

## Explicación requerimientos funcionales

Requerimiento 1:

```
db.reservas.aggregate([
  { $unwind: "$habitacion.clientes" },
  { $unwind: "$habitacion.clientes.consumos" },
  {
    $match: {
      $expr: {
        $and: [
          {
            $gte: [
              { $toDate: "$habitacion.clientes.consumos.fecha" },
              {
                $subtract: [
                  new Date(),
                  { $multiply: [365, 24, 60, 60, 1000] }
                ]
              }
            ]
          }
        ]
      },
      {
        $lte: [
          { $toDate: "$habitacion.clientes.consumos.fecha" },
          new Date()
        ]
      }
    ]
  }
],
```

```
{
  $group: {
    _id: { hotel: "$habitacion.hotel", habitacion: "$habitacion.numero" },
    totalConsumption: { $sum: "$habitacion.clientes.consumos.precio" }
  }
},
{ $project: { _id: 0, hotel: "$_id.hotel", numero: "$_id.habitacion", totalConsumption: 1 } }
})
```

#### Explicación:

Esta consulta de agregación realiza una descomposición de documentos mediante `\$unwind` para trabajar con los elementos de los arreglos `habitacion.clientes` y `habitacion.clientes.consumos`. Luego, con `\$match`, filtra los documentos basándose en la fecha de consumo dentro de un rango definido. La etapa `\$group` agrupa los documentos por el nombre del hotel y número de habitación, calculando el total de consumo por habitación. Finalmente, `\$project` configura la salida, eliminando el campo `\_id` y renombrando las claves pertinentes. La consulta ofrece un resumen del consumo total por habitación en cada hotel, con campos resultantes como `hotel`, `numero` y `totalConsumption`.

#### Requerimiento 2:

```
db.reservas.aggregate([
  { $match: { $expr: { $and: [ { $gte: [ { $toDate: "$inicio" }, { $subtract: [new Date(), {$multiply:[365, 24, 60, 60, 1000]}] } ] }, { $lte: [ { $toDate: "$fin" }, new Date() ] } ] } } },
  {
    $group: {
      _id: {
        hotel: "$habitacion.hotel",
        numero: "$habitacion.numero"
      },
      totalDays: {
        $sum: {
          $dateDiff: {
            startDate: { $toDate: "$inicio" },
            endDate: { $toDate: "$fin" },
```

```

        unit: "day"
    }
}
}
}
},
{
    $project: {
        _id: 0,
        hotel: "$_id.hotel",
        numero: "$_id.numero",
        porcentajeOcupacion: {
            $round: [
                {
                    $multiply: [
                        { $divide: ["$totalDays", 365] },
                        100
                    ]
                },
                2
            ]
        }
    }
}
])

```

#### Explicacion:

Este requerimiento empieza haciendo un match, donde queremos sacar todas las reservas que pasaron el último año. Para esto sacamos la fecha actual y sacamos la fecha de hace un año restándole a dicha fecha 365 días. Después revisamos que las fechas que buscamos estén entre la fecha de hace un año y la fecha actual (match). Ya con esto, podemos hacer una agrupación por habitaciones y hoteles dado que toda la información sobre estancias en cada habitación esté agrupada. De acá guardamos los días que duró cada reserva en cada habitación encontrando la

diferencia entre el día de inicio y fin de la reserva. Con la anterior operación ya hecha, podemos sumar todos los días de ocupación de cada habitación. Finalmente, en la proyección mostramos las celdas que nos interesan y hacemos una función de agregación donde dividimos el número total de días entre 365 y multiplicamos por 100 para obtener el porcentaje de ocupación.

Requerimiento 3:

```
var clienteId = 5;
var dInicio = "2023-11-21T00:00:00Z";
var dFin = "2023-11-29T00:00:00Z";

db.reservas.aggregate([
  { $unwind: "$habitacion.clientes" },
  { $match: { "habitacion.clientes.id": clienteId } },
  { $unwind: "$habitacion.clientes.consumos" },
  { $match: { "habitacion.clientes.consumos.fecha": { $gte: dInicio, $lt: dFin } } },
  {
    $group: {
      _id: "$habitacion.clientes.id",
      nombreC: { $first: "$habitacion.clientes.nombre" },
      totalConsumos: { $sum: "$habitacion.clientes.consumos.precio" }
    }
  },
  { $project: { _id: 0, clienteId: "$_id", clienteNombre: "$nombreC", totalConsumos: 1 } }
]);
```

Explicación:

Esta consulta de agregación busca obtener el resumen de consumos para un cliente específico en un rango de fechas dado. Primero, se utiliza `\$unwind` para descomponer los arrays `habitacion.clientes` y `habitacion.clientes.consumos`, permitiendo un análisis más detallado. Luego, se aplica `\$match` para filtrar documentos en base al ID del cliente y las fechas de consumo dentro del rango proporcionado. La etapa `\$group` agrupa los resultados por el ID del cliente, extrayendo el nombre del cliente y sumando los precios de los consumos. Finalmente, `\$project` configura la salida de la consulta, renombrando y seleccionando los campos relevantes, como `clienteId`, `clienteNombre` y `totalConsumos`.

