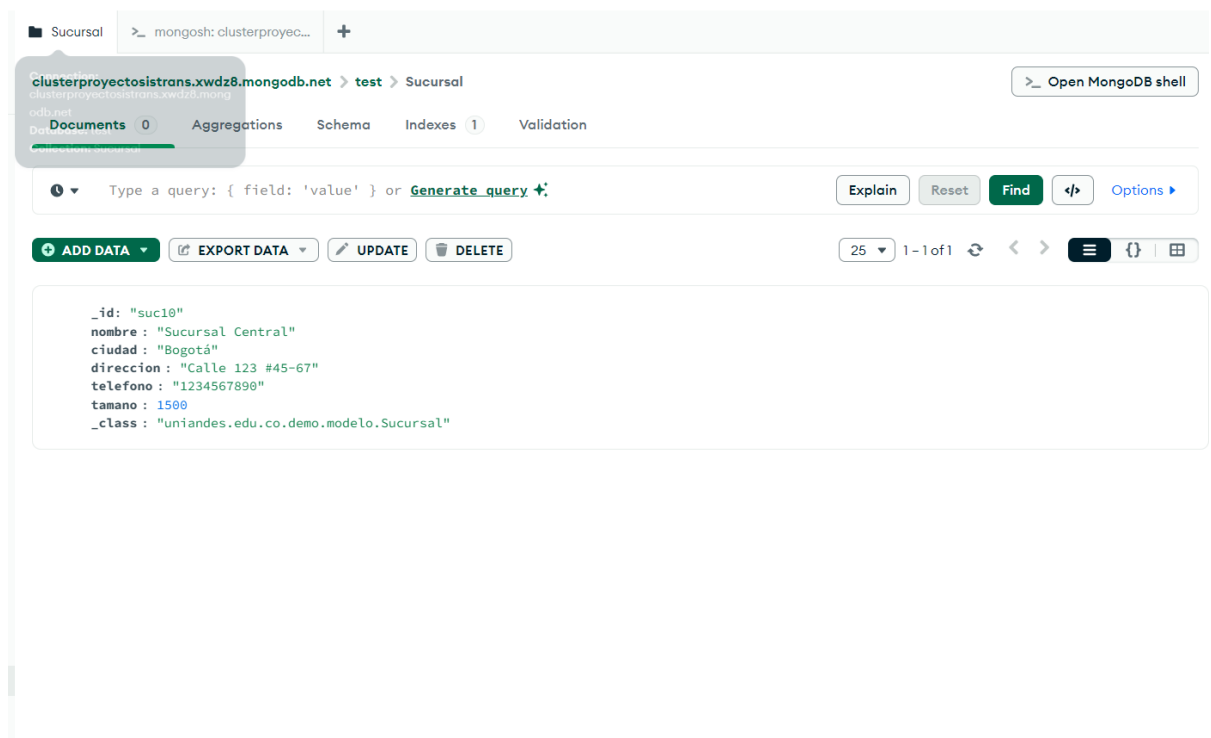
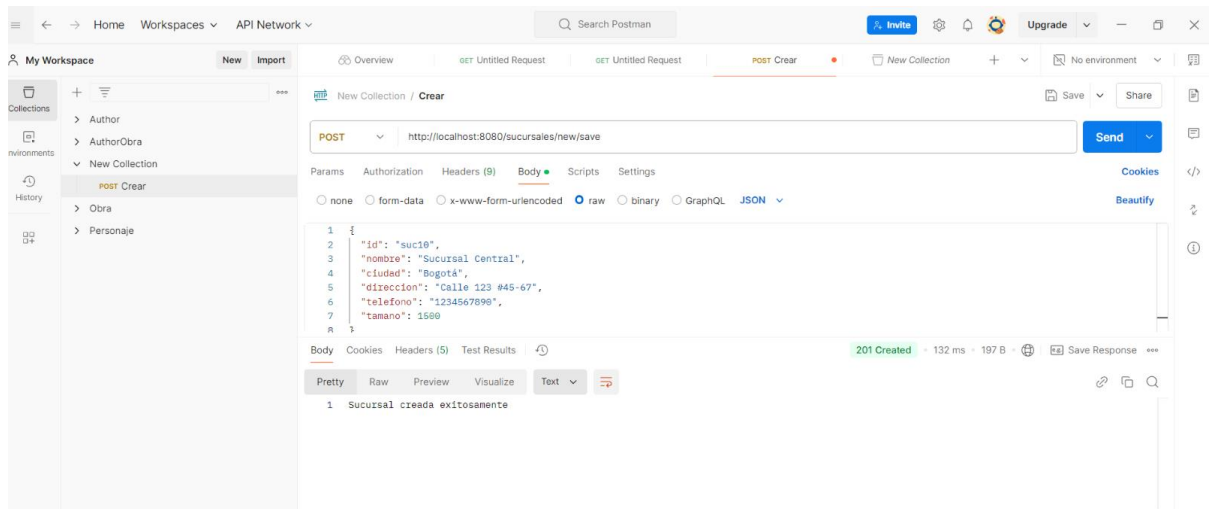


