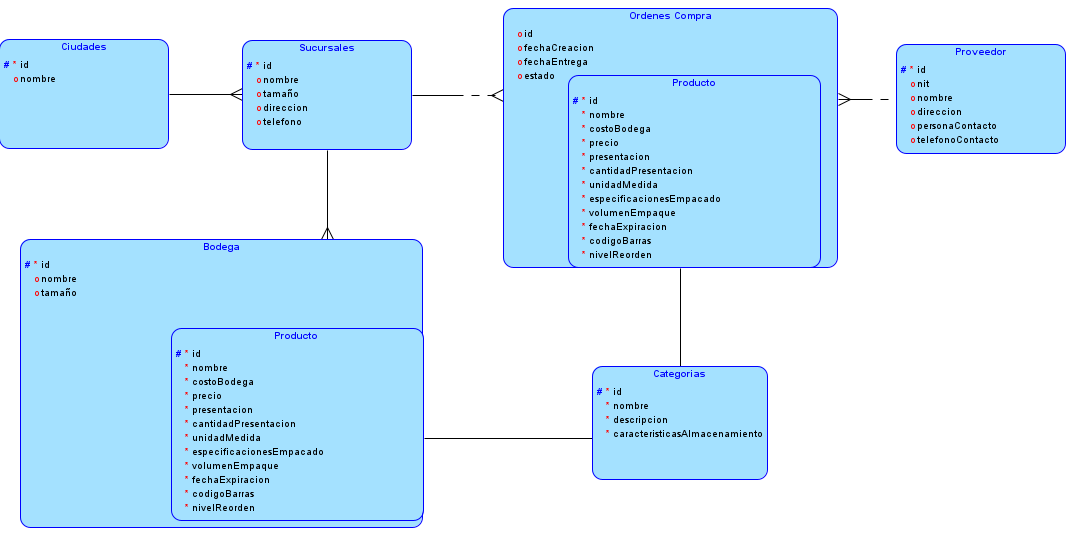
**Documento Entrega 4 del Proyecto**

**2 ) Análisis y modelo conceptual**



**3.a Análisis de la carga de trabajo**

1. **Identificación de las entidades y atributos:**
   * **Entidad: Sucursales**
     + Atributos: ID, nombre, tamaño, dirección, teléfono, ID\_ciudad.
   * **Entidad: Ciudades**
     + Atributos: ID, nombre.
   * **Entidad: Bodega**
     + Atributos: ID, nombre, tamaño, capacidad, ID\_sucursal, productos.
   * **Entidad: Producto**
     + Atributos: ID, nombre, costo en bodega, precio, presentación, cantidadPresentacion, unidadMedida, especificacionesEmpacado, volumenEmpaque, fechaExpiracion, código de barras, nivelReorden, ID\_categoria.
   * **Entidad: Categoría**
     + Atributos: ID, nombre, descripción, característicasAlmacenamiento.
   * **Entidad: Proveedor**
     + Atributos: ID, NIT, nombre, dirección, personaContacto, telefonoContacto.
   * **Entidad: Orden de Compra**
     + Atributos: ID, fecha de creación, fecha de entrega, estado, ID\_sucursal, ID\_proveedor, productos.

1. **Cuantificación de las entidades:**
   * Sucursales: 150.
   * Bodegas: 900.
   * Productos: 20,000.
   * Categorías: 8.
   * Proveedores: 10,000.
   * Órdenes de compra (anuales): ~70,000 (200,000 en 3 años).
2. **Análisis de operaciones de lectura y escritura para cada entidad:**

