

Entrega 3: Proyecto Sistemas transaccionales

1. Revise el caso de estudio propuesto. Identifique los elementos fundamentales que hacen parte del negocio que se describe.

- **1. Sucursales**

Descripción: SuperAndes tiene varias sucursales distribuidas en distintas ciudades, cada una identificada por su nombre, tamaño en metros cuadrados (m2), y la ciudad donde se ubica.

Detalles: Cada sucursal opera con independencia, aunque sigue los principios generales de la compañía. Las ciudades cuentan con múltiples sucursales.

- **2. Bodegas**

Descripción: Cada sucursal tiene una o varias bodegas que se utilizan para almacenar productos antes de ponerlos a la venta.

Detalles: Se maneja información como el nombre, el tamaño en m2, y la capacidad de almacenamiento de cada producto, así como el costo promedio del producto en cada bodega, que varía entre bodegas.

- **3. Productos**

Descripción: Los productos ofrecidos incluyen información detallada como nombre, costo en bodega, precio unitario de venta, presentación, especificaciones de empaque, y categoría.

Categorías: Los productos se clasifican en varias categorías (perecederos, no perecederos, aseo, congelados, prendas de vestir, muebles, herramientas, electrodomésticos). Cada categoría tiene información sobre código, nombre, descripción y características de almacenamiento.

- **4. Proveedores**

Descripción: Las sucursales adquieren productos de distintos proveedores, los cuales se identifican por su NIT, nombre, dirección, nombre de contacto y teléfono.

Relación: Los proveedores pueden ofrecer múltiples productos y una sucursal puede comprar los mismos productos a diferentes proveedores.

- **5. Órdenes de Compra**

Descripción: Las órdenes de compra se emiten cuando una sucursal necesita reabastecer productos.

Detalles: Cada orden contiene información sobre productos, cantidades, precios acordados con el proveedor, fecha esperada de entrega, y estado (vigente, entregada, anulada).

- **6. Recepción de Productos**

Descripción: La recepción de productos registra la llegada de productos comprados a una bodega de una sucursal.

Detalles: Esta recepción implica la actualización de la cantidad de existencias y del costo promedio de los productos en la bodega.

- **7. Niveles de Inventario y Reorden**

Descripción: Cada sucursal tiene un nivel mínimo de reorden definido para garantizar la disponibilidad de productos.

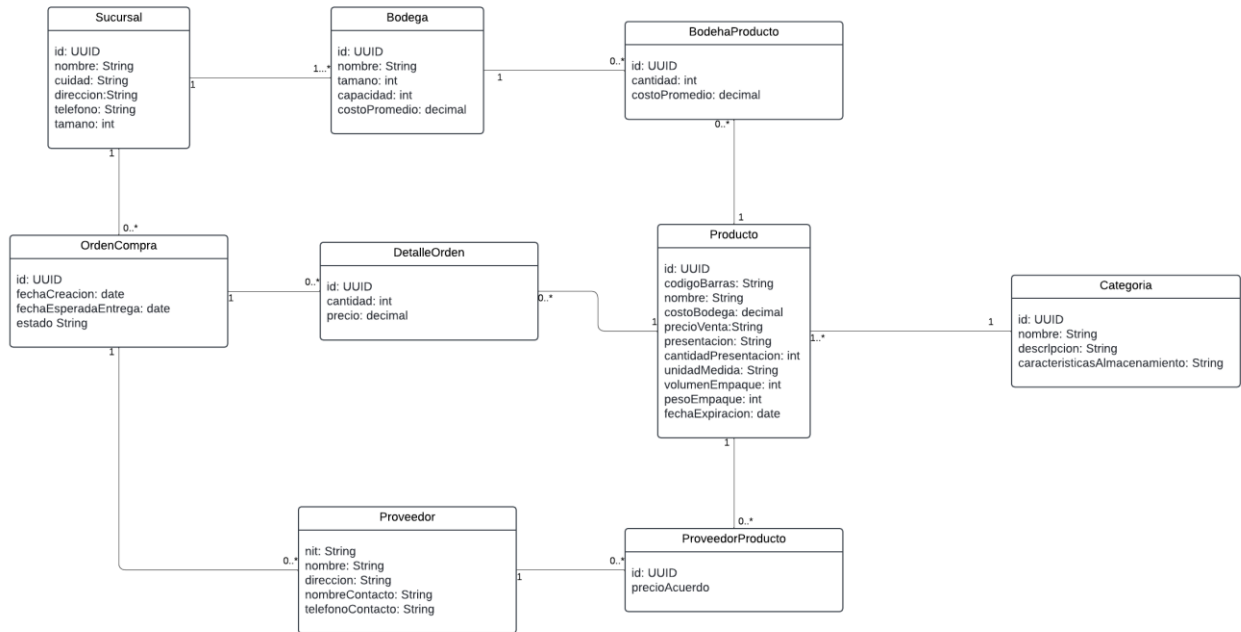
Detalles: Cuando las existencias de un producto bajan de este nivel, se procede a hacer una solicitud de compra.

