Entrega 3 — Sistemas Transaccionales
Juan Esteban Ríos González — 202221404
Alejandra Fruto Osorio — 202311287
Santiago Quevedo — 202312920

Universidad de los Andes

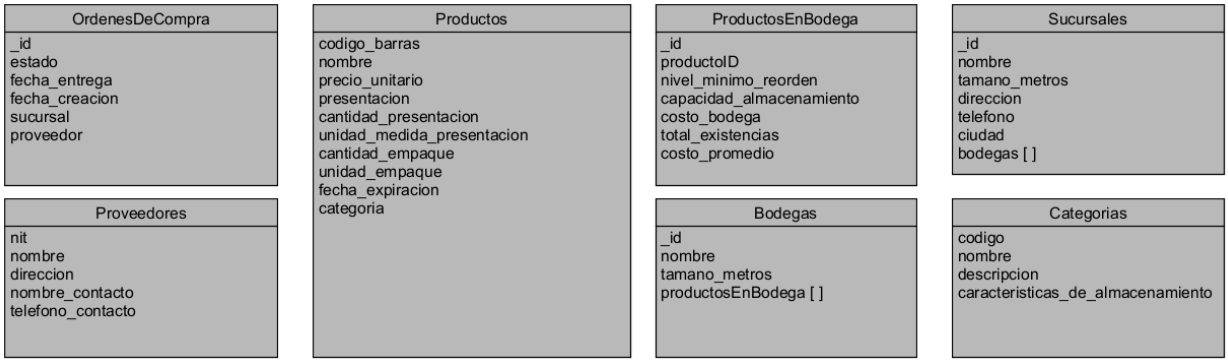
Proyecto - Entrega 3

PUNTO UNO: Elementos fundamentales del negocio

Los elementos fundamentales para el negocio son las órdenes de compra, los proveedores, los productos, las bodegas, las sucursales y las categorías.

PUNTO DOS: Análisis y modelo conceptual

Figura 1
Modelo conceptual



PUNTO TRES: Diseño de la base de datos

a) Análisis de la carga de trabajo (workload)

a) Entidades y sus atributos

Las entidades que serán utilizadas para el desarrollo de la aplicación son las siguientes: Orden de Compra, Sucursal, Proveedor, Producto, Categoria, Producto en Bodega, Bodega. Estas entidades han sido modeladas como colecciones.

La entidad Orden de Compra tiene los siguientes atributos:

- Id: El respectivo id de la orden de compra.
- Estado: El estado que toma la orden, puede ser vigente, entregada o anulada.
- Fecha_entrega: La fecha de entrega que se ha estipulado para la orden.
- Fecha_creacion: La fecha del día en que se creó la orden de compra
- Sucursal: Aquí se tiene referenciada la sucursal asociada a la orden.
- Proveedor: Aquí se tiene referenciado el proveedor asociado a la orden.

La entidad Sucursal tiene los siguientes atributos:

- Id: El respectivo id de la sucursal.
- Nombre: El nombre de la sucursal.
- Tamano_metros: El tamaño en metros de la sucursal.
- Direccion: La dirección de la sucursal.
- Telefono: El teléfono asociado a la sucursal.
- Ciudad: El nombre de la ciudad donde se encuentra la sucursal.
- Bodegas: Es un array con los ids de las bodegas asociadas a la sucursal. Se tienen referenciadas las bodegas.

La entidad Proveedor tiene los siguientes atributos:

- Nit: El respectivo id del proveedor.
- Nombre: El nombre de la empresa proveedora.
- Direccion: La dirección del proveedor.
- Nombre_contacto: El nombre de la persona de contacto del proveedor.
- Telefono_contacto: El teléfono de la persona de contacto del proveedor.

La entidad Producto tiene los siguientes atributos:

- Codigo_barras: El respectivo id del producto.
- Nombre: El nombre del producto.
- Precio_unitario: El precio de venta del producto.
- Presentacion: La forma en la que se presenta el producto.
- Cantidad_Presentacion: La cantidad de elementos que vienen en esa forma de presentación.
- Unidad_medida_presentacion: La unidad de medida del tipo de presentación del producto.
- Cantidad_empaque: La cantidad de elementos que vienen en el empaque de ese producto.
- Unidad_empaque: La unidad de medida de los elementos del empaque.
- Fecha_expiracion: La fecha de vencimiento del producto.
- Categoria: Aquí se tiene referenciada la categoría del producto.

La entidad Categoria tiene los siguientes atributos:

- Codigo: El respectivo id de la categoría.
- Nombre: El nombre de la categoría.
- Descripcion: La descripción de la categoría.
- Caracteristicas_de_almacenaimiento: Las características necesarias para que se almacenen correctamente los productos asociados a la categoría.

La entidad Bodega tiene los siguientes atributos:

- Id: El respectivo id de la bodega.
- Nombre: El nombre de la bodega.
- Tamano_metros: El tamaño en metros de la bodega.
- ProductosEnBodega: Un array con los ids de los productos que se encuentran en esa bodega (estos ids son de la colección productos en bodega y no de la colección productos). Se tienen referenciados los productos en bodega.

La entidad Productos en Bodega modela la relación entre la entidad Producto y la entidad Bodega y tiene los siguientes atributos:

- Id: El respectivo id de la relación.
- ProductoId: El id del producto asociado, aquí se encuentra el producto referenciado.
- Nivel_minimo_reorden: La cantidad minima de unidades que se deben tener en la bodega.
- Capacidad_almacenamiento: La capacidad de la bodega para almacenar este producto.
- Costo_bodega: El costo del producto en esa bodega.
- Total_existencias: El total de productos actuales en esa bodega.
- Costo_promedio: El costo promedio del producto.

b) Cuantificación de entidades

Para cada entidad se describe la cantidad estimada de registros. Estas aproximaciones están basadas en la información brindada en la sección “Carga de trabajo de la aplicación SuperAndes”. Y los registros están estimados considerando una cadena de almacenes de tamaño medio.

- Ordenes de Compra: 70.000 registros al año. En un período de 3 años se aproximan 200.000 registros.
- Sucursales: 150
- Proveedores: 10.000
- Productos: 20.000
- Categorias: 100
- Bodegas: 900
- Productos en Bodega: 20.000

c) Análisis de las operaciones de lectura y escritura

Para cada entidad se mostrará cuáles son las operaciones que se realizaran con ella y cuál es la información necesitada. A partir de esto, se analizará si es una operación de lectura o escritura. El análisis se encuentra en la siguiente tabla.

Entidades	Operaciones	Información necesitada	Tipo
Sucursal	Añadir sucursal	Detalles de la sucursal	Write
Sucursal	Buscar/Consultar sucursal	Id de la sucursal	Read
Bodega	Añadir bodega	Detalles de la bodega	Write
Bodega	Eliminar bodega	Id o nombre de la bodega	Write
Bodega	Buscar/Consultar bodega	Id de la bodega	Read
Proveedor	Añadir proveedor	Detalles del proveedor	Write
Proveedor	Modificar proveedor	Id del proveedor + detalles a modificar	Write
Proveedor	Buscar/Consultar proveedor	Id del proveedor	Read
Categoría	Añadir categoria	Detalles de la categoría	Write
Categoría	Buscar categoria	Id o nombre de la categoría	Read
Producto	Añadir producto	Detalles del producto + id categoria referenciada	Write
Producto	Buscar producto	Código o nombre del producto	Read
Producto	Modificar producto	Código del producto + detalles a modificar	Write
Producto en bodega	Añadir producto en bodega	Detalles del producto + id del producto asociado + id de la bodega	Write
Producto en bodega	Modificar producto en bodega	Id del producto en bodega	Write
Producto en bodega	Consultar información asociada de un producto en una	Id del producto en bodega	Read

	bodega		
Orden de Compra	Añadir orden	Detalles de la orden + id de la sucursal referenciada + id del proveedor referenciado	Write
Orden de Compra	Modificar estado de la orden	Id orden	Write
Orden de Compra	Buscar orden	Id orden	Read
Producto	Buscar productos que cumplen con cierta característica	Detalles de los productos + características necesitadas	Read
Producto, Categoria	Buscar productos dentro de esa categoría	Detalles de los productos + detalles de la categoría	Read
Producto, Sucursal	Buscar productos disponibles en esa sucursal	Detalles de los productos + detalles de la sucursal	Read
Producto, Bodega, Sucursal	Buscar productos disponibles en cada bodega de una sucursal	Detalles de los productos + Id sucursal + ids y nombres bodegas	Read

d) Cuantificación de las operaciones de lectura y escritura

Para cada entidad se mostrará la cuantificación de las operaciones de lectura y escritura. Esta cuantificación está basada en la información brindada en la sección “Carga de trabajo de la aplicación SuperAndes”.