|  |  |  |  |
| --- | --- | --- | --- |
| Entidades | Operación | Información necesaria | Tipo |
| Sucursales | Crear una sucursal | Nombre, tamaño, dirección, teléfono, ID\_ciudad | Escritura |
| Sucursales | Consultar la información básica de una sucursal | ID, nombre, tamaño, dirección, teléfono, ID\_ciudad | Lectura |
| Ciudades | Consultar la información de una ciudad | ID, nombre | Lectura |
| Bodegas | Crear una bodega | Nombre, tamaño, capacidad, ID\_sucursal, lista de productos | Escritura |
| Bodegas | Consultar detalles de una bodega | ID, nombre, tamaño, capacidad, ID\_sucursal, lista de productos | Lectura |
| Bodegas | Borrar una bodega | ID | Escritura |
| Productos | Crear un producto | Nombre, costo en bodega, precio, presentación, cantidadPresentacion, unidadMedida, especificacionesEmpacado, volumenEmpaque, fechaExpiracion, código de barras, nivelReorden, ID\_categoria | Escritura |
| Productos | Consultar detalles de un producto | ID, nombre, costo en bodega, precio, presentación, ID\_categoria | Lectura |
| Productos | Actualizar detalles de un producto | ID, información actualizada (atributos específicos) | Escritura |
| Categorías | Crear una categoría | Nombre, descripción, característicasAlmacenamiento | Escritura |
| Categorías | Consultar una categoría | ID, nombre, descripción, característicasAlmacenamiento | Lectura |
| Proveedores | Crear un proveedor | NIT, nombre, dirección, personaContacto, telefonoContacto | Escritura |
| Proveedores | Actualizar un proveedor | ID, información actualizada (atributos específicos) | Escritura |
| Proveedores | Consultar la información de un proveedor | ID, NIT, nombre, dirección, personaContacto, telefonoContacto | Lectura |
| Órdenes de Compra | Crear una orden de compra | Fecha de creación, fecha de entrega, estado, ID\_sucursal, ID\_proveedor, productos (ID, cantidad, precio) | Escritura |
| Órdenes de Compra | Consultar una orden de compra | ID, detalles del encabezado y productos asociados | Lectura |

1. **Análisis de operaciones de lectura y escritura para cada entidad:**

|  |  |  |  |
| --- | --- | --- | --- |
| Entidad | Tipo de operación | Frecuencia | Cantidad estimada |
| Sucursales | Creación/Modificación | 1 vez al mes | 150 |
| Sucursales | Consulta | 1 vez por semana | N/A |
| Bodegas | Creación/Modificación | 1 vez al mes | 900 |
| Bodegas | Consulta | 1 vez por semana | N/A |
| Proveedores | Creación/Modificación | 10 veces al día | 10,000 |
| Proveedores | Consulta | 200 veces al día | N/A |
| Órdenes de Compra | Creación | 200 veces al día | ~200,000 en 3 años |
| Órdenes de Compra | Consulta | 1,000 veces al día | N/A |
| Productos | Modificación | 10,000 veces al día | 20,000 |
| Productos | Consulta | 2,000 veces al día | N/A |

**3.b Descripción de las colecciones y relaciones (NoSQL)**

**a. Lista de entidades y su descripción**

1. Sucursales:

Representa las sucursales de SuperAndes.

Atributos: ID, nombre, tamaño, dirección, teléfono, ID\_ciudad.

1. Ciudades:

Representa las ciudades donde están ubicadas las sucursales.

Atributos: ID, nombre.

1. Bodegas:

Representa los almacenes físicos asociados a una sucursal.

Atributos: ID, nombre, tamaño, capacidad, ID\_sucursal, lista de productos.

1. Productos:

Representa los productos que maneja SuperAndes.

Atributos: ID, nombre, costo en bodega, precio, presentación, cantidadPresentacion, unidadMedida, especificacionesEmpacado, volumenEmpaque, fechaExpiracion, código de barras, nivelReorden, ID\_categoria.

1. Categorías:

Clasificación de los productos.

Atributos: ID, nombre, descripción, característicasAlmacenamiento.

1. Proveedores:

Representa los proveedores que abastecen productos a las sucursales.

Atributos: ID, NIT, nombre, dirección, personaContacto, telefonoContacto.

1. Órdenes de Compra:

Representa las solicitudes de productos realizadas a proveedores.

Atributos: ID, fecha de creación, fecha de entrega, estado, ID\_sucursal, ID\_proveedor, productos (ID, cantidad, precio).

**b. Relaciones entre entidades y cardinalidad**

1. Sucursales ↔ Ciudades:

Relación: 1: Muchos.

Una ciudad puede tener varias sucursales, pero cada sucursal pertenece a una sola ciudad.

