The Wizard of Oz", calificacion_edad: "G"},
  {nombre: "The Quiet Man", calificacion_edad: null},
  {nombre: "North by Northwest", calificacion_edad: null},
  {nombre: "The Last Tango in Paris", calificacion_edad: "NC-17"},
  {nombre: "Some Like it Hot", calificacion_edad: "PG-13"},
  {nombre: "A Night at the Opera", calificacion_edad: null},
  {nombre: "Citizen King", calificacion_edad: "G"}
]);
```

```
//insertamos datos de salas y referenciamos la clave foranea
```

```
db.SALAS.insertMany( [
  {nombre: "Odeon", pelicula: ObjectId("65370d18cdb217ae322686fe")},
  {nombre: "Imperial", pelicula: ObjectId("65370d18cdb217ae322686fa")},
  {nombre: "Majestic", pelicula: null},
  {nombre: "Royale", pelicula: ObjectId("65370d18cdb217ae322686ff")},
  {nombre: "Paraiso", pelicula: ObjectId("65370d18cdb217ae322686fc")},
  {nombre: "Nickelodeon", pelicula: null}
]);
```

Ejercicio 4 - TA17-PT4 (Consultas):

//insertamos datos de una pelicula y seguidamente la eliminamos

```
db.PELICULAS.insertOne( {nombre: "Uno, Dos, Tres", calificacion_edad: "PG"} );
```

```
db.PELICULAS.deleteOne( {nombre: "Uno, Dos, Tres"} );
```

//con el metodo remove no funciona, esta deprecado

//Captura:

```
>_MONGOSH
> db.PELICULAS.insertOne( {nombre: "Uno, Dos, Tres", calificacion_edad: "PG"} );
< {
  acknowledged: true,
  insertedId: ObjectId("654027677084f4f2464ca7df")
}
> db.PELICULAS.deleteOne( {nombre: "Uno, Dos, Tres"} );
< {
  acknowledged: true,
  deletedCount: 1
}
TA17-PT4>
```

//hacemos constar que todas las PELICULAS no calificadas han sido calificadas 'no recomendables para menores de 13 años'.

```
db.PELICULAS.updateMany({calificacion_edad: null}, {$set: {calificacion_edad: "PG-13"}});
```

//Captura:

```
>_MONGOSH
> db.PELICULAS.updateMany({calificacion_edad: null}, {$set: {calificacion_edad: "PG-13"}});
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
TA17-PT4>
```

//buscamos las salas que tienen películas

```
db.SALAS.find({pelicula: {$ne:null}});
```

//Captura:

```
>_MONGOSH
> db.SALAS.find({pelicula: {$ne:null}});
< [
  {
    _id: ObjectId("65370dcccdb217ae32268703"),
    nombre: 'Odeon',
    pelicula: ObjectId("65370d18cdb217ae322686fe")
  },
  {
    _id: ObjectId("65370dcccdb217ae32268704"),
    nombre: 'Imperial',
    pelicula: ObjectId("65370d18cdb217ae322686fa")
  },
  {
    _id: ObjectId("65370dcccdb217ae32268706"),
    nombre: 'Royale',
    pelicula: ObjectId("65370d18cdb217ae322686ff")
  },
  {
    _id: ObjectId("65370dcccdb217ae32268707"),
    nombre: 'Paraiso',
    pelicula: ObjectId("65370d18cdb217ae322686fc")
  }
]
TA17-PT4>
```


Ejercicio 5 - TA17-PT5 (Creación e inserciones):

```
//crear o mover a la base de datos
use TA17-PT5
```

```
//insertamos datos de despachos y creamos los documentos al poner los datos
```

```
db.DESPACHOS.insertMany( [
    {_id: 1, capacidad:3},
    {_id: 2, capacidad:2},
    {_id: 3, capacidad:5},
    {_id: 4, capacidad:2},
    {_id: 5, capacidad:3},
]);
```