Entidades	Operaciones	Información necesitada	Tipo	Tasa
Sucursal	Añadir sucursal	Detalles de la sucursal	Write	12/año
Sucursal	Buscar/Consultar sucursal	Id de la sucursal	Read	52/año
Bodega	Añadir bodega	Detalles de la bodega	Write	12/año
Bodega	Eliminar bodega	Id o nombre de la bodega	Write	1/año

Bodega	Buscar/Consultar bodega	Id de la bodega	Read	52/año
Proveedor	Añadir proveedor	Detalles del proveedor	Write	10/dia
Proveedor	Modificar proveedor	Id del proveedor + detalles a modificar	Write	10/dia
Proveedor	Buscar/Consultar proveedor	Id del proveedor	Read	200/dia
Categoría	Añadir categoria	Detalles de la categoría	Write	100/año
Categoría	Buscar categoria	Id o nombre de la categoría	Read	30/dia
Producto	Añadir producto al inventario	Detalles del producto + id de la categoría referenciada	Write	10000/dia
Producto	Buscar producto	Código o nombre del producto	Read	2000/dia
Producto	Modificar producto	Código del producto + detalles a modificar	Write	10000/dia
Producto en bodega	Añadir producto en bodega	Detalles del producto + id del producto asociado + id de la bodega	Write	10000/dia
Producto en bodega	Modificar producto en bodega	Id del producto en bodega	Write	10000/dia
Producto en bodega	Consultar información asociada de un producto en una bodega	Id del producto en bodega	Read	2000/dia
Orden de Compra	Añadir orden	Detalles de la orden + id de la sucursal referenciada + id del proveedor referenciado	Write	200/dia
Orden de Compra	Modificar estado de la orden	Id orden	Write	200/dia

Orden de Compra	Buscar orden	Id orden	Read	1000/dia
Producto	Buscar productos que cumplen con cierta característica	Detalles de los productos + características necesitadas	Read	60/dia
Producto, Categoria	Buscar productos dentro de esa categoría	Detalles de los productos + detalles de la categoría	Read	60/dia
Producto, Sucursal	Buscar productos disponibles en esa sucursal	Detalles de los productos + detalles de la sucursal	Read	60/dia
Producto, Bodega, Sucursal	Buscar productos disponibles en cada bodega de una sucursal	Detalles de los productos + Id sucursal + ids y nombres bodegas	Read	10/dia

b) Colecciones de datos y las relaciones entre ellas (NoSQL)

a) Lista de entidades con su descripción

- **Sucursal:** La sucursal es una entidad que ha sido modelada como colección. SuperAndes tiene múltiples sucursales. Una sucursal puede tener una o varias bodegas pero solo una ciudad. En la colección Sucursales las bodegas se encuentran referenciadas.
- **Bodega:** Las bodegas son donde se almacenan los productos. SuperAndes tiene múltiples bodegas cada bodega tiene una sucursal asociada. En la colección Bodegas los productos que se encuentran en esa bodega están referenciados.
- **Proveedor:** Los proveedores son los que llevan los productos a las bodegas. SuperAndes tiene varios proveedores, cada producto tiene varios proveedores y cada proveedor puede ofrecer varios productos.
- **Producto:** Los productos son los elementos que generan las ventas en SuperAndes. Un producto tiene una información extra asociada a la bodega en la que se encuentra (modelado como otra colección) y tienen una categoría, la cuál está referenciada.
- **Categoría**: Las categorías clasifican a los productos dependiendo de ciertas características. Una categoría puede tener varios productos.
- **Orden de Compra:** Las órdenes de compra son los elementos que llevan el registro de los productos comprados por SuperAndes y distribuidos por los

proveedores. Una orden de compra tiene una sucursal y un proveedor. Ambos se encuentran referenciados dentro de la orden.

- **Producto en Bodega:** Esta es una colección que modela la relación entre la colección Bodega y la colección Producto. Aquí se tiene información extra de los productos que se encuentran en una bodega. El producto asociado se encuentra referenciado.

b) Relaciones entre entidades y su cardinalidad

A continuación se describirán las relaciones entra las entidades y que tipo de cardinalidad poseen.

- **OrdenesDeCompra - Sucursal:** Esta relación está modelada en ordenes de compra, puesto que la sucursal asociada a la orden de compra se encuentra referenciada en esta. Esta es una relación de uno a muchos debido a que una orden de compra solo tiene una sucursal pero una sucursal tiene muchas ordenes de compra.
    - Esquema de asociación: Referenciado o Normalizado.
    - Cardinalidad: Uno a muchos.

- **OrdenesDeCompra - Proveedor:** Esta relación está modelada en ordenes de compra, puesta que el proveedor asociado a la orden de compra se encuentra referenciado en esta. Esta es una relación de uno a muchos debido a que un proveedor puede atender múltiples ordenes de compra pero una orden de compra solo es distribuida por un proveedor.
    - Esquema de asociación: Referenciado.
    - Cardinalidad: Uno a muchos.

- **Productos - Categoria:** Esta relación esta modelada por la colección productos, ya que la categoría del producto se encuentra referenciada dentro de esta. Esta es una relación de uno a muchos puesto que un producto solo puede pertenecer a una categoría pero una categoría puede clasificar muchos productos.
    - Esquema de asociación: Referenciado.
    - Cardinalidad: Uno a muchos.

- **Bodegas - ProductosEnBodega:** Esta relación esta modelada por la colección bodegas debido a que los productos se encuentran referenciados dentro de la colección en un Array. La relación es de muchos a muchos puesto que una bodega puede tener muchos productos almacenados y un producto se puede encontrar en múltiples bodegas.
    - Esquema de asociación: Referenciado.

    - Cardinalidad: Muchos a muchos.

- **Sucursales - Bodegas:** La relación está modelada por la colección sucursales ya que las bodegas están referenciadas como un Array de id's dentro de la colección. La cardinalidad es de uno a muchos. Una bodega solo puede pertenecer a una sucursal pero una sucursal puede tener muchas bodegas.
    - Esquema de asociación: Referenciado.
    - Cardinalidad: Uno a muchos.

- **ProductosEnBodega - Producto:** Esta relación contiene información extra sobre los productos que se encuentran en bodegas. Está modela por la colección ProductosEnBodega ya que el id del producto asociado se encuentra referenciado en esta colección. La cardinalidad es de uno a uno puesto que por cada producto hay producto en bodega y viceversa.
    - Esquema de asociación: Referenciado.
    - Cardinalidad: Uno a uno.

c) Análisis de la selección de esquema de asociación.

A continuación se muestra la tabla de análisis de esquema de relación entre entidades, en la cual están basados los análisis de la selección de esquemas.

Guideline Name	Question	Embed	Reference
Simplicity	Would keeping the pieces of information together lead to a simpler data model and code?	Yes	No
Go Together	Do the pieces of information have a "has-a," "contains," or similar relationship?	Yes	No
Query Atomicity	Does the application query the pieces of information together?	Yes	No
Update Complexity	Are the pieces of information updated together?	Yes	No
Archival	Should the pieces of information be archived at the same time?	Yes	No
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	No	Yes
Data Duplication	Would data duplication be too complicated to manage and undesired?	No	Yes
Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	No	Yes
Document Growth	Would the embedded piece grow without bound?	No	Yes
Workload	Are the pieces of information written at different times in a write-heavy workload?	No	Yes
Individuality	For the children side of the relationship, can the pieces exist by themselves without a parent?	No	Yes

- **OrdenesDeCompra - Sucursal:** Para esta relación se escogió un esquema de asociación referenciado debido a que se busca la individualidad, es decir que la entidad hijo pueda existir sin necesidad de un padre. También porque permite tanto actualizar las entidades como archivar informaciones sin que sea mandatorio alterar ambas colecciones.

- **OrdenesDeCompra - Proveedor:** El esquema de asociación elegido en esta relación es referenciado o normalizado puesto que le da simplicidad al código, permite la individualidad y reduce el tamaño de los documentos en caso tal de que la información de una de las entidades aumente.

- **Productos - Categoría:** En esta relación el esquema de asociación seleccionado es el normalizado puesto que con la entidad producto se realizan muchas consultas que no usan categoría por lo cuál se cumple con la atomicidad de las consultas. Además de esto, se disminuye la complejidad de las actualizaciones ya que un producto se modifica 20 veces más que una categoría.

- **Bodegas - ProductosEnBodega:** Para esta relación el esquema de asociación escogido es referenciado ya que aquí se presenta una relación de muchos a muchos y si se embebiera alguna de estas entidades en la otra, el tamaño de los documentos sería excesivo y se generaría una mayor duplicación de la información.

- **Sucursales - Bodegas:** El esquema de asociación escogido es referenciado puesto que facilita la individualidad, facilita la atomicidad de las consultas, disminuye la complejidad de las actualizaciones y además hace el código más simple.