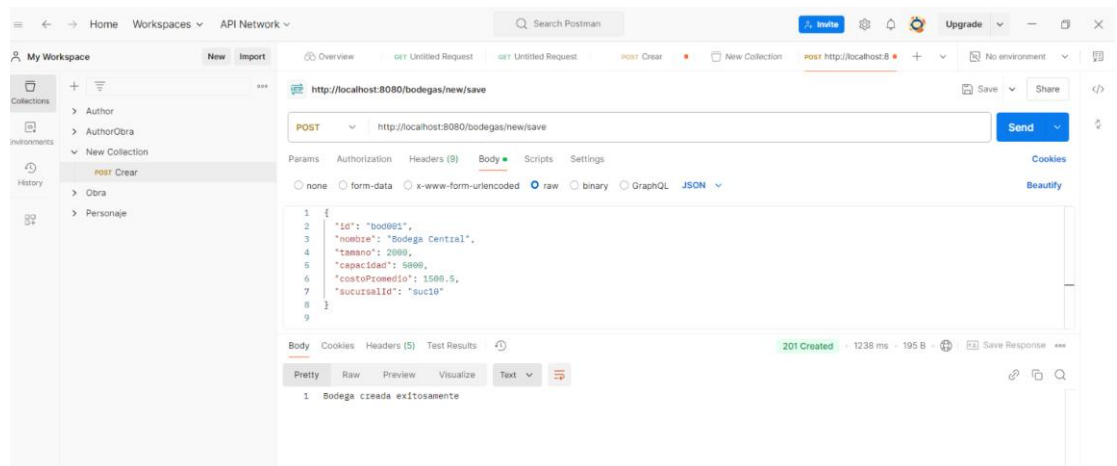
Pruebas Requerimientos Funcionales:

1. RF1 - CREAR UNA SUCURSAL Debe considerar la información básica de una sucursal. Recuerde que la sucursal está asociada a una sola ciudad.



2. RF2 - CREAR Y BORRAR UNA BODEGA Debe considerar toda la información necesaria de una bodega según el enunciado. Recuerde que una bodega debe estar asociada a una sucursal y una sucursal puede tener una o más bodegas.

- CREAR con Sucursal existente:



clusterproyectosistrans.xwdz8.mongodb.net > test > Bodega

Open MongoDB shell

Documents 41 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

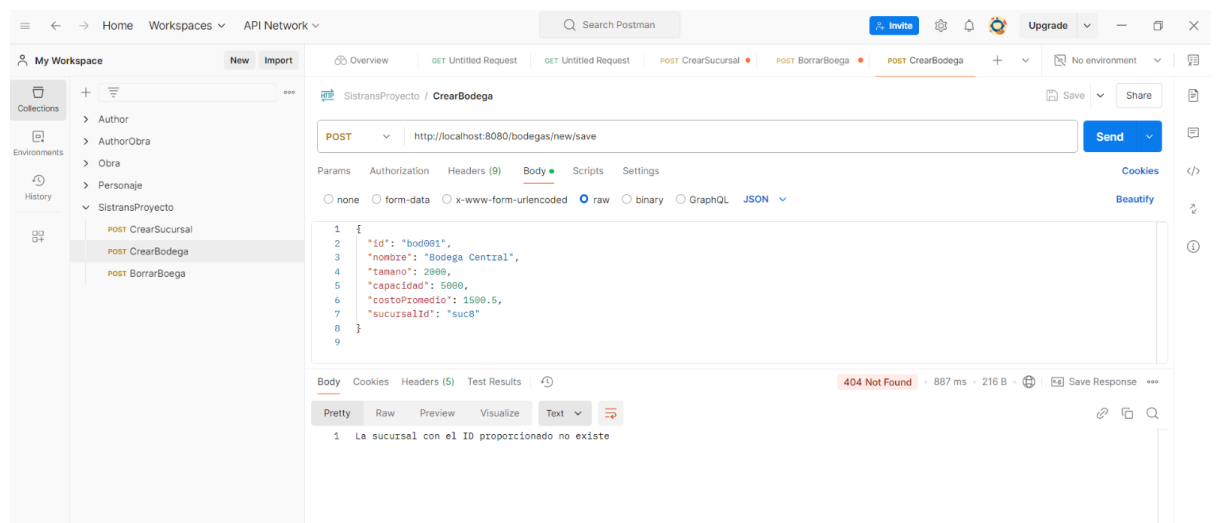
Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

25 1-1 of 1

```
_id: "bod001"
nombre: "Bodega Central"
tamano: 2000
capacidad: 5000
costoPromedio: 1500.5
sucursalId: "suc10"
_class: "uniandes.edu.co.demo.modelo.Bodega"
```

- CREAR con Sucursal no existe:



- BORRAR:

Overview GET Untitled Request GET Untitled Request POST CrearSucursal DEL **BorrarBoega** POST CrearBodega + No environment

SistransProyecto / **BorrarBoega** Save Share

DELETE http://localhost:8080/bodegas/bod001/delete Send

Params Authorization Headers (9) Body **Scripts** Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 260 ms 193 B Save Response

Pretty Raw Preview Visualize Text


1 Bodega eliminada exitosamente

clusterproyectosistrans.xwdz8.mongodb.net > test > Bodega Open MongoDB shell

Documents 1 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE 25 0 - 0 of 0



This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

Import Data

3. RF3 - CREAR Y ACTUALIZAR PROVEEDORES Debe tener en cuenta toda la información requerida para crear un nuevo proveedor y actualizar un proveedor existente.
- CREAR:

SistransProyecto / CrearProveedor

POST http://localhost:8080/proveedores/new/save

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "nit": "prov123",
3   "nombre": "Proveedor Ejemplo",
4   "direccion": "Calle 45 #67-89",
5   "nombreContacto": "Juan Pérez",
6   "telefonoContacto": "3216549870"
7 }
8
```

Body Cookies Headers (5) Test Results

201 Created • 102 ms • 198 B • Save Response

Pretty Raw Preview Visualize Text

```
1 Proveedor creado exitosamente
```

clusterproyectosistrans.xwdz8.mongodb.net > test > Proveedor

Documents (0) Aggregations Schema Indexes (1) Validation

Type a query: { field: 'value' } or Generate query

EXPLAIN RESET FIND

ADD DATA EXPORT DATA UPDATE DELETE

25 1 of 1

```
_id: "prov123"
nombre: "Proveedor Ejemplo"
direccion: "Calle 45 #67-89"
nombreContacto: "Juan Pérez"
telefonoContacto: "3216549870"
_class: "uniandes.edu.co.demo.modelo.Proveedor"
```

- ACTUALIZAR:

Overview GET Untitled Rec GET Untitled Rec POST CrearSucu DEL BorrارBoeg POST CrearBode POST CrearProve PUT BorrارProve + No environment

SistransProyecto / BorrارProveedor

PUT http://localhost:8080/proveedores/prov123/edit/save

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "nombre": "Proveedor Ejemplo Actualizado",
3   "direccion": "Calle 100 #50-20",
4   "nombreContacto": "María Gómez",
5   "telefonoContacto": "3109876543"
6 }
7
```

Body Cookies Headers (5) Test Results

200 OK • 1351 ms • 198 B • Save Response

Pretty Raw Preview Visualize Text

```
1 Proveedor actualizado exitosamente
```

4. RF4 - CREAR UNA CATEGORÍA DE PRODUCTO Para la creación, se debe registrar la información de las categorías de productos según el enunciado. Para la lectura, se solicita el código o nombre de la categoría, y como resultado, se obtiene la información de dicha categoría.

