

## ACADT UT6 P2

### Agregación Pipeline:

- Utilizando la colección **trabajadores** realiza las siguientes consultas:
  - Visualiza la edad media, la media de salario y el número de trabajadores que hayan tenido una prima de 30 o de 80.
  - Visualiza por población el número de trabajadores, el salario medio y el máximo salario.
  - Visualiza el nombre, apel y ape2 del empleado que tiene máximo salario.
  - A partir de la consulta anterior, obtén ahora el nombre, ape1, ape2 y salario del empleado que tiene máximo salario por cada población.
- Utilizando la colección **cities** resuelva las siguientes consultas utilizando el pipeline de agregaciones:
  - Mostrar los 3 países con mayor número de ciudades
  - Devolver los 3 países con mayor número de habitantes
  - Calcular y devolver la población total de España, Francia e Italia, en orden descendente.
  - Calcular la latitud y longitud media para cada zona horaria europea.
  - Devolver los países en la posición 5ª y 6ª si ordenamos su población total en orden descendente.
  - Muestra el total de timezone que hay en Europa.

```
1.A
db.trabajadores.aggregate([
  {
    $match: {
      primas: { $in: [30, 80] },
    },
  },
  {
    $group: {
      _id: "empleados",
      edadMedia: { $avg: "$edad" },
      mediaSalario: { $avg: "$salario" },
      numeroTrabajadores: { $sum: 1 },
    },
  },
])

1.B
db.trabajadores.aggregate([
  {
    $group: {
      _id: "$direccion.poblacion",
      numeroTrabajadores: { $sum: 1 },
      salarioMedio: { $avg: "$salario" },
      salarioMaximo: { $max: "$salario" },
    },
  },
])

1.C
db.trabajadores.aggregate([
  {
    $sort: {
      salario: -1,
    },
  },
  {
    $limit: 1,
  },
  {
    $project: {
      nombre: "$nombre.nomb",
      primerApe: "$nombre.ape1",
      segundoApe: "$nombre.ape2",
      salario: "$salario",
    },
  },
])

1.D
db.trabajadores.aggregate([
  {
    $sort: {
      salario: -1,
    },
  },
  {
    $group: {
      _id: "$direccion.poblacion",
      nombre: { $first: "$nombre.nomb" },
      primerApe: { $first: "$nombre.ape1" },
      segundoApe: { $first: "$nombre.ape2" },
      salario: { $first: "$salario" },
    },
  },
])

2.A
db.cities.aggregate([
  {
    $group: {
      _id: '$country',
      numeroCiudades: { $sum: 1 }
    },
  },
  {
    $sort: { numeroCiudades: -1 },
    $limit: 3
  }
])

2.B
db.cities.aggregate([
  {
    $group: {
      _id: '$country',
      numeroHabitantes: { $sum: '$population' }
    },
  },
  {
    $sort: { numeroHabitantes: -1 },
    $limit: 3
  }
])

2.C
db.cities.aggregate([
  {
    $match: {
      country: { $in: ['ES', 'FR', 'IT'] }
    },
  },
  {
    $group: {
      _id: '$country',
      numeroHabitantes: { $sum: '$population' }
    },
  },
  {
    $sort: { numeroHabitantes: -1 }
  }
])

2.D
db.cities.aggregate([
  {
    $match: {
      timezone: /Europe/i
    },
  },
  {
    $group: {
      _id: '$timezone',
      latitud: { $avg: '$location.latitude' },
      longitud: { $avg: '$location.longitude' }
    },
  },
])

2.E
db.cities.aggregate([
  {$group: {_id: "$country", totalPopulation: { $sum: "$population" }}},
  { $sort: { totalPopulation: -1 } },
  { $skip: 4 },
  { $limit: 2 }
])

2.F
db.cities.aggregate([
  { $match: { timezone: { $regex: /^Europe/ } } }, // Filtra las ciudades en zonas horarias europeas
  { $group: { _id: "$timezone" } }, // Agrupa por zona horaria
  { $count: "totalTimezones" } // Cuenta las zonas horarias
])
```