- **ProductosEnBodega - Producto:** En esta relación el esquema de asociación seleccionado es normalizado puesto que al ser una relación de cardinalidad uno a uno, se necesita evitar la duplicación de datos, aumentar la simplicidad a la hora de modelar y de codificar. Además reduce la complejidad de actualización.

d) Descripción gráfica de cada relación usando JSON

A continuación se muestra la descripción gráfica en JSON de cada relación en donde se observa el esquema de asociación utilizado.

OrdenesDeCompra - Sucursal (Referenciado):

Documento Orden de Compra

```
_id: 1
sucursal : 0
proveedor : "12345"
estado : "Vigente"
fecha_entrega : "2024-12-01"
fecha_creacion : "2024-11-20"
```

Documento Sucursal

```
_id: 0
tamano_metros : 40
direccion : "OK"
telefono : "3225177456"
ciudad : "Medellin"
bodegas : Array (2)
  0: 1
  1: 2
```

OrdenesDeCompra - Proveedor (Referenciado):

Documento Orden de Compra

```
_id: 1
sucursal : 0
proveedor : "12345"
estado : "Vigente"
fecha_entrega : "2024-12-01"
fecha_creacion : "2024-11-20"
```

Documento Proveedor

```
_id: "12345"
nombre : "Coca-Cola CentroAmerica"
direccion : "Cra 45 #45-78"
nombre_contacto : "Juan"
telefono_contacto : "35447855489"
```

Productos – Categoría (Referenciado):

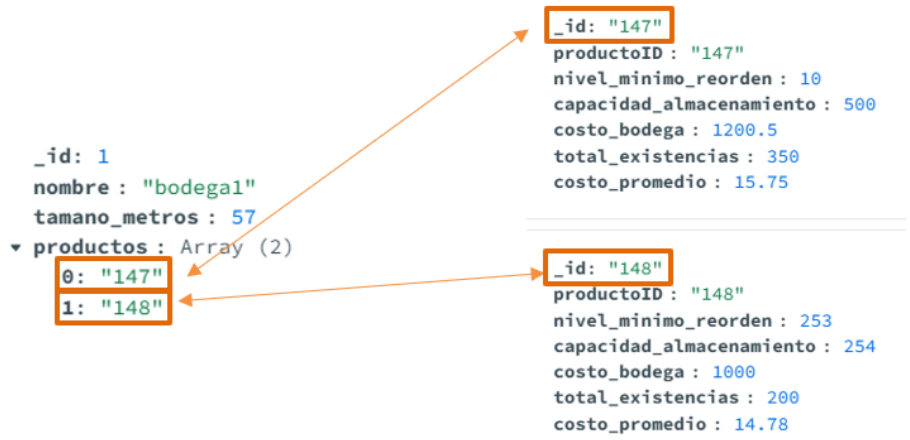
Documento Producto

```
_id: "147"
nombre : "Lechuga Romana"
precio_unitario : 11000
presentacion : "Unidad"
cantidad_presentacion : 1
unidad_medida_presentacion : "Unitario"
cantidad_empaque : 1
unidad_empaque : "unidad"
fecha_expiracion : "2024-11-20"
categoria :"20247"
```

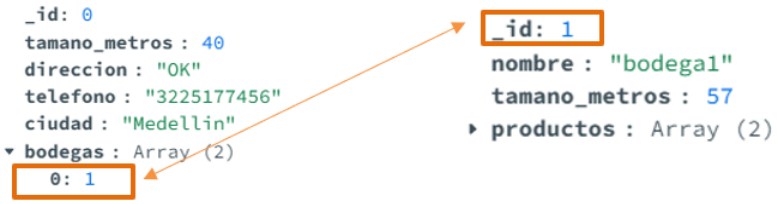
Documento Categoría

```
_id: "20247"
nombre : "Legumbres"
descripcion : "Todo desde..."
caracteristicas_de_almacenamiento : "Lugar seco y fresco"
```

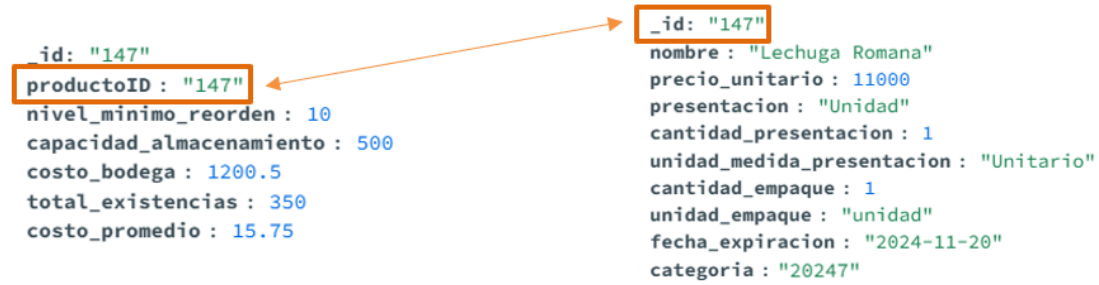
Bodegas – ProductosEnBodega (Referenciado):



Sucursales - Bodegas (Referenciado):



ProductosEnBodega – Producto (Referenciado):

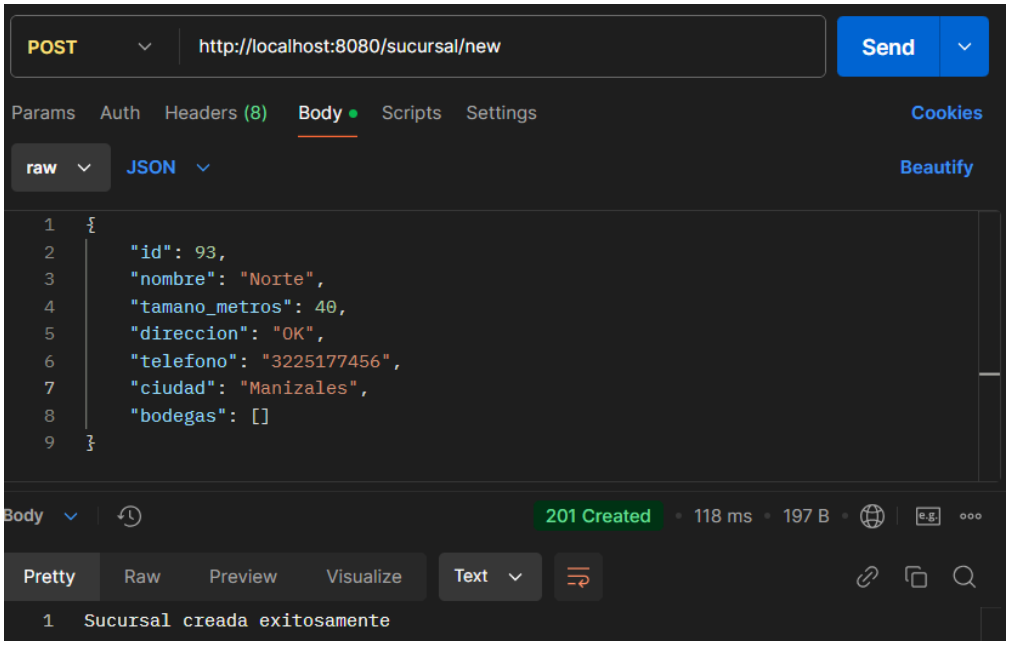


c) El archivo está anexo en el repositorio.

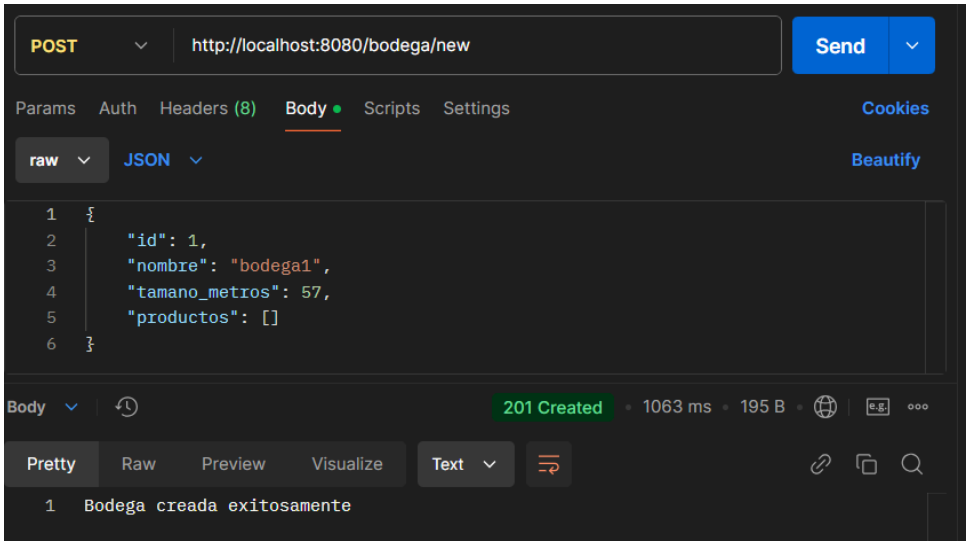
d) El archivo está anexo en el repositorio.