- CREAR:

The image shows two screenshots related to a REST API test and its data storage in MongoDB.

Top Screenshot (REST Client):

- URL:** `http://localhost:8080/categorias/new/save`
- Method:** `POST`
- Body (JSON):**

```
{
  "id": "cat001",
  "nombre": "Lacteos",
  "descripcion": "Productos derivados de la leche",
  "caracteristicasAlmacenamiento": "Refrigeración a 4°C"
}
```
- Status:** `201 Created` (138 ms, 199 B)
- Response Body:** `1 Categoría creada exitosamente`

Bottom Screenshot (MongoDB):

- Database:** `clusterproyectosistrans.xwdz8.mongodb.net`
- Collection:** `Categoria`
- Documents:** 0
- Query:** `Type a query: { field: 'value' } or Generate query`
- Document:**

```
{
  "_id": "cat001",
  "nombre": "Lacteos",
  "descripcion": "Productos derivados de la leche",
  "caracteristicasAlmacenamiento": "Refrigeración a 4°C",
  "_class": "uniandes.edu.co.demo.modelo.Categoria"
}
```

Obtener información por id y nombre:

Overview
GET Untitled
GET Untitled
POST CrearS
DEL BorrarrB
POST CrearE
POST CrearF
PUT BorrarP
POST CrearI
GET Catego
+
No environment

SistransProyecto / **CategoriaPorID**

GET
http://localhost:8080/categorias/cat001
Send

Params
Authorization
Headers (7)
Body
Scripts
Settings
Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body
Cookies
Headers (5)
Test Results
200 OK
440 ms
302 B
Save Response

Pretty
Raw
Preview
Visualize
JSON

```

1 {
2   "id": "cat001",
3   "nombre": "Lacteos",
4   "descripcion": "Productos derivados de la leche",
5   "caracteristicasAlmacenamiento": "Refrigeración a 4°C"
6 }

```

Postbot Ctrl Alt P

SistransProyecto / **CategoriaPorNombre**

GET
http://localhost:8080/categorias/buscarPorNombre/Lacteos
Send

Params
Authorization
Headers (7)
Body
Scripts
Settings
Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body
Cookies
Headers (5)
Test Results
200 OK
451 ms
304 B
Save Response

Pretty
Raw
Preview
Visualize
JSON

```

1 [
2   {
3     "id": "cat001",
4     "nombre": "Lacteos",
5     "descripcion": "Productos derivados de la leche",
6     "caracteristicasAlmacenamiento": "Refrigeración a 4°C"
7   }
8 ]

```

Postbot Ctrl Alt P

- RF5 - CREAR UN PRODUCTO** Para la creación de un producto, debe registrar toda la información de los productos según el enunciado. Recuerde que, en su creación, el producto debe ser asociado a una categoría existente. Recuerde igualmente que una categoría puede tener múltiples productos asociados. Para la lectura, se solicita el código o nombre del producto, y como resultado, se obtiene la información de dicho producto incluyendo la categoría a la que pertenece. Para la Actualización, se solicita el código del producto, y la

información que se quiere actualizar, y como resultado, se obtiene la actualización de dicho producto

- CREAR:

The image shows two screenshots related to a REST API test and database storage.

Top Screenshot (REST Client):

- Method:** POST
- URL:** http://localhost:8080/productos/new/save
- Body (JSON):**

```
2  {"codigoBarras": "prod001",
3   "nombre": "Leche Entera",
4   "costoBodega": 1500.0,
5   "precioVenta": 2000.0,
6   "presentacion": "Litro",
7   "cantidadPresentacion": 1,
8   "unidadMedida": "Litros",
9   "volumenEmpaque": 1000,
10  "pesoEmpaque": 1200,
11  "fechaExpiracion": "2025-12-31",
12  "categoriaId": "cat001", // Debe ser un ID válido de una categoría existente
13  "bodegasIds": ["bod001"], // Lista de bodegas a las que pertenece el producto (opcional)
14  "proveedoresIds": ["prov123"] // Lista de proveedores que suministran el producto (opcional)
15  }
16
```
- Status:** 201 Created
- Response Body:** 1 Producto creado exitosamente

