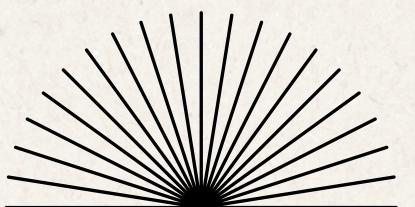


BASE DE DATOS STUFFIES

Integrantes
Leonel Pavez
Ignacio Garcia
Miguel Torres



Poblar Base de datos

Insert productos

```
print("--- Insertando Productos ---");

db.productos.insertMany([
```

Aqui aplicamos cada producto que tenemos en la tienda con un insertMany donde estan todos los detalles del producto

```
db.productos.insertMany([
{
  _id: 1,
  nombre: "Hoodie Boxy Fit White Dice V2",
  categoria: "polerones",
  precio: 39990,
  imagen: "https://stuffiesconcept.com/cdn/shop/files/WhiteDice1.png",
  imagen_hover: "https://stuffiesconcept.com/cdn/shop/files/WhiteDice2.png",
  descripcion: "Polerón boxy fit White Dice V2.",
  destacado: true,
  stock: {
    total: 10,
    tallas: [
      { talla: "S", cantidad: 3 },
      { talla: "M", cantidad: 5 },
      { talla: "L", cantidad: 2 },
      { talla: "XL", cantidad: 0 }
    ]
  },
  tags: ["new", "invierno"]
},
```

Insert aplicados en la base de datos

```
{
  _id: 2,
  nombre: "Star Player Blue Team T-Shirt",
  categoria: "poleras",
  precio: 10990,
  imagen: "https://stuffiesconcept.com/cdn/shop/files/1_594f01e1-55e5-4516-b0af-d2befc1aa113.png",
  imagen_hover: "https://stuffiesconcept.com/cdn/shop/files/2_221c9cfc-6049-4eb1-b7ec-3b19bd755c48.png",
  descripcion: "Polera Star Player Blue Team.",
  destacado: false,
  stock: {
    total: 13,
    tallas: [
      { talla: "M", cantidad: 8 },
      { talla: "L", cantidad: 4 },
      { talla: "XL", cantidad: 1 }
    ]
  },
  tags: ["futbol", "streetwear"]
},
```

Insert Cliente

Insert aplicados en la base de datos

```
print("--- Insertando Clientes ---");

db.clientes.insertMany([
```

Aquí aplicamos el mismo formato que tenemos en insertProductos pero para clientes

```
print("--- Insertando Clientes ---");

db.clientes.insertMany([
{
  _id: 100,
  nombre: "Juan Pérez",
  email: "juan.perez@example.com",
  telefono: "+56 9 1111 1111",
  direccion: {
    calle: "Av. Siempre Viva 123",
    ciudad: "Viña del Mar",
    region: "Valparaíso"
  }
},
```

```
{
  _id: 101,
  nombre: "María González",
  email: "maria.gonzalez@example.com",
  telefono: "+56 9 2222 2222",
  direccion: {
    calle: "Los Robles 456",
    ciudad: "Santiago",
    region: "RM"
  }
},
```

Insert Pedido

```
print("--- Insertando Pedidos ---");

db.pedidos.insertMany([
```

Y siguiendo con el mismo metodo que los de mas insert, seguimos utilizando insertMany para poblar la base de datos

```
print("--- Insertando Pedidos ---);

db.pedidos.insertMany([
{
  _id: 5000,
  clienteId: 100,
  fecha: new Date("2024-11-20"),
  estado: "CREADO",
  items: [
    {
      productoId: 1,
      nombreProducto: "Hoodie Boxy Fit White Dice V2",
      talla: "M",
      cantidad: 1,
      precioUnitario: 39990
    },
    {
      productoId: 3,
      nombreProducto: "Stella Boxy-Slim Black Tee",
      talla: "L",
      cantidad: 2,
      precioUnitario: 15990
    }
  ],
  total: 71970
},
```

Aqui podemos ver que al crear el pedido llamamos los datos específicos que queremos de los datos de productos