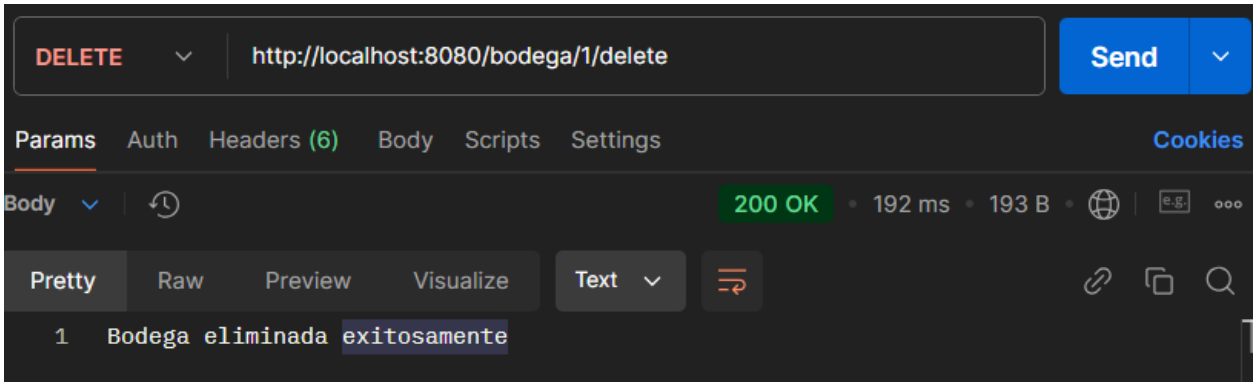
PUNTO CUATRO

RF1

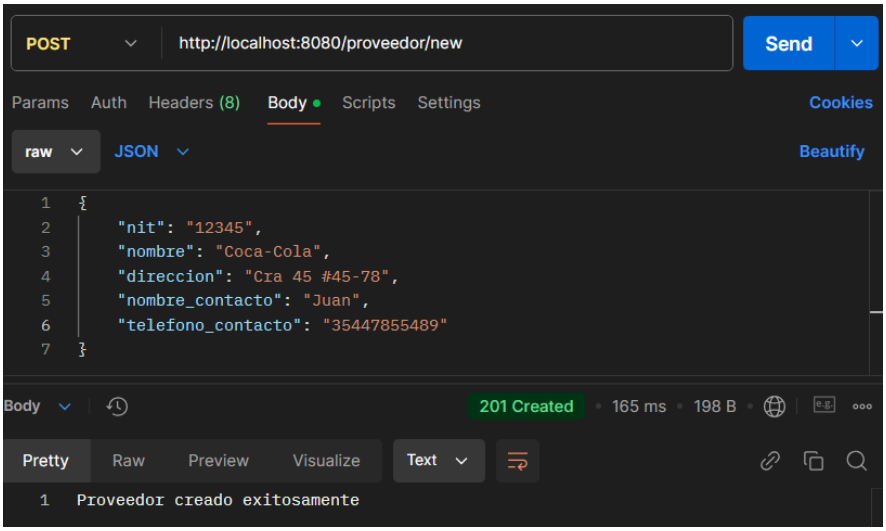


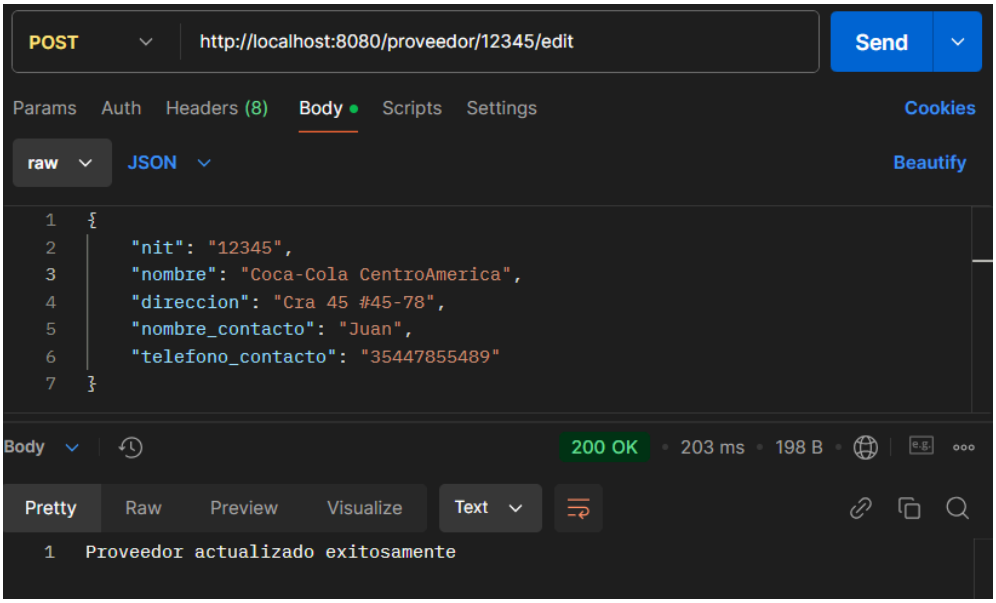
RF2



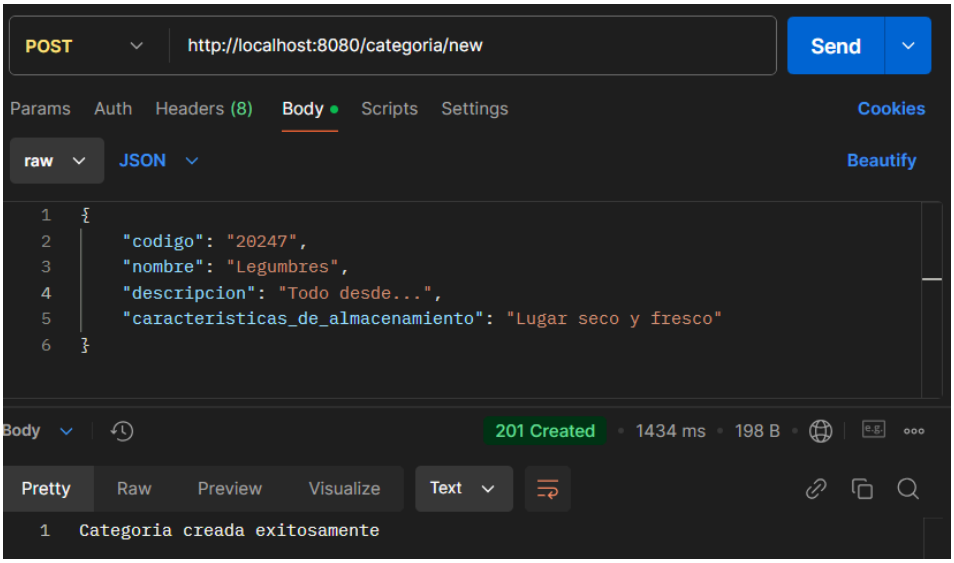

DELETE
http://localhost:8080/bodega/1/delete
Send
Params
Auth
Headers (6)
Body
Scripts
Settings
Cookies
Body
200 OK
192 ms
193 B
Pretty
Raw
Preview
Visualize
Text
1 Bodega eliminada exitosamente

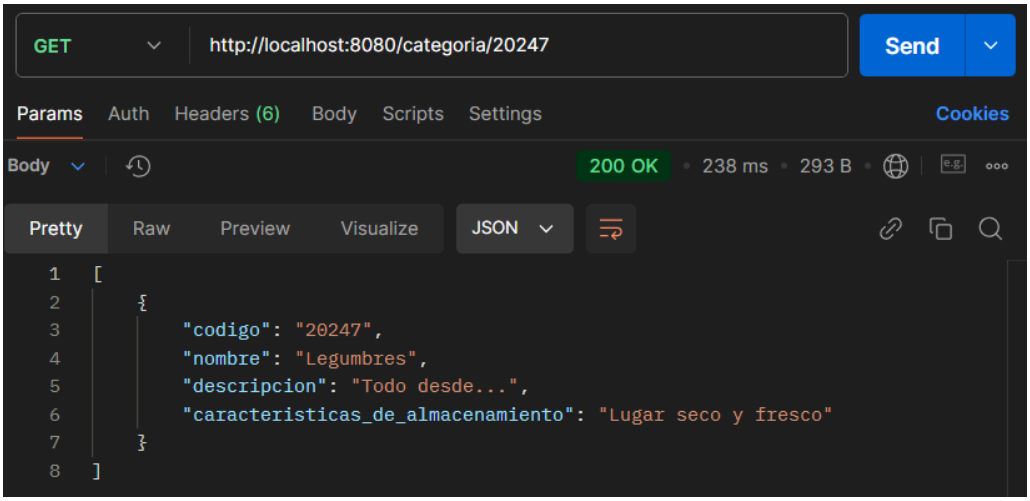
RF3


POST
http://localhost:8080/proveedor/new
Send
Params
Auth
Headers (8)
Body
Scripts
Settings
Cookies
raw
JSON
Beautify
{
"nit": "12345",
"nombre": "Coca-Cola",
"direccion": "Cra 45 #45-78",
"nombre_contacto": "Juan",
"telefono_contacto": "35447855489"
}
Body
201 Created
165 ms
198 B
Pretty
Raw
Preview
Visualize
Text
1 Proveedor creado exitosamente