- **8. Clientes y Ventas**

Descripción: Aunque no se menciona a detalle, se puede inferir que el negocio tiene clientes a quienes vende los productos disponibles en las sucursales.

- Estos elementos forman el eje principal del negocio de SuperAndes, definiendo cómo las sucursales gestionan inventarios, almacenan productos, se relacionan con proveedores y manejan órdenes de compra para garantizar la disponibilidad de productos a los clientes.

2. Análisis y modelo conceptual:



3. a. a. Identificación de entidades y sus atributos

Las entidades y sus atributos fueron ya identificados y son los siguientes:

Entidades:

Sucursal:

- id (UUID)
- nombre (String)
- ciudad (String)
- direccion (String)
- telefono (String)
- tamaño (int)

Bodega:

- id (UUID)
- nombre (String)
- tamaño (int)
- capacidad (int)
- costoPromedio (decimal)

Producto:

- codigoBarras (String)
- nombre (String)
- costoBodega (decimal)
- precioVenta (decimal)
- presentacion (String)
- cantidadPresentacion (int)
- unidadMedida (String)

- volumenEmpaque (int)
- pesoEmpaque (int)
- fechaExpiracion (date)

Categoría:

- id (UUID)
- nombre (String)
- descripcion (String)
- característicasAlmacenamiento (String)

Proveedor:

- nit (String)
- nombre (String)
- direccion (String)
- nombreContacto (String)
- telefonoContacto (String)

OrdenCompra:

- id (UUID)
- fechaCreacion (date)
- fechaEsperadaEntrega (date)
- estado (String)

DetalleOrden:

- id (UUID)
- cantidad (int)
- precio (decimal)

BodegaProducto (Clase Intermedia):

- id (UUID)
- cantidad (int)
- costoPromedio (decimal)

ProveedorProducto (Clase Intermedia):

- id (UUID)
- precioAcuerdo (decimal)

a.b. Cuantificación de las Entidades:

- **Sucursales:** ~150 sucursales
- **Bodegas:** ~900 bodegas (6 por cada sucursal, aproximadamente)
- **Proveedores:** ~10,000 proveedores
- **Productos:** ~20,000 productos
- **Órdenes de Compra:** ~70,000 órdenes anuales (~200,000 en una ventana de 3 años)
- **Detalle de Órdenes:** Cada orden de compra tiene múltiples detalles, estimado en ~500,000 en total
- **Relación BodegaProducto:** ~900 bodegas y cada una almacena múltiples productos (~100 productos por bodega)

- **Relación ProveedorProducto:** ~10,000 proveedores suministrando múltiples productos (~200,000 registros)

a.c. Análisis de operaciones de lectura y escritura:

Entidad	Lectura (R)	Escritura (W)
Sucursal	Bajas	Muy bajas
Bodega	Bajas	Muy bajas
Proveedor	Altas	Moderadas
Producto	Moderadas	Moderadas
OrdenCompra	Muy altas	Altas
DetalleOrden	Altas	Altas
BodegaProducto	Moderadas	Altas
ProveedorProducto	Moderadas	Moderadas

a.d. Cuantificación de operaciones de lectura y escritura:

Entidad	Cantidad de Registros	Cantidad Lectura (R)	Cantidad Escritura (W)
Sucursal	150	~1/semana	~1/mes
Bodega	900	~1/semana	~1/mes
Proveedor	10,000	~200/día	~10/día
Producto	20,000	~500/día	~50/día
OrdenCompra	200,000	~1,000/día	~200/día
DetalleOrden	500,000	~1,500/día	~500/día
BodegaProducto	90,000	~300/día	~100/día
ProveedorProducto	200,000	~400/día	~150/día

b. Describan las colecciones de datos y las relaciones entre ellas (NoSQL) que corresponden al modelo conceptual UML propuesto. Para ello, presenten lo siguiente:

b.a y b.c:

1. Sucursal

Descripción: Representa cada una de las sucursales de SuperAndes. Incluye la información básica sobre la ubicación y características de la sucursal.