1. Sucursales ↔ Bodegas:

Relación: 1: Muchos.

Una sucursal puede tener varias bodegas, pero cada bodega pertenece a una sola sucursal.

1. Categorías ↔ Productos:

Relación: 1: Muchos.

Una categoría puede tener muchos productos, pero cada producto pertenece a una sola categoría.

1. Proveedores ↔ Productos:

Relación: Muchos: Muchos.

Un proveedor puede abastecer muchos productos, y un producto puede ser abastecido por muchos proveedores.

1. Órdenes de Compra ↔ Proveedores:

Relación: Muchos: 1.

Una orden de compra está asociada a un proveedor, pero un proveedor puede tener muchas órdenes de compra.

1. Órdenes de Compra ↔ Productos:

Relación: Muchos: Muchos.

Una orden puede incluir varios productos, y un producto puede estar en varias órdenes de compra.

1. Bodegas ↔ Productos

Relación: 1: Muchos

Muchos productos pueden hacer parte de una misma bodega, pero un producto no puede pertenecer a distintas bodegas al tiempo.

**c. Tabla de análisis del esquema de asociación**

|  |  |  |
| --- | --- | --- |
| Relación | Esquema (Embebido/Referenciado) | Justificación |
| Sucursales ↔ Ciudades | Referenciado | Las ciudades tienen un número reducido de registros y se comparten entre sucursales. |
| Sucursales ↔ Bodegas | Referenciado | Una sucursal tiene pocas bodegas, no es necesario contener toda la información de cada bodega que hagan parte de una sucursal. |
| Categorías ↔ Productos | Referenciado | Las categorías agrupan muchos productos, y las consultas generalmente incluyen varios productos. |
| Proveedores ↔ Productos | Referenciado | La relación M:M es más eficiente referenciada para evitar duplicación masiva de datos. |
| Órdenes de Compra ↔ Proveedores | Referenciado | Las órdenes se consultan por separado, y los proveedores tienen un alcance global. |
| Órdenes de Compra ↔ Productos | Embebido | Los productos son una parte intrínseca de la orden y suelen consultarse junto con esta. |
| Bodegas ↔ Productos | Embebido | Es importante contar con la información completa de los productos almacenado en cada bodega. |

**3.d Creación de las Colecciones**

* Ciudad

db.ciudad.insertMany([

{ nombre: "Bogotá" },

{ nombre: "Medellín" }

])

* Sucursal

db.sucursal.insertMany([

{

nombre: "Sucursal Norte",

tamaño: 500,

direccion: "Calle 123 #45-67",

telefono: "3201234567",

ciudadId: ObjectId("64fffa6c5d9c1b6f3f0c59a7")

},

{

nombre: "Sucursal Sur",

tamaño: 400,

direccion: "Carrera 89 #10-25",

telefono: "3106549876",

ciudadId: ObjectId("675218d589276807a3e8d224")

}

])

* Bodega

db.bodega.insertMany([

{

nombre: "Bodega Central",

tamaño: 1000,

sucursalId: ObjectId("6752190089276807a3e8d226")

},

{

nombre: "Bodega Sur",

tamaño: 600,

sucursalId: ObjectId("6752190089276807a3e8d226")

}

])

* Categoría

db.categoria.insertMany([

{

nombre: "Alimentos",

descripcion: "Productos comestibles",

caracteristicasAlmacenamiento: "Temperatura controlada"

},

{

nombre: "Electrónica",

descripcion: "Dispositivos electrónicos",

caracteristicasAlmacenamiento: "Ambiente seco"

}

])

* Producto

db.producto.insertMany([

{

nombre: "Producto A",

costoBodega: 5000,

precio: 10000,

presentacion: "Caja",

cantidadPresentacion: 10,

unidadMedida: "kg",

especificacionesEmpacado: "Caja de cartón",

volumenEmpaque: 2000,

fechaExpiracion: new Date("2025-12-31"),

codigoBarras: "1234567890123",

nivelReorden: 5,

categoriaId: ObjectId("6752193689276807a3e8d22a")

}

])