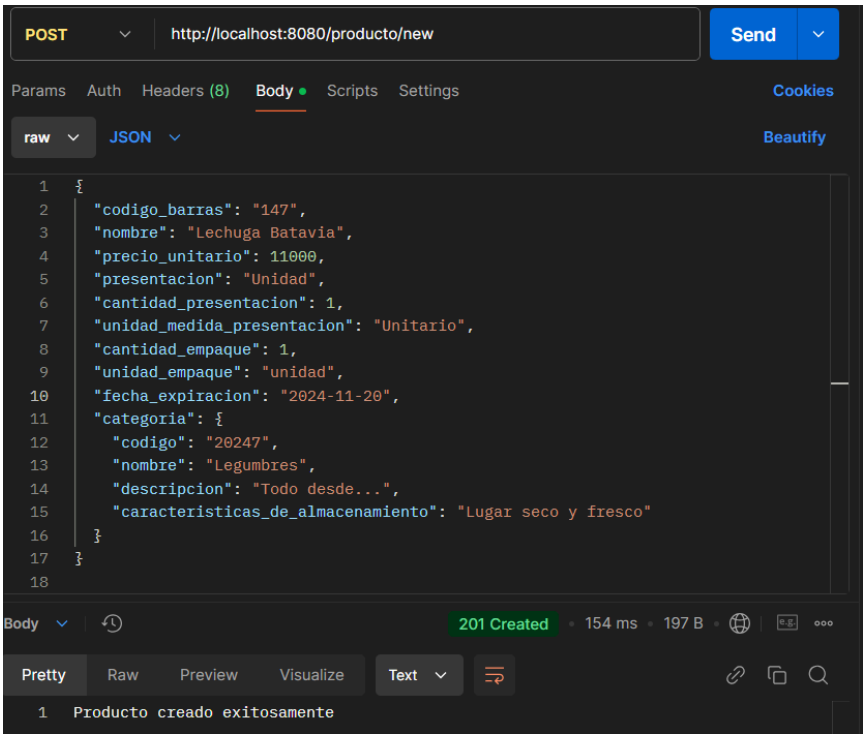

POST
http://localhost:8080/proveedor/12345/edit
Send
Params
Auth
Headers (8)
Body
Scripts
Settings
Cookies
raw
JSON
Beautify
{
"nit": "12345",
"nombre": "Coca-Cola CentroAmerica",
"direccion": "Cra 45 #45-78",
"nombre_contacto": "Juan",
"telefono_contacto": "35447855489"
}
Body
200 OK
203 ms
198 B
Pretty
Raw
Preview
Visualize
Text
1 Proveedor actualizado exitosamente

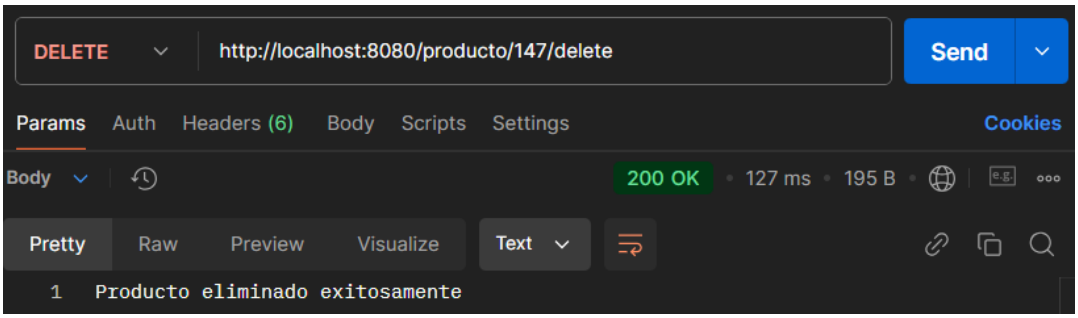
RF4

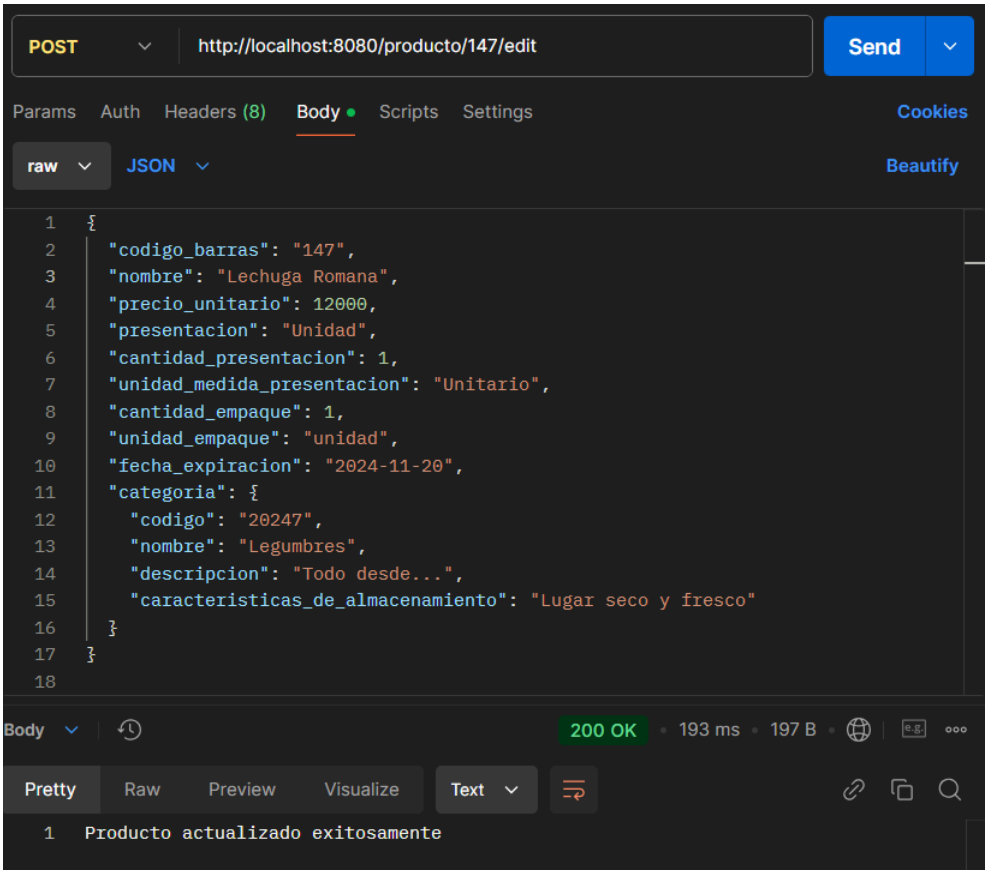

POST
http://localhost:8080/categoria/new
Send
Params Auth Headers (8) Body Scripts Settings
Cookies
raw JSON
Beautify
{
"codigo": "20247",
"nombre": "Legumbres",
"descripcion": "Todo desde...",
"caracteristicas_de_almacenamiento": "Lugar seco y fresco"
}
Body
201 Created
1434 ms
198 B
Pretty Raw Preview Visualize Text
Categoria creada exitosamente