Atributos:

- id: Identificador único de la sucursal (UUID).
- nombre: Nombre de la sucursal (String).
- ciudad: Ciudad donde se ubica la sucursal (String).
- direccion: Dirección de la sucursal (String).
- telefono: Teléfono de contacto de la sucursal (String).
- tamano: Tamaño de la sucursal en metros cuadrados (int).

Relaciones:

- Tiene una relación "uno a muchos" con la clase Bodega.
- Tiene una relación "uno a muchos" con la clase OrdenCompra.

2. Bodega

Descripción: Representa una bodega perteneciente a una sucursal, donde se almacenan los productos antes de ponerlos a la venta.

Atributos:

- id: Identificador único de la bodega (UUID).
- nombre: Nombre de la bodega (String).
- tamano: Tamaño de la bodega en metros cuadrados (int).
- capacidad: Capacidad de almacenamiento en términos de cantidad (int).
- costoPromedio: Costo promedio de los productos almacenados en la bodega (decimal).

Relaciones:

- Pertenece a una Sucursal (relación "muchos a uno").
- Tiene una relación "muchos a muchos" con la clase Producto, ya que una bodega almacena varios productos y un producto puede estar en varias bodegas. Esta relación se representa mediante una clase intermedia BodegaProducto.

3. Producto

Descripción: Representa un producto que se encuentra disponible para la venta en SuperAndes.

Atributos:

- codigoBarras: Código de barras único del producto (String).
- nombre: Nombre del producto (String).
- costoBodega: Costo del producto en la bodega (decimal).
- precioVenta: Precio de venta unitario del producto (decimal).

- presentacion: Descripción de la presentación del producto (String).
- cantidadPresentacion: Cantidad en la presentación (int).
- unidadMedida: Unidad de medida del producto (string).
- volumenEmpaque: Volumen del empaque del producto en cm³ (int).
- pesoEmpaque: Peso del empaque del producto en gramos (int).
- fechaExpiracion: Fecha de expiración del producto (date).

Relaciones:

- Pertenece a una Categoría (relación "muchos a uno").
- Tiene una relación "muchos a muchos" con la clase Bodega mediante la clase BodegaProducto.
- Tiene una relación "muchos a muchos" con la clase Proveedor a través de la clase ProveedorProducto.

4. Categoría

Descripción: Representa una categoría a la cual pertenece uno o varios productos. Las categorías ayudan a organizar y definir características comunes de los productos.

Atributos:

- id: Identificador único de la categoría (UUID).
- nombre: Nombre de la categoría (String).
- descripcion: Descripción de la categoría (String).
- caracteristicasAlmacenamiento: Características de almacenamiento de los productos de la categoría (String).

Relaciones:

- Tiene una relación "uno a muchos" con la clase Producto, ya que una categoría puede tener varios productos.

5. Proveedor

Descripción: Representa a un proveedor que suministra productos a las sucursales de SuperAndes.

Atributos:

- nit: NIT del proveedor (String).
- nombre: Nombre del proveedor (String).
- direccion: Dirección del proveedor (String).
- nombreContacto: Nombre de la persona de contacto del proveedor (String).
- telefonoContacto: Teléfono de la persona de contacto del proveedor (String).

Relaciones:

- Tiene una relación "muchos a muchos" con la clase Producto mediante la clase ProveedorProducto.
- Tiene una relación "uno a muchos" con la clase OrdenCompra.

6. OrdenCompra

Descripción: Representa una orden de compra emitida por una sucursal hacia un proveedor, para solicitar productos específicos.

Atributos:

- id: Identificador único de la orden de compra (UUID).
- fechaCreacion: Fecha de creación de la orden (date).
- fechaEsperadaEntrega: Fecha esperada de entrega de los productos (date).
- estado: Estado de la orden de compra (String: "vigente", "entregada", "anulada", etc).

Relaciones:

- Pertenece a una Sucursal (relación "muchos a uno").
- Pertenece a un Proveedor (relación "muchos a uno").
- Tiene una relación "uno a muchos" con la clase DetalleOrden.

7. DetalleOrden

Descripción: Representa el detalle de una orden de compra, especificando la cantidad y el precio de los productos solicitados.

Atributos:

- id: Identificador único del detalle de orden (UUID).
- cantidad: Cantidad del producto solicitado (int).
- precio: Precio del producto solicitado (decimal).

Relaciones:

- Pertenece a una OrdenCompra (relación "muchos a uno"). Se refiere a un Producto (relación "muchos a uno").