Bottom Screenshot (MongoDB):

- Database:** clusterproyectosistrans.xwdz8.mongodb.net
- Collection:** Producto
- Documents:** 0
- Query:** Type a query: { field: 'value' } or [Generate query](#)
- Document Preview:**

```
_id: "prod001"
nombre: "Leche Entera"
costoBodega: 1500
precioVenta: 2000
presentacion: "Litro"
cantidadPresentacion: 1
unidadMedida: "Litros"
volumenEmpaque: 1000
pesoEmpaque: 1200
fechaExpiracion: 2025-12-31T00:00:00.000+00:00
categoriaId: "cat001"
bodegasIds: Array (1)
proveedoresIds: Array (1)
_class: "uniandes.edu.co.demo.modelo.Producto"
```

- Lectura:

SistransProyecto / ObtenerProductoCodigo

GET

http://localhost:8080/productos/prod001

Send

ParamsAuthorizationHeaders (7)BodyScriptsSettings

Query Params

	Key	Value	Description	Bulk Edit
	Key	Value	Description	

BodyCookiesHeaders (5)Test Results

200 OK • 157 ms • 507 B

Save Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "codigoBarras": "prod001",
3   "nombre": "Leche Entera Actualizada",
4   "costoBodega": 1600.0,
5   "precioVenta": 2100.0,
6   "presentacion": "Litro",
7   "cantidadPresentacion": 1,
8   "unidadMedida": "Litros",
9   "volumenEmpaque": 1100,
10  "pesoEmpaque": 1300,
11  "fechaExpiracion": "2026-01-31T00:00:00.000+00:00",
12  "categoriaId": "cat001",
13  "bodegasIds": [
14    "bod001"
15  ],
16  "proveedoresIds": [
17    "prov123"
18  ]
19 }
```

SistransProyecto / ObtenerProductoNombre

GET

http://localhost:8080/productos/buscarPorNombre/Leche Entera Actualizada

Send

ParamsAuthorizationHeaders (7)BodyScriptsSettings

Query Params

	Key	Value	Description	Bulk Edit
	Key	Value	Description	

BodyCookiesHeaders (8)Test Results

404 Not Found • 14 ms • 403 B

Save Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "timestamp": "2024-12-01T23:37:10.430+00:00",
3   "status": 404,
4   "error": "Not Found",
5   "path": "/productos/buscarPorNombre/Leche%20Entera%20Actualizada"
6 }
```

- Actualización:

SistransProyecto / ActualizacionProducto

PUT http://localhost:8080/productos/prod001/edit

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "nombre": "Leche Entera Actualizada",
3   "costoBodega": 1600.0,
4   "precioVenta": 2100.0,
5   "presentacion": "Litro",
6   "cantidadPresentacion": 1,
7   "unidadMedida": "Litros",
8   "volumenEmpaque": 1100,
9   "pesoEmpaque": 1300,
10  "fechaExpiracion": "2026-01-31",
11  "categoriaId": "cat001", // Debe ser un ID válido de una categoría existente
12  "bodegasIds": ["bod001"], // Lista de bodegas a las que pertenece el producto (opcional)
13  "proveedoresIds": ["prov123"] // Lista de proveedores que suministran el producto (opcional)
14 }
15

```

Body Cookies Headers (5) Test Results

200 OK • 326 ms • 197 B

Pretty Raw Preview Visualize Text

1 Producto actualizado exitosamente

clusterprojectosistrans.xwdz8.mongodb.net > test > Producto

Documents 0 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or Generate query

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

25 1 - 1 of 1