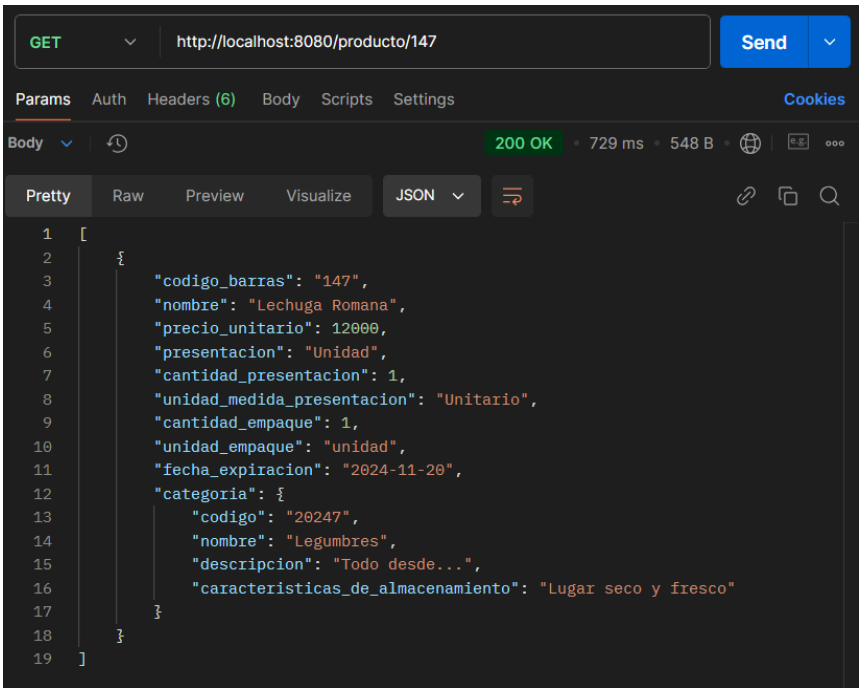

GET
http://localhost:8080/categoria/20247
Send
Params Auth Headers (6) Body Scripts Settings
Cookies
Body
200 OK
238 ms
293 B
Pretty Raw Preview Visualize JSON
[
{
"codigo": "20247",
"nombre": "Legumbres",
"descripcion": "Todo desde...",
"caracteristicas_de_almacenamiento": "Lugar seco y fresco"
}
]

RF5

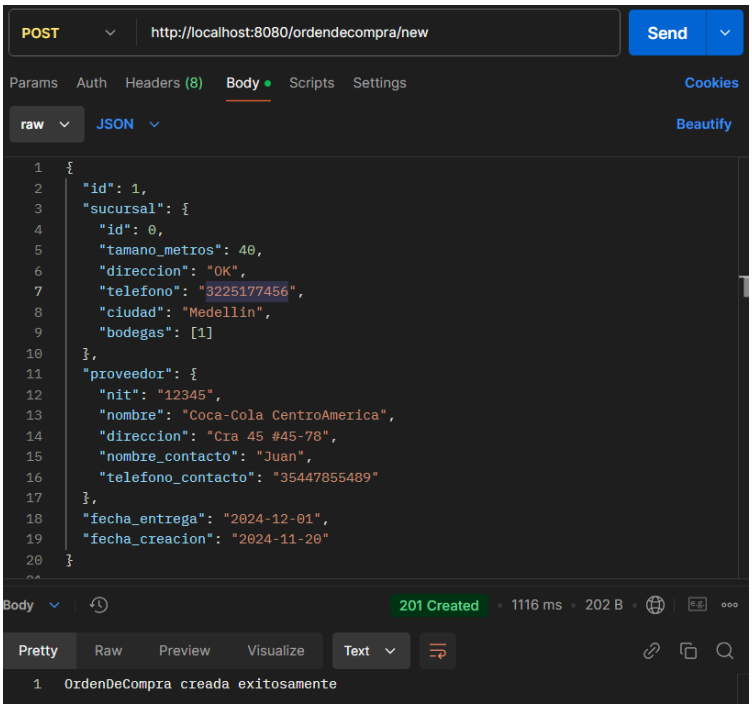

POST
http://localhost:8080/producto/new
Send
Params Auth Headers (8) Body Scripts Settings
Cookies
raw JSON
Beautify
{
"codigo_barras": "147",
"nombre": "Lechuga Batavia",
"precio_unitario": 11000,
"presentacion": "Unidad",
"cantidad_presentacion": 1,
"unidad_medida_presentacion": "Unitario",
"cantidad_empaque": 1,
"unidad_empaque": "unidad",
"fecha_expiracion": "2024-11-20",
"categoria": {
"codigo": "20247",
"nombre": "Legumbres",
"descripcion": "Todo desde...",
"caracteristicas_de_almacenamiento": "Lugar seco y fresco"
}
}
Body
201 Created
154 ms
197 B
Pretty Raw Preview Visualize
Text
1 Producto creado exitosamente


DELETE
http://localhost:8080/producto/147/delete
Send
Params Auth Headers (6) Body Scripts Settings
Cookies
Body
200 OK
127 ms
195 B
Pretty Raw Preview Visualize
Text
1 Producto eliminado exitosamente

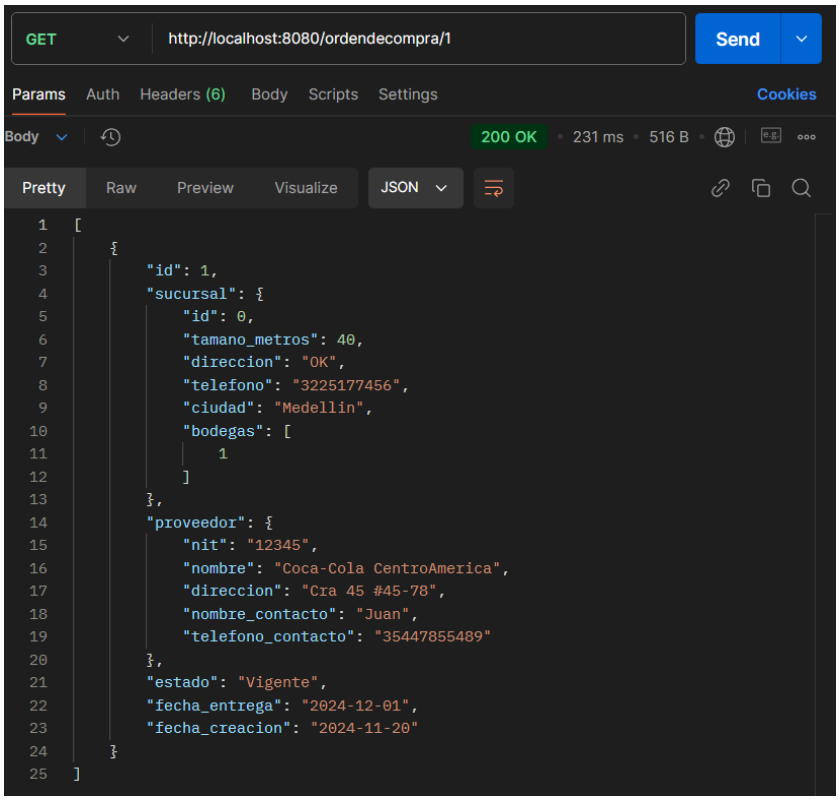

POST
http://localhost:8080/producto/147/edit
Send
Params Auth Headers (8) Body Scripts Settings
Cookies
raw JSON
Beautify
{
"codigo_barras": "147",
"nombre": "Lechuga Romana",
"precio_unitario": 12000,
"presentacion": "Unidad",
"cantidad_presentacion": 1,
"unidad_medida_presentacion": "Unitario",
"cantidad_empaque": 1,
"unidad_empaque": "unidad",
"fecha_expiracion": "2024-11-20",
"categoria": {
"codigo": "20247",
"nombre": "Legumbres",
"descripcion": "Todo desde...",
"caracteristicas_de_almacenamiento": "Lugar seco y fresco"
}
}
Body
200 OK
193 ms
197 B
Pretty Raw Preview Visualize Text
Producto actualizado exitosamente


GET
http://localhost:8080/producto/147
Send
Params Auth Headers (6) Body Scripts Settings
Cookies
Body
200 OK
729 ms
548 B
Pretty Raw Preview Visualize JSON
[
{
"codigo_barras": "147",
"nombre": "Lechuga Romana",
"precio_unitario": 12000,
"presentacion": "Unidad",
"cantidad_presentacion": 1,
"unidad_medida_presentacion": "Unitario",
"cantidad_empaque": 1,
"unidad_empaque": "unidad",
"fecha_expiracion": "2024-11-20",
"categoria": {
"codigo": "20247",
"nombre": "Legumbres",
"descripcion": "Todo desde...",
"caracteristicas_de_almacenamiento": "Lugar seco y fresco"
}
}
]