```
//insertamos datos de directores y referenciamos las claves foraneas
```

```
db.DIRECTORES.insertMany( [
    {_id: "11111111", nom_apels: "Sergi Rodriguez Utge", despacho: 1, dni_jefe: "77777777" },
    {_id: "22222222", nom_apels: "Antonio Rodriguez Gonzalez", despacho: 2, dni_jefe:
"55555555"},
    {_id: "33333333", nom_apels: "Maria Peinado Ramos", despacho: 4, dni_jefe:
"77777777"},
    {_id: "44444444", nom_apels: "Julia Cortes Molina", despacho: 5, dni_jefe:
"55555555"},
    {_id: "55555555", nom_apels: "Jose Perez Navarro", despacho: 3, dni_jefe:
"66666666"},
    {_id: "66666666", nom_apels: "Pedro Valero Gimenez", despacho: 4, dni_jefe:
"11111111"},
    {_id: "77777777", nom_apels: "Antonio Ochoa Gil", despacho: 4, dni_jefe: null},
    {_id: "88888888", nom_apels: "Nerea Prados Mata", despacho: 1, dni_jefe:
"55555555"},
    {_id: "99999999", nom_apels: "Juan Peralta Mota", despacho: 1, dni_jefe: "77777777"}
]);
```

Ejercicio 5 - TA17-PT5 (Consultas):

//eliminamos al director "Juan Peralta Mota" con _id "99999999"

```
db.DIRECTORES.deleteOne( {_id: "99999999", nom_apels: "Juan Peralta Mota"} );
```

//Captura:

```
>_MONGOSH
> db.DIRECTORES.remove( {_id: "99999999", nom_apels: "Juan Peralta Mota"} );
< {
  acknowledged: true,
  deletedCount: 1
}
TA17-PT5>
```

//Actualizamos al director con id "88888888" y le cambiamos el dni del jefe a "77777777"

```
db.DIRECTORES.updateOne({_id: "88888888"}, {$set: {dni_jefe: "77777777"}});
```

//Captura:

```
>_MONGOSH
> db.DIRECTORES.updateOne({_id: "88888888"}, {$set: {dni_jefe: "77777777"}});
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
TA17-PT5>
```

//Actualizamos los directores que esten en el despacho 2 y los movemos al 5

```
db.DIRECTORES.updateMany({despacho:2}, {$set: {despacho:5}});
```

//Captura:

```
>_MONGOSH
> db.DIRECTORES.updateMany({despacho:2}, {$set: {despacho:5}});
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
TA17-PT5> |
```

//mostramos las despachos sin directores

```
const array_ids_despachos_con_directores = db.DIRECTORES.distinct("despacho");
db.DESPACHOS.find( { _id: { $nin: array_ids_despachos_con_directores}});
```

//Captura:

```
>_MONGOSH
> const array_ids_despachos_con_directores = db.DIRECTORES.distinct("despacho");
db.DESPACHOS.find( { _id: { $nin: array_ids_despachos_con_directores}});
< {
  _id: 2,
  capacidad: 2
}
TA17-PT5>|
```

//mostramos los despachos que superan su capacidad de directores

```
const ids_despachos_saturados = []; //declaramos array para guardar las ids de los despachos saturados
```

//por cada despacho con empleados

```
db.DIRECTORES.distinct("despacho").forEach(function(value){
```

```
    //si la cantidad de directores que hay en el despacho es mayor que su capacidad se añade id
```

```
    if(db.DIRECTORES.countDocuments({"despacho": value}) >
```

```
db.DESPACHOS.findOne({_id: value}).capacidad){
```

```
        ids_despachos_saturados.push(value); //añadimos id al array
```

```
    }
```

```
});
```

//mostramos datos de despachos saturados

```
db.DESPACHOS.find( { _id: { $in: ids_despachos_saturados}});
```

//Captura:

```
>_MONGOSH
> //mostramos los despachos que superan su capacidad de directores
const ids_despachos_saturados = []; //declaramos array para guardar las ids de los despachos saturados
//por cada despacho con empleados
db.DIRECTORES.distinct("despacho").forEach(function(value){
    //si la cantidad de directores que hay en el despacho es mayor que su capacidad se añade id
    if(db.DIRECTORES.countDocuments({"despacho": value}) > db.DESPACHOS.findOne({_id: value}).capacidad){
        ids_despachos_saturados.push(value); //añadimos id al array
    }
});
//mostramos datos de despachos saturados
db.DESPACHOS.find( { _id: { $in: ids_despachos_saturados}});
< {
  _id: 4,
  capacidad: 2
}
TA17-PT5>|
```

Ejercicio 6 - TA17-PT6 (Creación e inserciones):

//crear o mover a la base de datos

use TA17-PT6

//insertamos datos de proveedores y creamos los documentos al poner los datos