8. BodegaProducto (Clase Intermedia)

Descripción: Gestiona la relación entre **bodega** y **producto** para representar los productos almacenados en cada bodega.

Atributos:

- id: Identificador único de la relación (UUID).
- cantidad: Cantidad disponible del producto en la bodega (int).
- costoPromedio: Costo promedio del producto en esa bodega (decimal).

Relaciones:

- Pertenece a una Bodega (relación "muchos a uno").
- Pertenece a un Producto (relación "muchos a uno").

9. ProveedorProducto (Clase Intermedia)

Descripción: Gestiona la relación entre **proveedor** y **producto**, representando los productos que un proveedor puede ofrecer y el precio acordado.

Atributos:

- id: Identificador único de la relación (UUID).
- precioAcuerdo: Precio acordado con el proveedor para el producto (decimal).

Relaciones:

- Pertenece a un Proveedor (relación "muchos a uno").
- Pertenece a un Producto (relación "muchos a uno").

b.c. El análisis de selección de esquema de asociación (referenciado o embebido) para cada relación entre entidades. Para ello use la tabla de análisis vista en clase, la cual se retoma en el anexo C, junto con los resultados del análisis de la carga de trabajo (workload), descrita antes.

Relación Sucursal - Bodega

- **Esquema recomendado:** Referenciado.
- **Justificación:**
 - **Simplicidad:** La relación "uno a muchos" entre sucursal y bodega se maneja mejor con referencias para mantener un modelo de datos limpio y escalable.
 - **Crecimiento del Documento:** Embedir bodegas dentro de sucursales podría hacer crecer los documentos sin límite si hay muchas bodegas.
 - **Cardinalidad:** Dado que cada sucursal puede tener múltiples bodegas, la cardinalidad es alta en el lado hijo, por lo que un esquema referenciado es más adecuado.

Relación Bodega - Producto (BodegaProducto)

- **Esquema recomendado:** Referenciado mediante una colección intermedia (BodegaProducto).
- **Justificación:**
 - **Cardinalidad:** La relación es "muchos a muchos", lo que hace necesario el uso de una colección intermedia para gestionar los datos de cada producto almacenado en múltiples bodegas.

- **Consulta y Actualización:** Las operaciones de consulta y actualización sobre productos y bodegas se realizan de manera separada, lo cual favorece el uso de referencias.

Relación Producto - Categoría

- **Esquema recomendado:** Referenciado.
- **Justificación:**
 - **Individualidad:** Los productos deben existir sin depender directamente de la categoría, ya que cada producto tiene atributos individuales que no cambian en función de su categoría.
 - **Tamaño del Documento:** Embedir categorías en los productos no aportaría simplicidad y aumentaría el tamaño innecesariamente.

Relación Proveedor - Producto (ProveedorProducto)

- **Esquema recomendado:** Referenciado mediante una colección intermedia (ProveedorProducto).
- **Justificación:**
 - **Cardinalidad y Crecimiento:** La relación "muchos a muchos" entre proveedores y productos implica alta cardinalidad y posibles cambios frecuentes, por lo que un esquema referenciado es más flexible.
 - **Tamaño del Documento:** Mantener la relación como referenciada evita documentos grandes y facilita la actualización.

Relación Sucursal - OrdenCompra

- **Esquema recomendado:** Referenciado.
- **Justificación:**
 - **Atomicidad de Consulta:** Las órdenes de compra se consultan de manera separada de la información de la sucursal, lo cual favorece el uso de referencias.
 - **Crecimiento del Documento:** Las órdenes de compra podrían crecer sin límite, haciendo que el embebido no sea una buena opción.

Relación OrdenCompra - DetalleOrden

- **Esquema recomendado:** Embebido.
- **Justificación:**

- **Simplicidad:** Los detalles de una orden están estrechamente ligados a la orden de compra en sí, lo cual justifica embeadirlos para facilitar el acceso conjunto.
- **Consulta y Actualización:** Los detalles se consultan y actualizan junto con la orden, lo cual favorece un esquema embebido.

Relación BodegaProducto y ProveedorProducto

- **Esquema recomendado:** Referenciado.
- **Justificación:**
 - **Cardinalidad:** Ambas relaciones son "muchos a muchos", lo cual requiere colecciones intermedias para gestionar de forma eficiente las asociaciones entre bodegas-productos y proveedores-productos.

b.d. Una descripción gráfica usando Json de cada relación entre entidades en donde presente un ejemplo de datos junto con el esquema de asociación usado (referenciado o embebido). En el anexo D se muestra un ejemplo de lo que se requiere.