```

_id: "prod001"
nombre: "Leche Entera Actualizada"
costoBodega: 1600
precioVenta: 2100
presentacion: "Litro"
cantidadPresentacion: 1
unidadMedida: "Litros"
volumenEmpaque: 1100
pesoEmpaque: 1300
fechaExpiracion: 2026-01-31T00:00:00.000+00:00
categoriaId: "cat001"
bodegasIds: Array (1)
proveedoresIds: Array (1)
_class: "uniandes.edu.co.demo.modelo.Producto"

```

6. RF6 - CREAR UNA ORDEN DE COMPRA PARA UNA SUCURSAL Debe considerar toda la información de las órdenes de compra según lo descrito arriba. Para completar este documento, desde un punto de vista gráfico, si el usuario tuviera una interfaz gráfica, vería dos partes: el encabezado y el detalle. El encabezado contiene los siguientes datos de entrada: - La fecha de creación de la orden (corresponde a la fecha del día según el sistema, la cual aparece automáticamente y el usuario no la puede modificar). - La sucursal, si se tuviera interfaz gráfica, es escogida por el usuario de la lista de sucursales ya existentes. En postman se recibiría el id de la sucursal. - El proveedor al que se le quiere comprar, si se tuviera interfaz gráfica, es escogido de la lista de

proveedores ya existentes. En postman se recibiría el id del proveedor. - La fecha esperada de entrega de los productos, la cual es ingresada por el usuario. El detalle por su lado contiene: - Uno o más productos que se quieren comprar, si se tuviera interfaz gráfica, son escogidos por el usuario de los productos existentes. En postman se recibiría el id de cada producto. - Las cantidades de los productos que se desean comprar, los cuales son ingresados por el usuario. - Los precios de los productos que se desean comprar, los cuales son ingresados por el usuario. Al ser creada, la orden de compra queda en estado “vigente”.

- CREAR:

The image shows two screenshots. The top screenshot is from Postman, showing a POST request to `http://localhost:8080/ordenesCompra/new/save` with a JSON body. The body contains an order with ID "orden001", a branch ID "suc10", a provider ID "prov123", an expected delivery date "2024-12-15T00:00:00Z", and two details: "detalle001" (product "prod001", quantity 50, price 15.0) and "detalle002" (product "prod002", quantity 30, price 20.0). The response is a 201 Created status with the message "Orden de compra creada exitosamente".

The bottom screenshot is from the MongoDB Compass interface, showing the document stored in the `OrdenCompra` collection. The document includes fields for `_id`, `sucursalId`, `proveedorId`, `fechaCreacion`, `fechaEsperadaEntrega`, `estado`, `detalles` (an array of two objects), and `_class`.