```
db.PROVEEDORES.insertMany( [  
    {_id: 1, nombre: "MSI"},  
    {_id: 2, nombre: "Siemens"},  
    {_id: 3, nombre: "Nvidia"},  
    {_id: 4, nombre: "Intel"},  
    {_id: 5, nombre: "AMD"},  
    {_id: 6, nombre: "Sapphire"}  
]);
```

//insertamos datos de piezas y creamos los documentos al poner los datos

```
db.PIEZAS.insertMany( [  
    {_id: 1, nombre:"MSI Radeon RX"},  
    {_id: 2, nombre:"Tesla K80"},  
    {_id: 3, nombre:"Ventus 2X"},  
    {_id: 4, nombre:"GeForce RTX 4060"},  
    {_id: 5, nombre:"GeForce RTX 3070"},  
    {_id: 6, nombre:"Core i9"},  
    {_id: 7, nombre:"Core i7"},  
    {_id: 8, nombre:"Ryzen 7"},  
    {_id: 9, nombre:"Ryzen 5"},  
    {_id: 10, nombre:"Pulse AMD"}  
]);
```

//insertamos datos de suministra con 2 claves primarias

```
db.SUMINISTRA.insertMany( [  
    {_id:{ codigo_pieza: 1, id_proveedor: 1}, precio:150},  
    {_id:{ codigo_pieza: 1, id_proveedor: 5}, precio:180},  
    {_id:{ codigo_pieza: 2, id_proveedor: 3}, precio:250},  
    {_id:{ codigo_pieza: 3, id_proveedor: 3}, precio:263},  
    {_id:{ codigo_pieza: 4, id_proveedor: 3}, precio:488},  
    {_id:{ codigo_pieza: 5, id_proveedor: 3}, precio:253},  
    {_id:{ codigo_pieza: 4, id_proveedor: 1}, precio:142},  
    {_id:{ codigo_pieza: 5, id_proveedor: 1}, precio:825},  
    {_id:{ codigo_pieza: 6, id_proveedor: 4}, precio:247},  
    {_id:{ codigo_pieza: 7, id_proveedor: 4}, precio:693},  
    {_id:{ codigo_pieza: 8, id_proveedor: 5}, precio:932},  
    {_id:{ codigo_pieza: 9, id_proveedor: 5}, precio:376},  
    {_id:{ codigo_pieza: 10, id_proveedor: 6}, precio:175},  
    {_id:{ codigo_pieza: 10, id_proveedor: 5}, precio:134}  
]);
```

Ejercicio 6 - TA17-PT6 (Consultas):

//mostramos los proveedores con piezas

```
db.PROVEEDORES.find( { _id: { $in: db.SUMINISTRA.distinct("_id.id_proveedor")}}});
```

//Captura:

```
>_MONGOSH
> db.PROVEEDORES.find( { _id: { $in: db.SUMINISTRA.distinct("_id.id_proveedor")}});
< {
  _id: 1,
  nombre: 'MSI'
}
{
  _id: 3,
  nombre: 'Nvidia'
}
{
  _id: 4,
  nombre: 'Intel'
}
{
  _id: 5,
  nombre: 'AMD'
}
{
  _id: 6,
  nombre: 'Sapphire'
}
TA17-PT6>
```

//borramos la pieza numero 10

```
db.PIEZAS.remove({_id: 10});
```

```
db.SUMINISTRA.remove({"_id.codigo_pieza": 10});
```

//Captura:

```
>_MONGOSH
> db.PIEZAS.remove({_id: 10});
db.SUMINISTRA.remove({"_id.codigo_pieza": 10});
< {
  acknowledged: true,
  deletedCount: 2
}
TA17-PT6>
```

//mostramos el precio medio de los departamentos que tienen piezas