Crud de la base de datos

Create

En esta parte creamos los documentos dentro de la base de datos

```
// Insertar un nuevo producto
print("--- Insertando nuevo producto ---");
db.productos.insertOne({
  _id: 5,
  nombre: "Hoodie Oversize Black",
  categoria: "polerones",
  precio: 44990,
  imagen: "https://example.com/hoodie-black.png",
  imagen_hover: "https://example.com/hoodie-black-2.png",
  descripcion: "Hoodie oversize color negro.",
  destacado: false,
  stock: {
    total: 15,
    tallas: [
      { talla: "M", cantidad: 5 },
      { talla: "L", cantidad: 7 },
      { talla: "XL", cantidad: 3 }
    ]
  },
  tags: ["nuevo", "negro"]
});
```

```
// Insertar un nuevo pedido
print("\n--- Insertando nuevo pedido ---");
db_pedidos.insertOne({
  _id: 5002,
  clienteId: 102,
  fecha: new Date(),
  estado: "CREADO",
  items: [
    {
      productId: 4,
      nombreProducto: "Gorro Beanie Clásico",
      talla: "Única",
      cantidad: 2,
      precioUnitario: 9990
    }
  ],
  total: 19980
});
```

al final del otro código utilizamos findOne para verificar que se haya insertado correctamente

```
db.productos.findOne({ _id: 5 });
```

```
db_pedidos.findOne({ _id: 5002 });
```

Aquí aplicamos los insertOne para insertar un nuevo producto y para insertar un nuevo pedido

Read

En el read leemos la informacion almacenada en MongoDB

Para consultar los datos aplicamos find lo que nos ayuda a buscar especificamente lo que buscamos

```
// Consultar todos los productos
print("--- Todos los productos ---");
db.productos.find({}, { nombre: 1, precio: 1, categoria: 1 }).pretty();

// Consultar productos con precio > 15000
print("\n--- Productos con precio > $15,000 ---");
db.productos.find({ precio: { $gt: 15000 } }, { nombre: 1, precio: 1 }).pretty();

// Buscar productos destacados
print("\n--- Productos destacados ---");
db.productos.find({ destacado: true }, { nombre: 1, destacado: 1 }).pretty();

// Buscar productos de categoría "poleras"
print("\n--- Productos categoría 'poleras' ---");
db.productos.find({ categoria: "poleras" }, { nombre: 1, precio: 1 }).pretty();

// Buscar clientes de Santiago
print("\n--- Clientes de Santiago ---");
db.clientes.find({ "direccion.ciudad": "Santiago" }, { nombre: 1, "direccion.ciudad": 1 }).pretty();
```

usamos lookup que es como un JOIN entre pedido y clientes, lo que nos ayuda a combinar la informacion de cliente dentro del pedido

```
// Consulta con agregación: pedidos con nombre de cliente
print("\n--- Pedidos con información del cliente ---");
db_pedidos.aggregate([
  {
    $lookup: {
      from: "clientes",
      localField: "clienteId",
      foreignField: "_id",
      as: "cliente_info"
    }
  },
  {
    $unwind: "$cliente_info"
  },
  {
    $project: {
      pedido_id: "$_id",
      fecha: 1,
      estado: 1,
      nombre_cliente: "$cliente_info.nombre",
      total: 1
    }
  }
]).pretty();
```

Update

Aqui es donde podemos modificar los datos

Lo primero que hacemos aqui es utilizar el find para validar el producto que se quiere modificar, luego con un updateOne modificamos el precio (como en el ejemplo) y por ultimo volvemos a validar el objeto

```
// Actualizar precio de un producto
print("--- Antes del cambio ---");
db.productos.findOne({ _id: 2 }, { nombre: 1, precio: 1 });

db.productos.updateOne(
  { _id: 2 },
  { $set: { precio: 11990 } }
);

print("\n--- Después del cambio ---");
db.productos.findOne({ _id: 2 }, { nombre: 1, precio: 1 });
```

Delete

En el apartado Delete, primero creamos un producto temporal, luego lo verificamos utilizando findOne, despues eliminamos el producto con deleteOne y por ultimo volvemos a utilizar findOne para validar que el producto fue eliminado

```
// Crear producto temporal para eliminar
print("--- Creando producto temporal ---");
db.productos.insertOne({
  _id: 99,
  nombre: "Producto Temporal",
  categoria: "test",
  precio: 1000,
  destacado: false,
  stock: { total: 0, tallas: [] },
  tags: ["test"]
});

print("Verificando que existe:");
db.productos.findOne({ _id: 99 });

// Eliminar el producto temporal
print("\n--- Eliminando producto temporal ---");
db.productos.deleteOne({ _id: 99 });

print("Verificando que fue eliminado:");
db.productos.findOne({ _id: 99 }); // Devuelve null
```

Consultas Avanzadas

Una de las consultas avanzadas que tenemos nos ayuda a ver las tallas disponibles por producto

```
// Unwind de tallas (desnormalizar)
print("--- Tallas disponibles por producto ---");
db.productos.aggregate([
  {
    $project: {
      nombre: 1,
      tallas: "$stock.tallas"
    }
  },
  {
    $unwind: "$tallas"
  },
  {
    $project: {
      nombre: 1,
      talla: "$tallas.talla",
      cantidad: "$tallas.cantidad"
    }
  }
]).pretty();
```