

# Formulario Académico: Consultas Avanzadas en MongoDB

## 1. Consultas Avanzadas con MongoDB

### 1.1. 1. Agrupar por país y calcular porcentaje

```
[
  { $project: { pais: 1, _id: 0 } },
  { $group: {
    _id: "$pais",
    total: { $count: {} }
  } },
  { $addFields: {
    porcentaje: {
      $multiply: [
        { $divide: ["$total", 2000] },
        100
      ]
    }
  } }
]
```

Listing 1: Calcular porcentaje por país basado en 2000 registros

### 1.2. 2. Dominio de correos más común por país

```
[
  { $addFields: {
    ext_correo: {
      $arrayElemAt: [
        { $split: ["$email", "@"] },
        1
      ]
    }
  } },
  { $group: {
    _id: { pais: "$pais", ext_correo: "$ext_correo" },
    total: { $count: {} }
  } },
  { $sort: { total: -1 } },
  { $group: {
    _id: "$_id.pais",
    ext_correo: { $first: "$_id.ext_correo" },
    total: { $first: "$total" }
  } }
]
```

```
}  
}  
]
```

Listing 2: Dominio de correo electrónico más usado por país

### 1.3. 3. Comparar edad promedio por género

```
[  
  { $group: {  
    _id: "$genero.genero",  
    promedio_edad: { $avg: "$edad" }  
  }  
},  
{ $sort: { promedio_edad: -1 } }  
]
```

Listing 3: Edad promedio por género

### 1.4. 4. Pacientes con múltiples condiciones médicas

```
[  
  { $project: {  
    nombre: 1,  
    num_condiciones: { $size: "$condiciones" }  
  }  
},  
{ $match: {  
  num_condiciones: { $gt: 2 }  
}  
}  
]
```

Listing 4: Filtrar pacientes con más de dos enfermedades

### 1.5. 5. Búsqueda de diagnósticos por palabra clave

```
{  
  diagnostico: {  
    $regex: "diabetes",  
    $options: "i"  
  }  
}
```

Listing 5: Buscar "diabetes" diagnóstico sin importar mayúsculas

### 1.6. 6. Consultar pacientes con IMC fuera del rango normal

```
{
  $or: [
    { imc: { $lt: 18.5 } },
    { imc: { $gt: 24.9 } }
  ]
}
```

Listing 6: Pacientes con bajo peso o sobrepeso

### 1.7. 7. Usar \$lookup para obtener información cruzada

```
[
  {
    $lookup: {
      from: "medicos",
      localField: "id_medico",
      foreignField: "_id",
      as: "info_medico"
    }
  },
  { $unwind: "$info_medico" },
  { $project: {
    nombre_paciente: "$nombre",
    medico: "$info_medico.nombre"
  }
}
]
```

Listing 7: Relacionar pacientes con médicos

### 1.8. 8. Clasificar pacientes según rango etario

```
[
  {
    $addFields: {
      grupo_edad: {
        $switch: {
          branches: [
            { case: { $lt: ["$edad", 18] }, then: "Niño" },
            { case: { $lt: ["$edad", 60] }, then: "Adulto" }
          ],
          default: "Adulto mayor"
        }
      }
    }
  },
  {
    $group: {
      _id: "$grupo_edad",
      total: { $sum: 1 }
    }
  }
]
```

```

    }
  }
]

```

Listing 8: Clasificación por rango de edad

### 1.9. 9. Mostrar porcentaje de vacunados por dosis

```

[
  {
    $group: {
      _id: null,
      total: { $sum: 1 },
      dosis1: {
        $sum: { $cond: [ { $eq: ["$dosis_covid.dosis_1", "Si"] },
          1, 0 ] }
      },
      dosis2: {
        $sum: { $cond: [ { $eq: ["$dosis_covid.dosis_2", "Si"] },
          1, 0 ] }
      },
      dosis3: {
        $sum: { $cond: [ { $eq: ["$dosis_covid.dosis_3", "Si"] },
          1, 0 ] }
      }
    }
  },
  {
    $project: {
      dosis1: 1,
      dosis2: 1,
      dosis3: 1,
      porcentaje1: { $multiply: [ { $divide: ["$dosis1", "$total" ] }, 100 ] },
      porcentaje2: { $multiply: [ { $divide: ["$dosis2", "$total" ] }, 100 ] },
      porcentaje3: { $multiply: [ { $divide: ["$dosis3", "$total" ] }, 100 ] }
    }
  }
]

```

Listing 9: Pacientes con dosis 1, 2 y 3

### 1.10. 10. Facet para múltiples análisis simultáneos

```

[
  {
    $facet: {
      genero: [

```

```
    { $group: { _id: "$genero.genero", total: { $sum: 1 } } }
  ],
  orientacion: [
    { $group: { _id: "$genero.orientacion_sexual", total: {
      $sum: 1 } } }
  ],
  identidad: [
    { $group: { _id: "$genero.identidad_genero", total: {
      $sum: 1 } } }
  ]
}
]
```

Listing 10: Comparar varios campos a la vez

### 1.11. 11. Uso de collation para comparar texto de forma insensible a mayúsculas

```
db.peliculas.find(
  { title: "A Corner in Wheat" },
  { title: 1, year: 1 }
).collation({ locale: "en", strength: 1 })
```

Listing 11: Uso de collation para buscar título sin distinguir mayúsculas

```
db.peliculas.find().sort({ title: 1 }).collation({ locale: "en",
  strength: 1 })
```

Listing 12: Ordenar títulos de películas sin distinguir mayúsculas

```
db.usuarios.find(
  { nombre: "Jose" }
).collation({ locale: "es", strength: 1 })
```

Listing 13: Buscar usuarios por nombre sin importar acentos ni mayúsculas

```
db.estudiantes.find(
  { nombre: { $gt: "ana" } }
).collation({ locale: "es", strength: 1 })
```

Listing 14: Comparación insensible a mayúsculas y diacríticos

**Nota sobre collation:** El parámetro `collation` permite controlar cómo se comparan y ordenan cadenas de texto en MongoDB. El atributo `strength` define el nivel de sensibilidad:

- **1:** Ignora mayúsculas y acentos (insensible a mayúsculas y diacríticos).
- **2:** Sensible a acentos, pero no a mayúsculas.
- **3:** Sensible a mayúsculas y acentos (por defecto).

- **4:** Sensible incluso a variantes como la puntuación.

**Nota Final:** Este formulario presenta un conjunto avanzado de consultas en MongoDB útiles para análisis exploratorio, estadísticas descriptivas y relaciones entre documentos. Se sugiere su uso en contextos académicos o evaluativos para estudiantes con conocimientos intermedios o avanzados en bases de datos NoSQL.