Relación Sucursal - Bodega (Referenciado)

Descripción: Cada sucursal tiene una o más bodegas. Se utiliza un esquema referenciado para mantener la relación de manera escalable.

Codigo:

```
{
  "Sucursal": {
    "id": "UUID-1234",
    "nombre": "Sucursal Norte",
    "ciudad": "Bogotá",
    "direccion": "Calle 123 #45-67",
    "telefono": "1234567890",
    "tamano": 500,
    "bodegas": [
      "UUID-Bodega-1",
      "UUID-Bodega-2"
    ]
  },
  "Bodega": {
    "id": "UUID-Bodega-1",
    "nombre": "Bodega Principal",
```

```
"tamano": 300,  
"capacidad": 10000,  
"costoPromedio": 15000.5,  
"sucursal": "UUID-1234"  
}  
}
```

2. Relación Bodega - Producto (Referenciado mediante Clase Intermedia BodegaProducto)

Descripción: La relación es "muchos a muchos" y se gestiona a través de la clase intermedia BodegaProducto.

Codigo:

```
{  
  "BodegaProducto": {  
    "id": "UUID-BodegaProducto-1",  
    "bodegalid": "UUID-Bodega-1",  
    "productoid": "UUID-Producto-1",  
    "cantidad": 500,  
    "costoPromedio": 16000.0  
  },  
  "Bodega": {  
    "id": "UUID-Bodega-1",  
    "nombre": "Bodega Principal",  
    "productos": [  
      "UUID-BodegaProducto-1"  
    ]  
  },  
  "Producto": {  
    "id": "UUID-Producto-1",  
    "codigoBarras": "123456789012",  
    "nombre": "Arroz Super Premium",  
    "costoBodega": 16000.0,  
    "precioVenta": 20000.0  
  }  
}
```

3. Relación Producto - Categoría (Referenciado)

Descripción: Un producto pertenece a una categoría y se hace mediante una referencia para evitar la duplicación de datos.

Codigo:

```
{
  "Producto": {
    "id": "UUID-Producto-1",
    "codigoBarras": "123456789012",
    "nombre": "Arroz Super Premium",
    "categoria": "UUID-Categoria-1"
  },
  "Categoria": {
    "id": "UUID-Categoria-1",
    "nombre": "Alimentos",
    "descripcion": "Productos de consumo básico",
    "caracteristicasAlmacenamiento": "Lugar seco y fresco"
  }
}
```

4. Relación Proveedor - Producto (Referenciado mediante Clase Intermedia ProveedorProducto)

Descripción: La relación "muchos a muchos" se gestiona a través de la clase intermedia ProveedorProducto.

Codigo:

```
{
  "ProveedorProducto": {
    "id": "UUID-ProveedorProducto-1",
    "proveedorId": "UUID-Proveedor-1",
    "productoId": "UUID-Producto-1",
    "precioAcuerdo": 15000.0
  },
  "Proveedor": {
    "nit": "900123456-7",
    "nombre": "Distribuidora Los Andes",
    "productos": [
      "UUID-ProveedorProducto-1"
    ]
  }
}
```

```
},  
"Producto": {  
  "id": "UUID-Producto-1",  
  "codigoBarras": "123456789012",  
  "nombre": "Arroz Super Premium"  
}  
}
```

5. Relación Sucursal - OrdenCompra (Referenciado)

Descripción: Cada sucursal tiene varias órdenes de compra. Se mantiene como referencia para un acceso independiente.

Codigo:

```
{  
  "Sucursal": {  
    "id": "UUID-1234",  
    "nombre": "Sucursal Norte",  
    "ordenesCompra": [  
      "UUID-OrdenCompra-1",  
      "UUID-OrdenCompra-2"  
    ]  
  },  
  "OrdenCompra": {  
    "id": "UUID-OrdenCompra-1",  
    "fechaCreacion": "2024-11-01",  
    "fechaEsperadaEntrega": "2024-11-10",  
    "estado": "vigente",  
    "sucursal": "UUID-1234"  
  }  
}
```

6. Relación OrdenCompra - DetalleOrden (Embeber)

Descripción: Los detalles de una orden de compra están embebidos dentro de la orden para facilitar la consulta conjunta.