```

{
  "_id": "orden001",
  "sucursalId": "suc10",
  "proveedorId": "prov123",
  "fechaCreacion": "2024-12-01T23:58:08.054+00:00",
  "fechaEsperadaEntrega": "2024-12-15T00:00:00.000+00:00",
  "estado": "vigente",
  "detalles": [
    {
      "id": "detalle001",
      "productoId": "prod001",
      "cantidad": 50,
      "precio": 15.0
    },
    {
      "id": "detalle002",
      "productoId": "prod002",
      "cantidad": 30,
      "precio": 20.0
    }
  ],
  "_class": "uniandes.edu.co.demo.modelo.OrdenCompra"
}

```

7. RF7 – LEER UNA ORDEN DE COMPRA La aplicación recibe el número de la orden de compra. Como resultado se obtiene la información del encabezado y del detalle descritos antes.

-CONSULTA:

POST Crea

GET Categ

GET Categ

POST Crea

PUT Actua

GET Obter

GET Obter

Orden

POST Cre

GET Obte.

> + ▾

No environment ▾

SistransProyecto / ObtenerOrdenCompralID

Save ▾Share

GET ▾http://localhost:8080/ordenesCompra/orden001

Send ▾

ParamsAuthorizationHeaders (7)BodyScriptsSettings

Cookies

BodyCookiesHeaders (5)Test Results🕒

200 OK • 187 ms • 503 B • 🌐📄 Save Response ⋮

PrettyRawPreviewVisualizeJSON ▾🔧

1 {

2 "id": "orden001",

3 "sucursalId": "suc10",

4 "proveedorId": "prov123",

5 "fechaCreacion": "2024-12-01T23:58:00.054+00:00",

6 "fechaEsperadaEntrega": "2024-12-15T00:00:00.000+00:00",

7 "estado": "vigente",

8 "detalles": [

9 {

10 "id": "detalle001",

11 "productoId": "prod001",

12 "cantidad": 50,

13 "precio": 15.0

14 },

15 {

16 "id": "detalle002",

17 "productoId": "prod002",

18 "cantidad": 30,

19 "precio": 20.0

20 }

21]

22 }

RF Búsqueda:

1. Rango Precio:

POST Crea

GET Categ

GET Categ

POST Crea

PUT Actua

GET Obter

GET Obter

Orden

POST Cre

GET Obte.

> + ▾

No environment ▾

SistransProyecto / ProductosRangoPrecio

Save ▾Share

GET ▾http://localhost:8080/productos/rango-precio?precioMin=10&precioMax=20

Send ▾

ParamsAuthorizationHeaders (7)BodyScriptsSettings

Cookies

BodyCookiesHeaders (5)Test Results🕒

200 OK • 132 ms • 844 B • 🌐📄 Save Response ⋮

PrettyRawPreviewVisualizeJSON ▾🔧

1 [

2 {

3 "codigoBarras": "1234567890123",

4 "nombre": "Queso Cheddar",

5 "costoBodega": 15.0,

6 "precioVenta": 20.0,

7 "presentacion": "Bloque",

8 "cantidadPresentacion": 500,

9 "unidadMedida": "gramos",

10 "volumenEmpaque": 300,

11 "pesoEmpaque": 500,

12 "fechaExpiracion": "2024-12-31T00:00:00.000+00:00",

13 "estado": "vigente"

14 }

15]

- Fecha de Expiracion:

SistransProyecto / FechaExpiracion

GET

http://localhost:8080/productos/fecha-expiracion-posterior?fecha=2024-12-01

Send

Params • Authorization Headers (7) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	fecha	2024-12-01			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK • 116 ms • 501 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "codigoBarras": "1234567890123",
4     "nombre": "Queso Cheddar",
5     "costoBodega": 15.0,
6     "precioVenta": 20.0,
7     "presentacion": "Bloque",
8     "cantidadPresentacion": 500,
9     "unidadMedida": "gramos",
10    "volumenEmpaque": 300,
11    "pesoEmpaque": 500,
12    "fechaExpiracion": "2024-12-31T00:00:00.000+00:00",
13    "categoriaId": "cat001",
14    "bodegasIds": [
15      "bod001"
16    ],
17    "proveedoresIds": [
18      "prov001"
19    ]
20  }
21 ]
```

SistransProyecto / FechaExpiracionInferior

GET

http://localhost:8080/productos/fecha-expiracion-inferior?fecha=2024-12-01

Send

Params • Authorization Headers (7) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	fecha	2024-12-01			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK • 104 ms • 508 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "codigoBarras": "1234567890124",
4     "nombre": "Leche Entera",
5     "costoBodega": 10.0,
6     "precioVenta": 15.0,
7     "presentacion": "Botella",
8     "cantidadPresentacion": 1000,
9     "unidadMedida": "mililitros",
10    "volumenEmpaque": 1000,
11    "pesoEmpaque": 1050,
12    "fechaExpiracion": "2024-11-30T00:00:00.000+00:00",
13    "categoriaId": "cat001",
14    "bodegasIds": [
15      "bod001"
16    ],
17    "proveedoresIds": [
18      "prov001"
19    ]
20  }
21 ]
```

Postbot
Ctrl Alt P

Postbot Runner Start Proxy Cookies Vault Trash

2. Productos por Sucursal:

SistransProyecto / ProdPorSucursal

GET

http://localhost:8080/bodegas/inventario/suc001

Send

Params

Authorization

Headers (7)

Body

Scripts

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (5)

Test Results

200 OK • 1144 ms • 431 B • Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  [
2    {
3      "cantidadActual": 500,
4      "cantidadMinima": 10,
5      "costoPromedio": 15.0,
6      "nombreBodega": "Bodega Refrigerada",
7      "nombreProducto": "Queso Cheddar"
8    },
9    {
10     "cantidadActual": 1000,
11     "cantidadMinima": 10,
12     "costoPromedio": 15.0,
13     "nombreBodega": "Bodega Refrigerada",
14     "nombreProducto": "Leche Entera"
15   }
16 ]
```

Postbot
Ctrl Alt P

Postbot Runner Start Proxy Cookies Vault Trash