```
db.SUMINISTRA.aggregate([{$group : {_id: "$_id.id_proveedor", precio_medio : {$avg : "$precio"}}}]);
```

//Captura:

```
>_MONGOSH
> db.SUMINISTRA.aggregate([{$group : {_id: "$_id.id_proveedor", precio_medio : {$avg : "$precio"}}}]);
< {
  _id: 1,
  precio_medio: 372.3333333333333
}
{
  _id: 3,
  precio_medio: 313.5
}
{
  _id: 4,
  precio_medio: 470
}
{
  _id: 5,
  precio_medio: 496
}
TA17-PT6>|
```

Ejercicio 7 - TA17-PT7 (Creación e inserciones):

//crear o mover a la base de datos

use TA17-PT7

//insertamos datos de productos y creamos los documentos al poner los datos

```
db.PRODUCTOS.insertMany( [  
    {_id: 1, nombre: "Platano de canarias", precio: 1.22},  
    {_id: 2, nombre: "Silla Gamer", precio: 120.05},  
    {_id: 3, nombre: "Televisor Lg", precio: 413.99},  
    {_id: 4, nombre: "Lechuga", precio: 0.80},  
    {_id: 5, nombre: "Libreta", precio: 2.30},  
    {_id: 6, nombre: "CocaCola", precio: 1.10}  
]);
```

//insertamos datos de cajeros y creamos los documentos al poner los datos

```
db.CAJEROS.insertMany( [  
    {_id: 1, nom_apels: "Antonio Fernandez Moreno"},  
    {_id: 2, nom_apels: "Jose Francisco Borbon"},  
    {_id: 3, nom_apels: "Mike Matas Fernandez"},  
    {_id: 4, nom_apels: "Sergi Rodriguez Utge"},  
    {_id: 5, nom_apels: "Alba Fonts Capdevila"},  
    {_id: 6, nom_apels: "Carla Ribas Perea"}  
]);
```

//insertamos datos de maquinas registradoras y creamos los documentos al poner los datos

```
db.MAQUINAS_REGISTRADORAS.insertMany( [  
    {_id: 1, piso: 1},  
    {_id: 2, piso: 1},  
    {_id: 3, piso: 2},  
    {_id: 4, piso: 2},  
    {_id: 5, piso: 3},  
    {_id: 6, piso: 5}  
]);
```

//insertamos datos de ventas con 3 claves primarias

```
db.VENTA.insertMany( [  
    {_id:{ cajero: 1, maquina: 1, producto: 1}},  
    {_id:{ cajero: 3, maquina: 1, producto: 2}},  
    {_id:{ cajero: 2, maquina: 2, producto: 3}},  
    {_id:{ cajero: 1, maquina: 2, producto: 4}},  
    {_id:{ cajero: 5, maquina: 2, producto: 5}},  
    {_id:{ cajero: 2, maquina: 2, producto: 6}},  
    {_id:{ cajero: 1, maquina: 3, producto: 1}},  
    {_id:{ cajero: 4, maquina: 3, producto: 2}},  
    {_id:{ cajero: 4, maquina: 5, producto: 3}},  
    {_id:{ cajero: 6, maquina: 5, producto: 4}}  
]);
```

Ejercicio 7 - TA17-PT7 (Consultas):

//los productos con precio menores o iguales a 100 se añade un 10% de su valor

```
db.PRODUCTOS.updateMany({precio:{$lte:100}}, {$mul: {precio: Decimal128( "1.10" )}});
```

//Captura:

```
>_MONGOSH
> db.PRODUCTOS.updateMany({precio:{$lte:100}}, {$mul: {precio: Decimal128( "1.10" )}});
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
TA17-PT7>
```

//mostramos las maquinas con cajeros

```
db.MAQUINAS_REGISTRADORAS.find( { _id: { $in: db.VENTA.distinct("_id.maquina")}});
```

//Captura:

```
>_MONGOSH
> db.MAQUINAS_REGISTRADORAS.find( { _id: { $in: db.VENTA.distinct("_id.maquina")}});
< {
  _id: 1,
  piso: 1
}
{
  _id: 2,
  piso: 1
}
{
  _id: 3,
  piso: 2
}
{
  _id: 5,
  piso: 3
}
TA17-PT7>
```


Ejercicio 8 - TA17-PT8 (Creación e inserciones):