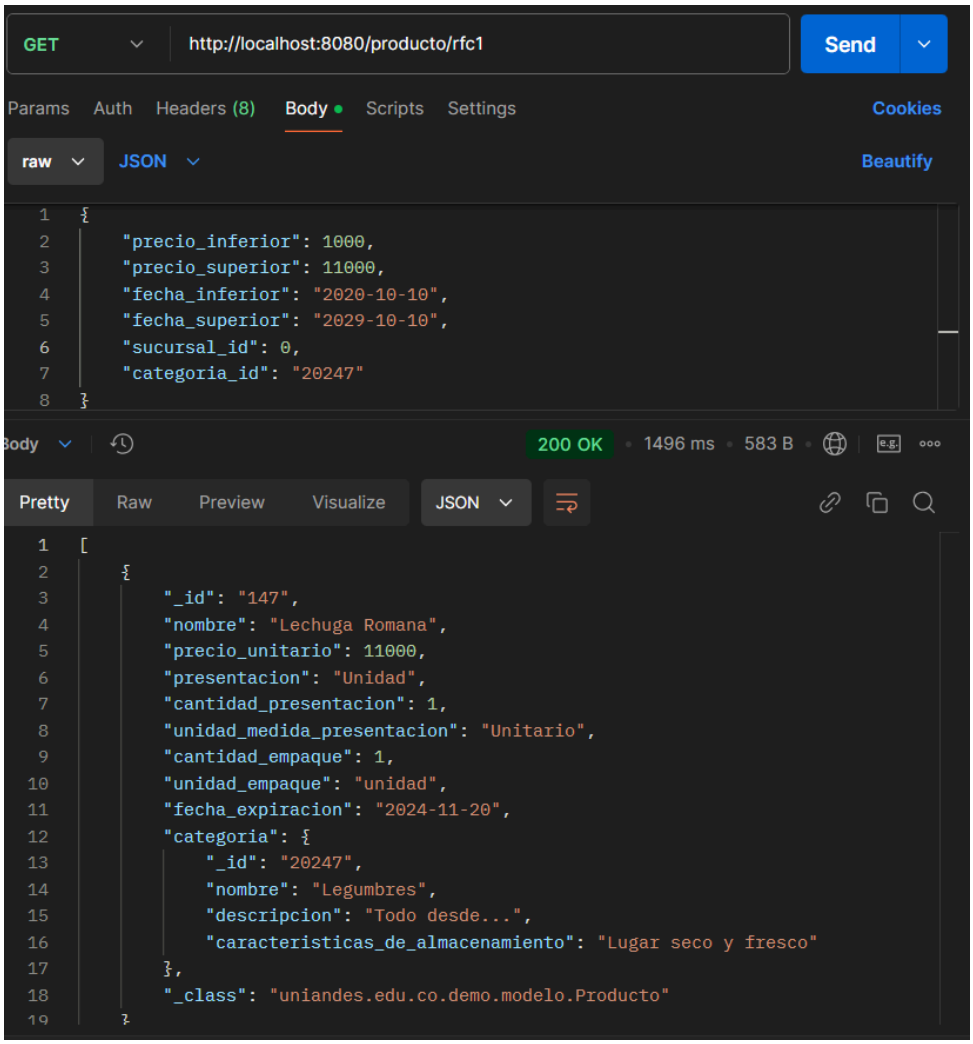
RF6


POST
http://localhost:8080/ordendecompra/new
Send
Params Auth Headers (8) Body Scripts Settings
Cookies
raw JSON
Beautify
{
"id": 1,
"sucursal": {
"id": 0,
"tamano_metros": 40,
"direccion": "OK",
"telefono": "3225177456",
"ciudad": "Medellin",
"bodegas": [1]
},
"proveedor": {
"nit": "12345",
"nombre": "Coca-Cola CentroAmerica",
"direccion": "Cra 45 #45-78",
"nombre_contacto": "Juan",
"telefono_contacto": "35447855489"
},
"fecha_entrega": "2024-12-01",
"fecha_creacion": "2024-11-20"
}
Body
201 Created
1116 ms
202 B
Pretty Raw Preview Visualize Text
OrdenDeCompra creada exitosamente

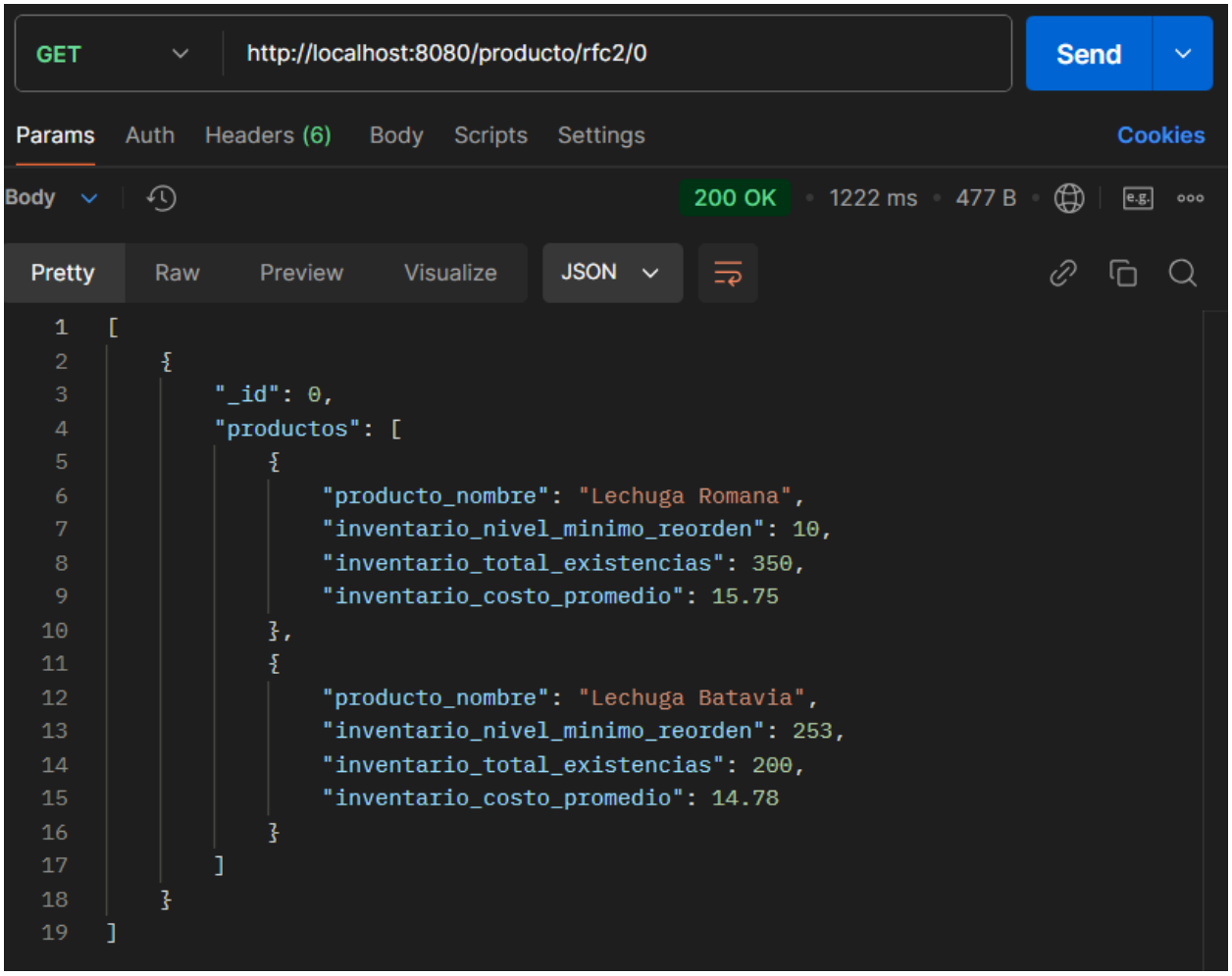
RF7


GET
http://localhost:8080/ordendecompra/1
Send
Params Auth Headers (6) Body Scripts Settings
Cookies
Body
200 OK
231 ms
516 B
Pretty Raw Preview Visualize JSON
[
{
"id": 1,
"sucursal": {
"id": 0,
"tamano_metros": 40,
"direccion": "OK",
"telefono": "3225177456",
"ciudad": "Medellin",
"bodegas": [
1
]
},
"proveedor": {
"nit": "12345",
"nombre": "Coca-Cola CentroAmerica",
"direccion": "Cra 45 #45-78",
"nombre_contacto": "Juan",
"telefono_contacto": "35447855489"
},
"estado": "Vigente",
"fecha_entrega": "2024-12-01",
"fecha_creacion": "2024-11-20"
}
]

RFC1


GET
http://localhost:8080/producto/rfc1
Send
Params
Auth
Headers (8)
Body
Scripts
Settings
Cookies
raw
JSON
Beautify
{
"precio_inferior": 1000,
"precio_superior": 11000,
"fecha_inferior": "2020-10-10",
"fecha_superior": "2029-10-10",
"sucursal_id": 0,
"categoria_id": "20247"
}
200 OK
1496 ms
583 B
Pretty
Raw
Preview
Visualize
JSON
[
{
"_id": "147",
"nombre": "Lechuga Romana",
"precio_unitario": 11000,
"presentacion": "Unidad",
"cantidad_presentacion": 1,
"unidad_medida_presentacion": "Unitario",
"cantidad_empaque": 1,
"unidad_empaque": "unidad",
"fecha_expiracion": "2024-11-20",
"categoria": {
"_id": "20247",
"nombre": "Legumbres",
"descripcion": "Todo desde...",
"caracteristicas_de_almacenamiento": "Lugar seco y fresco"
},
"_class": "uniandes.edu.co.demo.modelo.Producto"
}