* Proveedor

db.proveedor.insertMany([

{

nit: "123456789",

nombre: "Proveedor A",

direccion: "Calle 50 #10-20",

personaContacto: "Juan Pérez",

telefonoContacto: "3001234567"

},

{

nit: "987654321",

nombre: "Proveedor B",

direccion: "Carrera 80 #20-30",

personaContacto: "Ana Gómez",

telefonoContacto: "3107654321"

}

])

* Ordenes Compra

db.ordenesCompra.insertMany([

{

\_id: ObjectId("6752207f89276807a3e8d234"),

fechaCreacion: new Date("2024-01-01"),

fechaEntrega: new Date("2024-01-15"),

estado: "Pendiente",

idSucursal: ObjectId("675218d589276807a3e8d224"),

idProveedor: ObjectId("67521a6189276807a3e8d230"),

productos: [

{ productoId: ObjectId("6752195589276807a3e8d22c"), cantidad: 50 },

{ productoId: ObjectId("6752195589276807a3e8d22c"), cantidad: 30 }

]

},

{

\_id: ObjectId("6752207f89276807a3e8d235"),

fechaCreacion: new Date("2024-02-01"),

fechaEntrega: new Date("2024-02-10"),

estado: "Entregado",

idSucursal: ObjectId("675218d589276807a3e8d224"),

idProveedor: ObjectId("67521a6189276807a3e8d230"),

productos: [

{ productoId: ObjectId("6752195589276807a3e8d22c"), cantidad: 20 },

{ productoId: ObjectId("6752195589276807a3e8d22c"), cantidad: 40 }

]

}

])

**3. d Creación de los esquemas de validación**

* **Ciudades**

db.runCommand({

collMod: "ciudades",

validator: {

$jsonSchema: {

bsonType: "object",

required: ["nombre"],

properties: {

nombre: {

bsonType: "string",

description: "Debe ser una cadena de texto y es obligatorio",

minLength: 3

}

}

}

},

validationAction: "warn" })

* **Sucursales**

db.runCommand({

collMod: "sucursal",

validator: {

$jsonSchema: {

bsonType: "object",

required: ["nombre", "tamaño", "direccion", "telefono"],

properties: {

nombre: {

bsonType: "string",

description: "El nombre debe ser una cadena de texto y es obligatorio"

},

tamaño: {

bsonType: "int",

minimum: 1,

description: "El tamaño debe ser un número entero mayor que 0"

},

direccion: {

bsonType: "string",

description: "La dirección debe ser una cadena de texto y es obligatorio"

},

telefono: {

bsonType: "string",

pattern: "^[0-9]{10}$",

description: "El teléfono debe ser una cadena de texto con 10 dígitos numéricos"

},

ciudadId: {

}

}

}

},

validationAction: "warn"

})

* **Bodega**

db.runCommand({

collMod: "bodega",

validator: {

$jsonSchema: {

bsonType: "object",

required: ["nombre", "tamaño", "sucursalId"],

properties: {

nombre: {

bsonType: "string",

description: "El nombre debe ser una cadena de texto y es obligatorio"

},

tamaño: {

bsonType: "int",

minimum: 1,

description: "El tamaño debe ser un número entero mayor que 0"

},

sucursalId: {

bsonType: "objectId",

description: "El campo sucursalId debe ser un ObjectId y es obligatorio"

}

}

}

},

validationAction: "warn"

})

* **Categoría**

db.runCommand({

collMod: "categoria",

validator: {

$jsonSchema: {

bsonType: "object",

required: ["nombre", "descripcion", "caracteristicasAlmacenamiento"],

properties: {

nombre: {

bsonType: "string",

description: "El nombre debe ser una cadena de texto y es obligatorio"

},

descripcion: {

bsonType: "string",

description: "La descripción debe ser una cadena de texto y es obligatorio"

},

caracteristicasAlmacenamiento: {

bsonType: "string",

description: "Las características de almacenamiento deben ser una cadena de texto y son obligatorias"

}

}

}

},

validationAction: "warn"

})