Requerimiento 7:

```
db.reservas.aggregate([
  {
    $unwind:
      /**
       * path: Path to the array field.
       * includeArrayIndex: Optional name for index.
       * preserveNullAndEmptyArrays: Optional
```

```

    * toggle to unwind null and empty values.
    */
{
  path: "$habitacion.clientes",
  includeArrayIndex: "string",
  preserveNullAndEmptyArrays: true,
},
},
{
  $group: {
    _id: "$habitacion.clientes.id",
    _nombre: {
      $first: "$habitacion.clientes.nombre",
    },
    _fechas: {
      $push: "$habitacion.clientes.entrada",
    },
    _salidas: {
      $push: "$habitacion.clientes.salida",
    },
  },
},
{
  $project: {
    _nombre: "$_nombre",
    buencliente: [
      {
        $anyElementTrue: {
          $map: {
            input: "$_fechas",
            in: {
              $cond: {
                if: {
                  $and: [
                    {
                      $gte: [
                        "$$this",
                        "2023-01-01T00:00:00Z",
                      ],
                    },
                    {
                      $lt: [
                        "$$this",
                        "2023-04-01T00:00:00Z",
                      ],
                    },
                  ],
                },
              },
            ],
          },
        },
      ],
    ],
  },
},

```

```

        then: 1,
        else: 0,
      },
    },
  },
},
{
  $anyElementTrue: {
    $map: {
      input: "$_fechas",
      in: {
        $cond: {
          if: {
            $and: [
              {
                $gte: [
                  "$$this",
                  "2023-04-01T00:00:00Z",
                ],
              },
              {
                $lt: [
                  "$$this",
                  "2023-07-01T00:00:00Z",
                ],
              },
            ],
          },
          then: 1,
          else: 0,
        },
      },
    },
  },
},
{
  $anyElementTrue: {
    $map: {
      input: "$_fechas",
      in: {
        $cond: {
          if: {
            $and: [
              {
                $gte: [
                  "$$this",
                  "2023-07-01T00:00:00Z",

```

```

    ],
    },
    {
      $lt: [
        "$$this",
        "2023-10-01T00:00:00Z",
      ],
    },
  ],
},
then: 1,
else: 0,
},
},
},
},
{
  $anyElementTrue: {
    $map: {
      input: "$_fechas",
      in: {
        $cond: {
          if: {
            $and: [
              {
                $gte: [
                  "$$this",
                  "2023-10-01T00:00:00Z",
                ],
              },
              {
                $lte: [
                  "$$this",
                  "2023-12-31T00:00:00Z",
                ],
              },
            ],
          },
          then: 1,
          else: 0,
        },
      },
    },
  },
},
},
},
},
],
},

```

```

    },
    {
      $project: {
        _nombre: 1,
        allTrue: {
          $gte: [
            {
              $size: {
                $filter: {
                  input: "$buencliente",
                  as: "bool",
                  cond: {
                    $eq: ["$$bool", true],
                  },
                },
              },
            },
            {
              $size: {
                $filter: {
                  input: "$malcliente",
                  as: "bool",
                  cond: {
                    $eq: ["$$bool", true],
                  },
                },
              },
            },
          ],
        },
      },
    },
  ],
},
{
  $match: {
    allTrue: true,
  },
},
{
  $project: {
    _id: 1,
    _nombre: 1,
  },
},
])

```

#### Explicación:

Lo primero que hacemos es un unwind para aplanar el documento y poder usar clientes. Después hacemos una agrupación por id para poder tener toda la información de cada cliente único, en esta agrupación también guardamos los nombres de los clientes con los ids correspondientes de la agrupación y hacemos push sobre la fecha de inicio y fin de la reserva, con lo cual creamos documentos que podemos consultar más adelante. Lo siguiente que hacemos es proyectar nuestra información junto con un arreglo que guarda 1s y 0s, con los cuales comprobaremos si un cliente se quedó en los 4 trimestres. Para lograr esto hacemos un condicional que guarda 1s cuando se encuentre que un cliente se quedó en el hotel en el primer trimestre del año (logramos esto haciendo lte y gte entre las fechas de inicio de la reserva y los límites del trimestre), después repetimos este proceso con los otros tres trimestres obteniendo un arreglo con 4 posiciones con 1s y



0s. Después hacemos una proyección de la información que mostraremos al final y del resultado de nuestra función “allTrue” que revisa si el arreglo está completamente lleno de 1s, si “allTrue” es cierto entonces proyectamos estos resultados. Finalmente, proyectamos el id y nombre de los resultados anteriores.