RFC2



PUNTO CINCO

Funcionalidad del esquema de validación

proveedores:

```
>_MONGOSH

> use SuperAndes
< switched to db SuperAndes
> db.proveedores.insertOne({
    _id: 12345, // debe ser una cadena
    nombre: "C", // debe ser mayor a 3 caracteres
    direccion: "", // debe ser mayor a 1 caracter
    nombre_contacto: null, // no puede ser nulo
    telefono_contacto: "123456789012345" // longitud
  });
▸ MongoServerError: Document failed validation
Atlas atlas-v82rze-shard-0 [primary] SuperAndes>
```

Ordenes De Compra:

```
>_MONGOSH

> use SuperAndes
< switched to db SuperAndes
> db.ordenesdecompra.insertOne({
    _id: "123", // debe ser un entero
    sucursal: null, // es requerido (no debe ser nulo)
    proveedor: "Proveedor XYZ", // debe ser un objeto
    estado: 12345, // debe ser una cadena
    fecha_entrega: 20231129, // debe ser una cadena
    fecha_creacion: null // es requerido
  });
▸ MongoServerError: Document failed validation
Atlas atlas-v82rze-shard-0 [primary] SuperAndes>
```

Productos:

```
>_MONGOSH

> use SuperAndes
< switched to db SuperAndes
> db.productos.insertOne({
    _id: 148, // debe ser un string
    nombre: "chocolate",
    precio_unitario: 25000,
    presentacion: null, // no puede ser nulo
    cantidad_presentacion: "diez", // debe ser un entero
    unidad_medida_presentacion: "Unitario",
    cantidad_empaque: -1, // no puede ser negativo
    unidad_empaque: "Caja",
    fecha_expiracion: "2025-10-11",
    categoria: "Frutas" // debe ser un objeto
  });
▸ MongoServerError: Document failed validation
Atlas atlas-v82rze-shard-0 [primary] SuperAndes>
```

Categoria:

```
>_MONGOSH

> use SuperAndes
< switched to db SuperAndes
> db.categorias.insertOne({
    _id: 12345, // debe ser un string
    nombre: "sopas",
    descripcion: null, // es requerido
    caracteristicas_de_almacenamiento: "mantener en lugares frescos"
  });
▸ MongoServerError: Document failed validation
Atlas atlas-v82rze-shard-0 [primary] SuperAndes>
```

Bodegas:

```
> use SuperAndes
< switched to db SuperAndes
> db.bodegas.insertOne({
    _id: "uno", // debe ser un entero
    nombre: "bodega de prueba",
    tamano_metros: 100, //
    productos: null //debe ser un array
  });
▸ MongoServerError: Document failed validation
Atlas atlas-v82rze-shard-0 [primary] SuperAndes>
```

Sucursales:

```
>_MONGOSH
> use SuperAndes
< switched to db SuperAndes
> db.sucursales.insertOne({
    _id: 48,
    nombre: "Sucursal de prueba",
    tamano_metros: 500, //
    direccion: "calle 22 #64-8",
    telefono: 12345, // debe ser un string
    ciudad: "Cali",
    bodegas: "bodega1" // debe ser un array de enteros
  });
▸ MongoServerError: Document failed validation
Atlas atlas-v82rze-shard-0 [primary] SuperAndes>
```

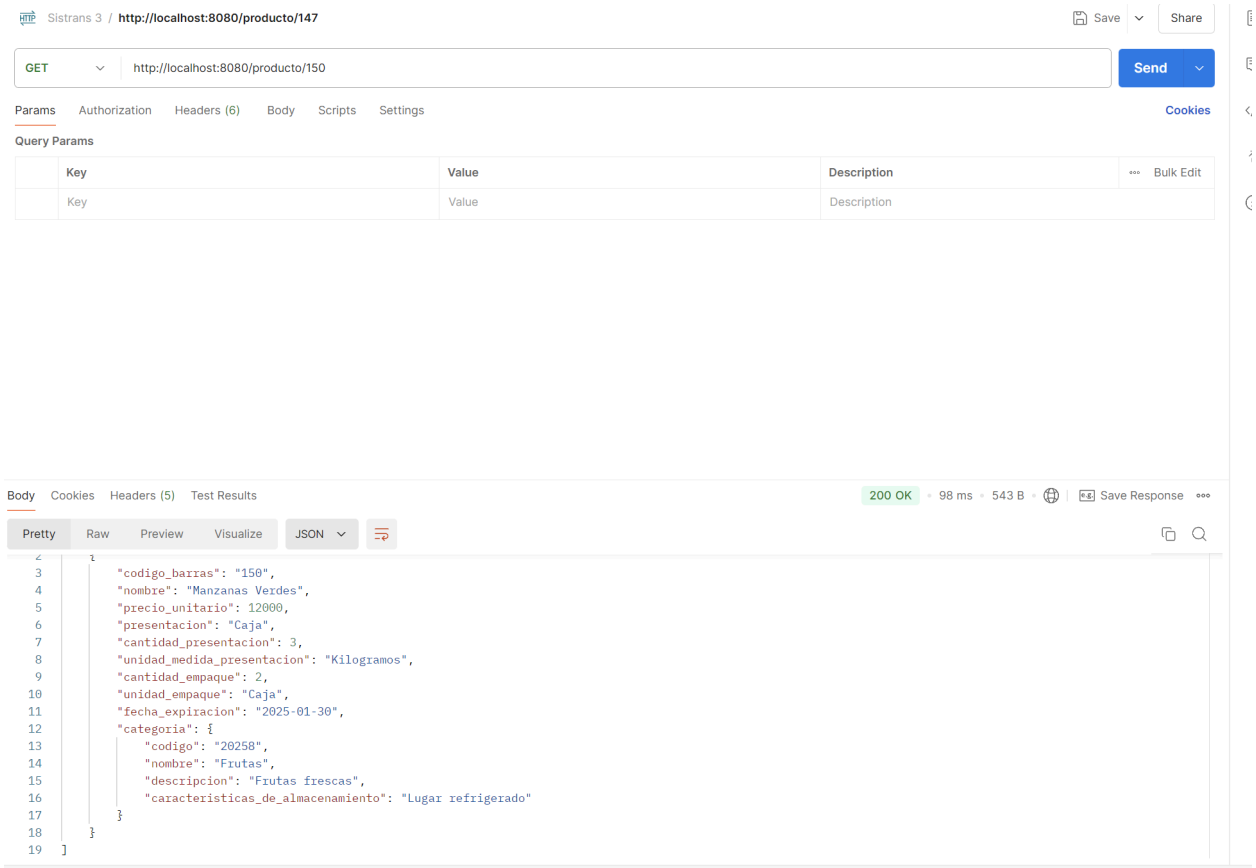
ProductosenBodega:

poblamiento de datos.

En el repositorio se encuentran tanto los scripts del esquema de validación como un pequeño script para poblar la base de datos.
En el script podemos encontrar que se agregó este objeto:

```
{
  _id: "150",
  nombre: "Manzanas Verdes",
  precio_unitario: 12000,
  presentacion: "Caja",
  cantidad_presentacion: 3,
  unidad_medida_presentacion: "Kilogramos",
  cantidad_empaque: 2,
  unidad_empaque: "Caja",
  fecha_expiracion: "2025-01-30",
  categoria: {
    _id: "20258",
    nombre: "Frutas",
    descripcion: "Frutas frescas",
    caracteristicas_de_almacenamiento: "Lugar refrigerado"
  },
 }
]);
```

Utilizamos postman para evidenciar la correcta entrada de este objeto en la base de datos



En mongoDB también podemos verificar la correcta entrada y que se ajustó todos los esquemas de validación

```
}
> db.productos.find({ _id: "150" }).pretty();
< {
  _id: '150',
  nombre: 'Manzanas Verdes',
  precio_unitario: 12000,
  presentacion: 'Caja',
  cantidad_presentacion: 3,
  unidad_medida_presentacion: 'Kilogramos',
  cantidad_empaque: 2,
  unidad_empaque: 'Caja',
  fecha_expiracion: '2025-01-30',
  categoria: {
    _id: '20258',
    nombre: 'Frutas',
    descripcion: 'Frutas frescas',
    caracteristicas_de_almacenamiento: 'Lugar refrigerado'
  },
  _class: 'uniandes.edu.co.demo.modelo.Producto'
}
```