Codigo:

```
{  
  "OrdenCompra": {
```

```

    "id": "UUID-OrdenCompra-1",
    "fechaCreacion": "2024-11-01",
    "fechaEsperadaEntrega": "2024-11-10",
    "estado": "vigente",
    "detalles": [
      {
        "id": "UUID-DetalleOrden-1",
        "productoid": "UUID-Producto-1",
        "cantidad": 100,
        "precio": 20000.0
      },
      {
        "id": "UUID-DetalleOrden-2",
        "productoid": "UUID-Producto-2",
        "cantidad": 50,
        "precio": 50000.0
      }
    ]
  }
}

```

7. Relación BodegaProducto y ProveedorProducto (Referenciado)

Descripción: Estas relaciones son gestionadas mediante clases intermedias para poder mantener la asociación entre bodegas y productos, así como entre proveedores y productos.

Codigo:

```

{
  "BodegaProducto": {
    "id": "UUID-BodegaProducto-1",
    "bodegaid": "UUID-Bodega-1",
    "productoid": "UUID-Producto-1",
    "cantidad": 500,
    "costoPromedio": 16000.0
  },
  "ProveedorProducto": {
    "id": "UUID-ProveedorProducto-1",
    "proveedorid": "UUID-Proveedor-1",
    "productoid": "UUID-Producto-1",

```

```
"precioAcuerdo": 15000.0
}
}
```

c. Cree en MongoDB las colecciones principales de su base según lo descrito antes. Puede usar Compass o Mongo Shell (Mongosh) para realizar este proceso. Guarde lo hecho en un archivo. Anexe a los entregables los archivos con los scripts utilizados
Sentencia en MONGOSH:

```
// Crear colección Sucursal
db.createCollection("Sucursal");

// Crear colección Bodega
db.createCollection("Bodega");

// Crear colección Producto
db.createCollection("Producto");

// Crear colección Categoría
db.createCollection("Categoría");

// Crear colección Proveedor
db.createCollection("Proveedor");

// Crear colección OrdenCompra
db.createCollection("OrdenCompra");

// Crear colección DetalleOrden
db.createCollection("DetalleOrden");

// Crear colección BodegaProducto (Clase Intermedia)
db.createCollection("BodegaProducto");

// Crear colección ProveedorProducto (Clase Intermedia)
db.createCollection("ProveedorProducto");
```

d. Cree los esquemas de validación para cada colección. Puede usar Compass o Mongo Shell (Mongosh) para realizar este proceso. Guarde lo hecho en un archivo. Anexe a los entregables los archivos con los scripts utilizados.

(Ambos vinculos son al mismo doc con el codigo del punto, si no sirve el primero intenta ingresar al segundo con el correo uniandes)

Vinculos:

[Proyecto Sistemas Transaccionales punto 3.docx](#)

[Proyecto Sistemas Transaccionales punto 3](#)

4. - RF3:

```
>_MONGOSH
> use test
< already on db test
> // Intentar insertar un documento en la colección Proveedores sin cumplir con el esquema
db.Proveedor.insertOne({
  nombre: "Proveedor Fallido", // Correcto
  // Falta la dirección requerida
  nombreContacto: "Contacto Fallido", // Correcto
  // Falta el teléfono de contacto requerido
});
✖ ▶ MongoServerError: Document failed validation
Atlas atlas-9on5rm-shard-0 [primary] test>
```

Codigo:

// Intentar insertar un documento en la colección Proveedores sin cumplir con el esquema

```
db.Proveedor.insertOne({
  nombre: "Proveedor Fallido", // Correcto
  // Falta la dirección requerida
  nombreContacto: "Contacto Fallido", // Correcto
  // Falta el teléfono de contacto requerido
});
```

RF6:

```
> db.OrdenCompra.insertOne({
  // Falta `fechaCreacion`, que es requerida
  fechaEsperadaEntrega: new Date("2024-12-31"), // Correcto
  estado: "vigente" // Correcto
  // Falta `fechaCreacion`
});
✖ ▶ MongoServerError: Document failed validation
Atlas atlas-9on5rm-shard-0 [primary] test>
```

Codigo:

```
db.OrdenCompra.insertOne({
  // Falta `fechaCreacion`, que es requerida
  fechaEsperadaEntrega: new Date("2024-12-31"), // Correcto
  estado: "vigente" // Correcto
  // Falta `fechaCreacion`
});
```

- Aquí está el código de MONGOSH con el que poblamos inicialmente la base de datos:
[Poblar los datos](#)
- Por si acaso, no se si hay que entregar esto, aca esta la prueba de que todos los RF sirven:
[Pruebas de todos los RF.docx](#)