```
//crear o mover a la base de datos
use TA17-PT8
```

```
//insertamos datos de facultades y creamos los documentos al poner los datos
```

```
db.FACULTAD.insertMany( [
    {_id: 1, nombre: "Facultad Pepito"},
    {_id: 2, nombre: "Facultad Antoniete Vila"},
    {_id: 3, nombre: "Nostrum Pupum"},
    {_id: 4, nombre: "Amigos de la Tierra"},
    {_id: 5, nombre: "Alegria Facutativa"},
    {_id: 6, nombre: "Facultad Josete"},
    {_id: 7, nombre: "Facultad de derecho"}
]);
```

```
//insertamos datos de investigadores con 1 "clave foranea"
```

```
db.INVESTIGADORES.insertMany( [
    {_id: "11111111", nom_apels: "Antonio Fernandez Moreno", facultad: 1},
    {_id: "22222222", nom_apels: "Jose Francisco Borbon", facultad: 1},
    {_id: "33333333", nom_apels: "Mike Matas Fernandez", facultad: 1},
    {_id: "44444444", nom_apels: "Alba Fonts Capdevila", facultad: 3},
    {_id: "55555555", nom_apels: "Carla Ribas Perea", facultad: 3},
    {_id: "66666666", nom_apels: "David Maza Olivar", facultad: 3},
    {_id: "77777777", nom_apels: "Adrian Marconi Mendoza", facultad: 5},
    {_id: "88888888", nom_apels: "Julia Cantos Ochoa", facultad: 5},
    {_id: "99999999", nom_apels: "Daniel Sierra Robles", facultad: 5}
]);
```

```
//insertamos datos de equipos con 1 "clave foranea"
```

```
db.EQUIPOS.insertMany( [
    {_id: "ABCD", nombre: "Equipo 1", facultad: 1},
    {_id: "BCDE", nombre: "Equipo 2", facultad: 2},
    {_id: "CDEF", nombre: "Equipo 3", facultad: 3},
    {_id: "DEFG", nombre: "Equipo 4", facultad: 4},
    {_id: "EFGH", nombre: "Equipo 5", facultad: 5},
    {_id: "FGHI", nombre: "Equipo 6", facultad: 6},
    {_id: "GHIJ", nombre: "Equipo 7", facultad: 7}
]);
```

```
//insertamos datos de reservas con 2 claves primarias
```

```
db.RESERVA.insertMany( [
    {_id:{ dni: "11111111", num_serie: "ABCD"}, comienzo:"2022-02-01", fin:"2023-02-01"},
    {_id:{ dni: "22222222", num_serie: "ABCD"}, comienzo:"2022-03-01", fin:"2023-03-01"},
    {_id:{ dni: "33333333", num_serie: "ABCD"}, comienzo:"2022-04-01", fin:"2023-04-01"},
    {_id:{ dni: "44444444", num_serie: "CDEF"}, comienzo:"2022-03-01", fin:"2022-06-01"},
    {_id:{ dni: "55555555", num_serie: "CDEF"}, comienzo:"2022-03-01", fin:"2022-06-15"},
    {_id:{ dni: "66666666", num_serie: "CDEF"}, comienzo:"2022-03-01", fin:"2022-06-15"},
    {_id:{ dni: "77777777", num_serie: "EFGH"}, comienzo:"2023-01-15", fin:"2023-12-03"},
    {_id:{ dni: "88888888", num_serie: "EFGH"}, comienzo:"2023-02-15", fin:"2023-12-03"},
    {_id:{ dni: "99999999", num_serie: "EFGH"}, comienzo:"2023-03-15", fin:"2023-12-03"}
]);
```

Ejercicio 8 - TA17-PT8 (Consultas):

//mostramos los integrantes del equipo "Equipo 1"

const codigo_equipo = db.EQUIPOS.findOne({nombre: "Equipo 1"})._id; //guardamos el codigo de equipo correspondiente a "Equipo 1"

const dnis_equipo = db.RESERVA.distinct("_id.dni", {"_id.num_serie" : codigo_equipo});

//guardamos el array con dnis de los investigadores del equipo

db.INVESTIGADORES.find({"_id": {\$in: dnis_equipo}});

//Captura:

```
>_MONGOSH
> const codigo_equipo = db.EQUIPOS.findOne({nombre: "Equipo 1"})._id; //guardamos el codigo de equipo correspondiente a "Equipo 1"
const dnis_equipo = db.RESERVA.distinct("_id.dni", {"_id.num_serie" : codigo_equipo}); //guardamos el array con dnis de los investigadores del equipo
db.INVESTIGADORES.find({"_id": {$in: dnis_equipo}});
< [
  {
    _id: '11111111',
    nom_apels: 'Antonio Fernandez Moreno',
    facultad: 1
  },
  {
    _id: '22222222',
    nom_apels: 'Jose Francisco Borbon',
    facultad: 1
  },
  {
    _id: '33333333',
    nom_apels: 'Mike Matas Fernandez',
    facultad: 1
  }
]
TA17-PT8>
```

//mostramos todos los investigadores ordenados de forma alfabetica por su nombre
db.INVESTIGADORES.find().sort({nom_apels:1});

//Captura:

```
>_MONGOSH
> db.INVESTIGADORES.find().sort({nom_apels:1});
< {
  _id: '77777777',
  nom_apels: 'Adrian Marconi Mendoza',
  facultad: 5
}
{
  _id: '44444444',
  nom_apels: 'Alba Fonts Capdevila',
  facultad: 3
}
{
  _id: '11111111',
  nom_apels: 'Antonio Fernandez Moreno',
  facultad: 1
}
{
  _id: '55555555',
  nom_apels: 'Carla Ribas Perea',
  facultad: 3
}
{
  _id: '99999999',
  nom_apels: 'Daniel Sierra Robles',
  facultad: 5
}
{
  _id: '66666666',
  nom_apels: 'David Maza Olivar',
  facultad: 3
}
{
  _id: '22222222',
  nom_apels: 'Jose Francisco Borbon',
  facultad: 1
}
{
  _id: '88888888',
  nom_apels: 'Julia Cantos Ochoa',
  facultad: 5
}
{
  _id: '33333333',
  nom_apels: 'Mike Matas Fernandez',
  facultad: 1
}
}
```

TA17-PT8> |

Ejercicio 9 - TA17-PT9 (Creación e inserciones):

```
//crear o mover a la base de datos
use TA17-PT9
```

```
//insertamos datos de científicos y creamos los documentos al poner los datos
```

```
db.CIENTIFICOS.insertMany( [
    {nom_apels: "Antonio Fernandez Moreno"},
    {nom_apels: "Jose Francisco Borbon"},
    {nom_apels: "Mike Matas Fernandez"},
    {nom_apels: "Alba Fonts Capdevila"},
    {nom_apels: "Carla Ribas Perea"},
    {nom_apels: "David Maza Olivar"},
    {nom_apels: "Adrian Marconi Mendoza"},
    {nom_apels: "Julia Cantos Ochoa"},
    {nom_apels: "Daniel Sierra Robles"}
]);
```

```
//insertamos datos de proyectos y creamos los documentos al poner los datos
```

```
db.PROYECTOS.insertMany( [
    {nombre: "Proyecto 1", horas: 158},
    {nombre: "Proyecto 2", horas: 176},
    {nombre: "Proyecto 3", horas: 257},
    {nombre: "Proyecto 4", horas: 258},
    {nombre: "Proyecto 5", horas: 679},
    {nombre: "Proyecto 6", horas: 86},
    {nombre: "Proyecto 7", horas: 375},
    {nombre: "Proyecto 8", horas: 284},
    {nombre: "Proyecto 9", horas: 965}
]);
```