* **Producto:**

db.runCommand({

collMod: "producto",

validator: {

$jsonSchema: {

bsonType: "object",

required: ["nombre", "costoBodega", "precio", "presentacion", "cantidadPresentacion", "unidadMedida", "especificacionesEmpacado", "volumenEmpaque", "fechaExpiracion", "codigoBarras", "nivelReorden", "categoriaId"],

properties: {

nombre: {

bsonType: "string",

description: "El nombre debe ser una cadena de texto y es obligatorio"

},

costoBodega: {

bsonType: "int",

minimum: 0,

description: "El costoBodega debe ser un número entero y mayor o igual a 0"

},

precio: {

bsonType: "int",

minimum: 0,

description: "El precio debe ser un número entero y mayor o igual a 0"

},

presentacion: {

bsonType: "string",

description: "La presentación debe ser una cadena de texto y es obligatoria"

},

cantidadPresentacion: {

bsonType: "int",

minimum: 1,

description: "La cantidad de presentación debe ser un número entero mayor que 0"

},

unidadMedida: {

bsonType: "string",

description: "La unidad de medida debe ser una cadena de texto y es obligatoria"

},

especificacionesEmpacado: {

bsonType: "string",

description: "Las especificaciones de empacado deben ser una cadena de texto y son obligatorias"

},

volumenEmpaque: {

bsonType: "int",

minimum: 0,

description: "El volumen de empaque debe ser un número entero mayor o igual a 0"

},

fechaExpiracion: {

bsonType: "date",

description: "La fecha de expiración debe ser una fecha válida y es obligatoria"

},

codigoBarras: {

bsonType: "string",

description: "El código de barras debe ser una cadena de texto y es obligatorio"

},

nivelReorden: {

bsonType: "int",

minimum: 1,

description: "El nivel de reorden debe ser un número entero mayor o igual a 1"

},

categoriaId: {

bsonType: "objectId",

description: "El campo categoriaId debe ser un ObjectId y es obligatorio"

}

}

}

},

validationAction: "warn"

})

* **Proveedor**

db.runCommand({

collMod: "proveedor",

validator: {

$jsonSchema: {

bsonType: "object",

required: ["nit", "nombre", "direccion", "personaContacto", "telefonoContacto"],

properties: {

nit: {

bsonType: "string",

description: "El NIT debe ser una cadena de texto y es obligatorio"

},

nombre: {

bsonType: "string",

description: "El nombre debe ser una cadena de texto y es obligatorio"

},

direccion: {

bsonType: "string",

description: "La dirección debe ser una cadena de texto y es obligatoria"

},

personaContacto: {

bsonType: "string",

description: "La persona de contacto debe ser una cadena de texto y es obligatoria"

},

telefonoContacto: {

bsonType: "string",

pattern: "^[0-9]{10}$",

description: "El teléfono de contacto debe ser una cadena de texto con 10 dígitos"

}

}

}

},

validationAction: "warn"

})

* **Ordenes Compra:**

db.runCommand({

collMod: "ordenesCompra",

validator: {

$jsonSchema: {

bsonType: "object",

required: ["fechaCreacion", "fechaEntrega", "estado", "idSucursal", "idProveedor", "productos"],

properties: {

fechaCreacion: {

bsonType: "date",

description: "La fecha de creación debe ser una fecha válida y es obligatoria"

},

fechaEntrega: {

bsonType: "date",

description: "La fecha de entrega debe ser una fecha válida y es obligatoria"

},

estado: {

bsonType: "string",

description: "El estado debe ser una cadena de texto y es obligatorio"

},

idSucursal: {

bsonType: "objectId",

description: "El campo idSucursal debe ser un ObjectId y es obligatorio"

},

idProveedor: {

bsonType: "objectId",

description: "El campo idProveedor debe ser un ObjectId y es obligatorio"

},

productos: {

bsonType: "array",

items: {

bsonType: "object",

required: ["productoId", "cantidad"],

properties: {

productoId: {

bsonType: "objectId",

description: "El campo productoId debe ser un ObjectId y es obligatorio"

},

cantidad: {

bsonType: "int",

minimum: 1,

description: "La cantidad debe ser un número entero mayor que 0"

}

}

},

description: "El campo productos debe ser un arreglo de objetos con los campos productoId y cantidad"

}

}

}

},

validationAction: "warn"

})