```
//insertamos datos de asignados a
```

```
db.ASIGNADO_A.insertMany( [
    {cientifico: ObjectId("65380497929d44aa3dbb65b7"), proyecto:
    ObjectId("65380497929d44aa3dbb65c0")},
    {cientifico: ObjectId("65380497929d44aa3dbb65b8"), proyecto:
    ObjectId("65380497929d44aa3dbb65c3")},
    {cientifico: ObjectId("65380497929d44aa3dbb65b9"), proyecto:
    ObjectId("65380497929d44aa3dbb65c5")},
    {cientifico: ObjectId("65380497929d44aa3dbb65ba"), proyecto:
    ObjectId("65380497929d44aa3dbb65c0")},
    {cientifico: ObjectId("65380497929d44aa3dbb65bb"), proyecto:
    ObjectId("65380497929d44aa3dbb65c7")},
    {cientifico: ObjectId("65380497929d44aa3dbb65bc"), proyecto:
    ObjectId("65380497929d44aa3dbb65c0")},
    {cientifico: ObjectId("65380497929d44aa3dbb65bd"), proyecto:
    ObjectId("65380497929d44aa3dbb65c7")},
    {cientifico: ObjectId("65380497929d44aa3dbb65be"), proyecto:
    ObjectId("65380497929d44aa3dbb65c3")},
    {cientifico: ObjectId("65380497929d44aa3dbb65bf"), proyecto:
    ObjectId("65380497929d44aa3dbb65c0")},
    {cientifico: ObjectId("65380497929d44aa3dbb65be"), proyecto:
    ObjectId("65380497929d44aa3dbb65c5")},
    {cientifico: ObjectId("65380497929d44aa3dbb65b7"), proyecto:
    ObjectId("65380497929d44aa3dbb65c5")}
]);
```

Ejercicio 9 - TA17-PT9 (Consultas):

```
//mostramos los cientificos que trabajan en el proyecto "Proyecto 1" en orden alfabetico
const id_proyecto = db.PROYECTOS.findOne({nombre: "Proyecto 1"})._id; //guardamos oid del proyecto
const ids_cientificos_de_proyecto = db.ASIGNADO_A.distinct("cientifico", {"proyecto": id_proyecto}); //guardamos array de los oids de los cientificos del proyecto
db.CIENTIFICOS.find({"_id": {$in: ids_cientificos_de_proyecto}}).sort({nom_apels:1});
```

//Captura:

```
>_MONGOSH
> const id_proyecto = db.PROYECTOS.findOne({nombre: "Proyecto 1"})._id; //guardamos oid del proyecto
const ids_cientificos_de_proyecto = db.ASIGNADO_A.distinct("cientifico", {"proyecto": id_proyecto}); //guardamos array de los oids de los cientificos del proyecto
db.CIENTIFICOS.find({"_id": {$in: ids_cientificos_de_proyecto}}).sort({nom_apels:1});
< {
  _id: ObjectId("65380497929d44aa3dbb65ba"),
  nom_apels: 'Alba Fonts Capdevila'
}
{
  _id: ObjectId("65380497929d44aa3dbb65b7"),
  nom_apels: 'Antonio Fernandez Moreno'
}
{
  _id: ObjectId("65380497929d44aa3dbb65bf"),
  nom_apels: 'Daniel Sierra Robles'
}
{
  _id: ObjectId("65380497929d44aa3dbb65bc"),
  nom_apels: 'David Maza Olivar'
}
TA17-PT9>
```

//mostramos los proyectos sin cientificos

```
db.PROYECTOS.find( { _id: { $nin: db.ASIGNADO_A.distinct("proyecto") } });
```

//Captura:

```
>_MONGOSH
> db.PROYECTOS.find( { _id: { $nin: db.ASIGNADO_A.distinct("proyecto") } });
< {
  _id: ObjectId("65380497929d44aa3dbb65c1"),
  nombre: 'Proyecto 2',
  horas: 176
}
{
  _id: ObjectId("65380497929d44aa3dbb65c2"),
  nombre: 'Proyecto 3',
  horas: 257
}
{
  _id: ObjectId("65380497929d44aa3dbb65c4"),
  nombre: 'Proyecto 5',
  horas: 679
}
{
  _id: ObjectId("65380497929d44aa3dbb65c6"),
  nombre: 'Proyecto 7',
  horas: 375
}
{
  _id: ObjectId("65380497929d44aa3dbb65c8"),
  nombre: 'Proyecto 9',
  horas: 965
}
TA17-PT9>
```