

Sistemas transaccionales

Proyecto grupo 6

Documentación con imágenes

Integrantes:

- María Lucía Benavides
- Ángela Jiménez
- Kevin Castillo

Notas:

Los cambios hechos para que funcionaran todas las pruebas de la entrega 1 eran más que todo de sintaxis y de conexión con bases de datos, además se crearon pruebas para los requerimientos de consulta de la entrega 1 y quedaron funcionando perfectamente.

1. Creación de tablas:

Table CIUDADES created.

Table SUCURSALES created.

Table PROVEEDORES created.

Table CATEGORIAS created.

Table BODEGAS created.

Table CLIENTES created.

Table PRODUCTOS created.

Table RECEPCIONES created.

Table RECEPCIONES created.

Table INFO_RECEPCION created.

Table NIVELES_REORDEN created.

Table ORDENES_COMPRA created.

Table PRODUCTO_BODEGA created.

Table PRODUCTO_ORDEN created.

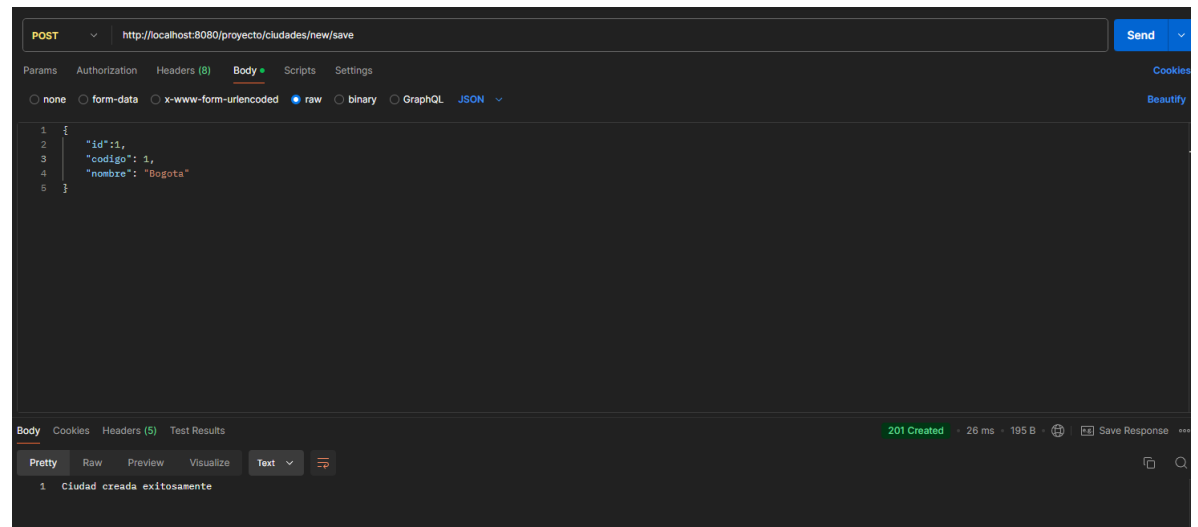
Table VENTAS created.

2. Pruebas en PostMan de requerimientos:

Requerimientos funcionales:

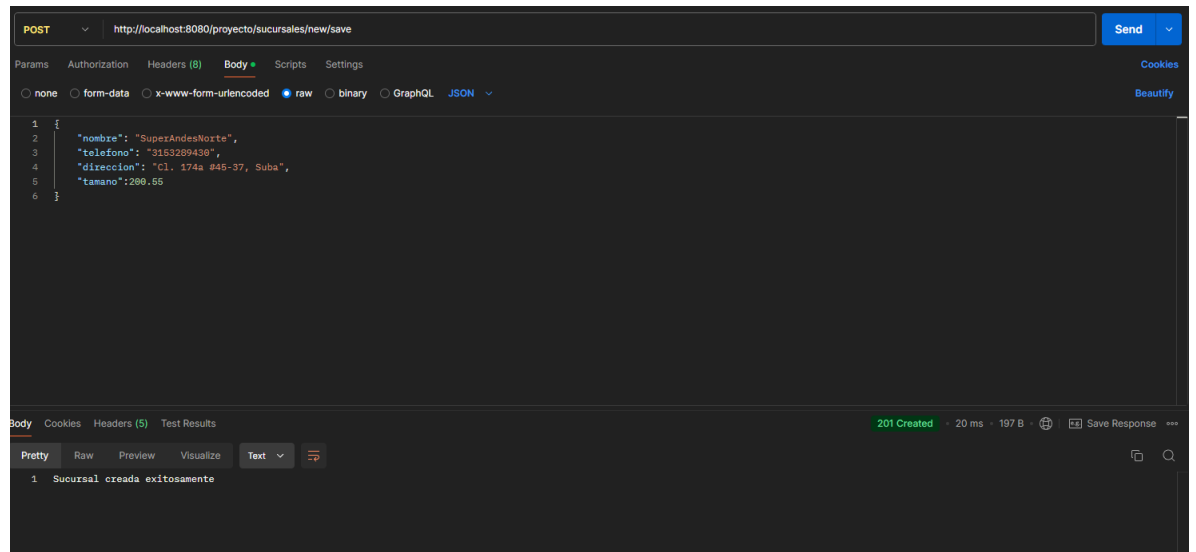
-Requerimiento funcional 1:

Crear una ciudad:



-Requerimiento funcional 2:

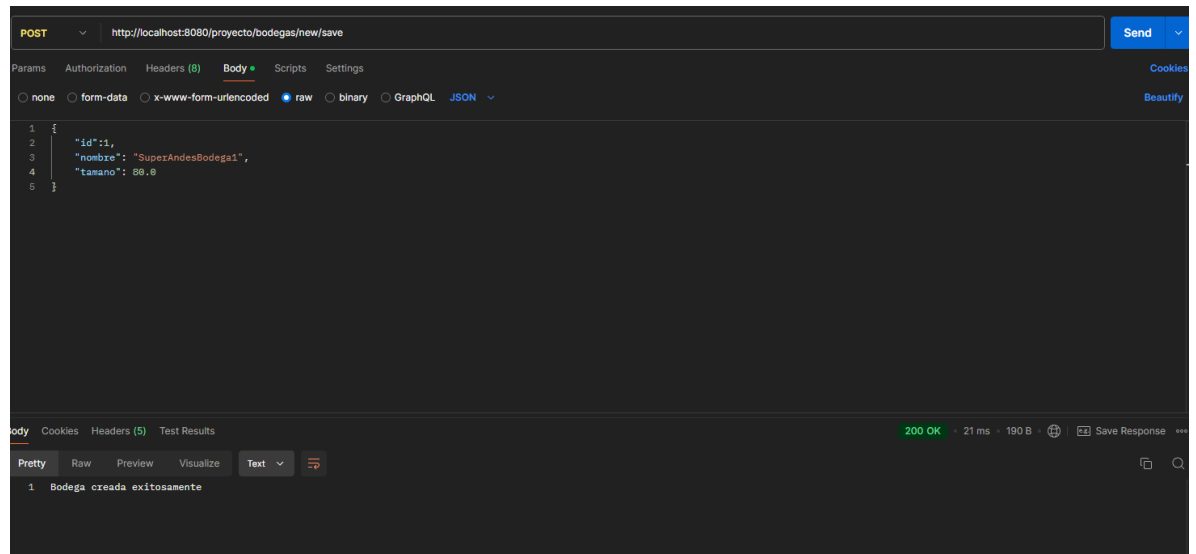
Crear una sucursal:



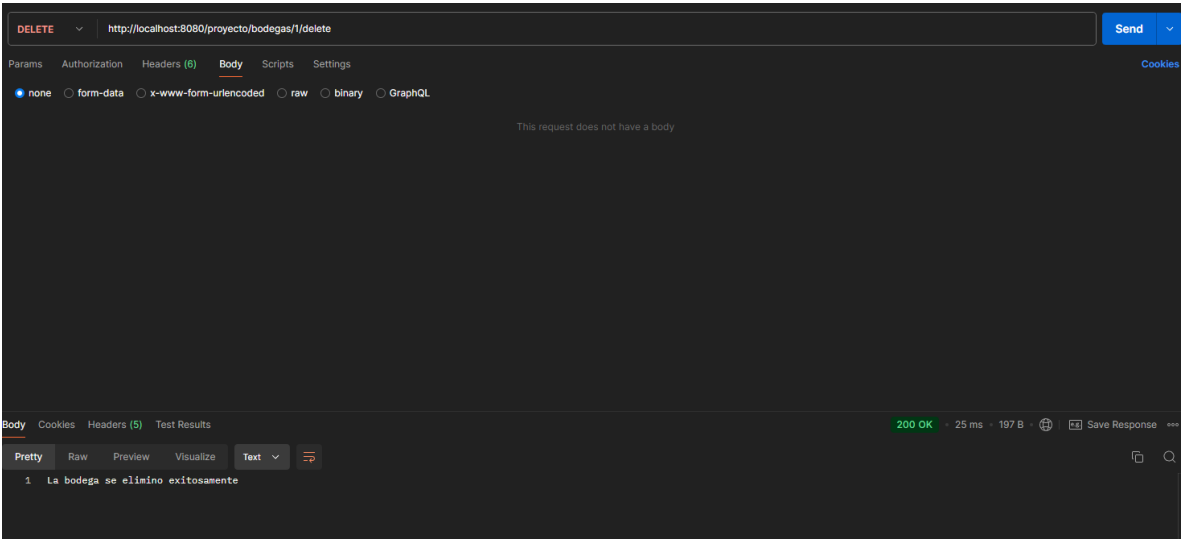
-Requerimiento funcional 3:

Crear y borrar una bodega:

Crear:



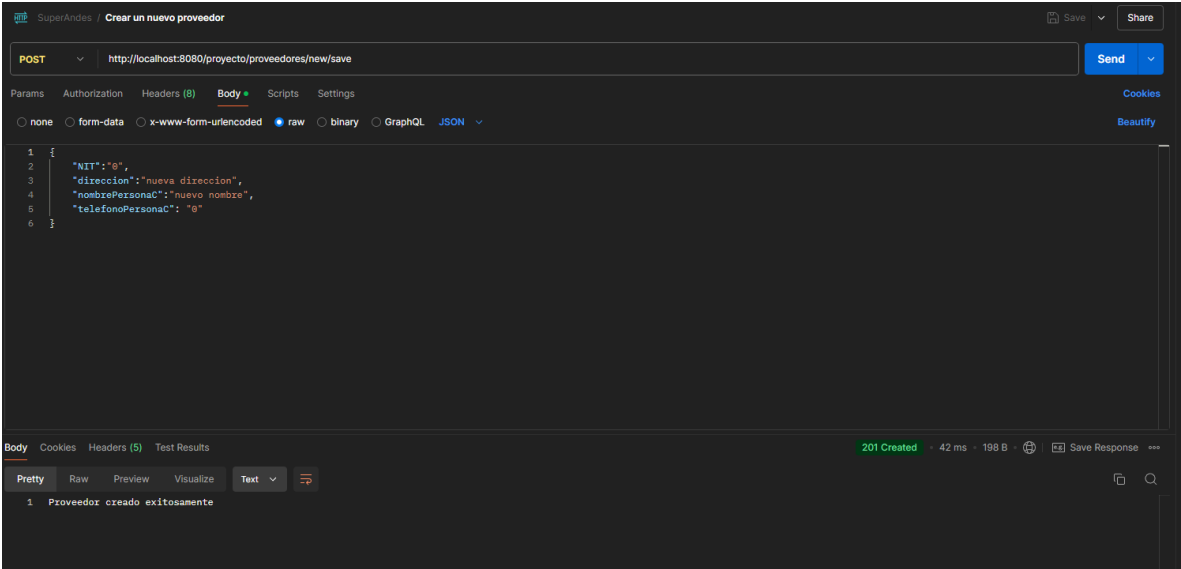
Borrar:



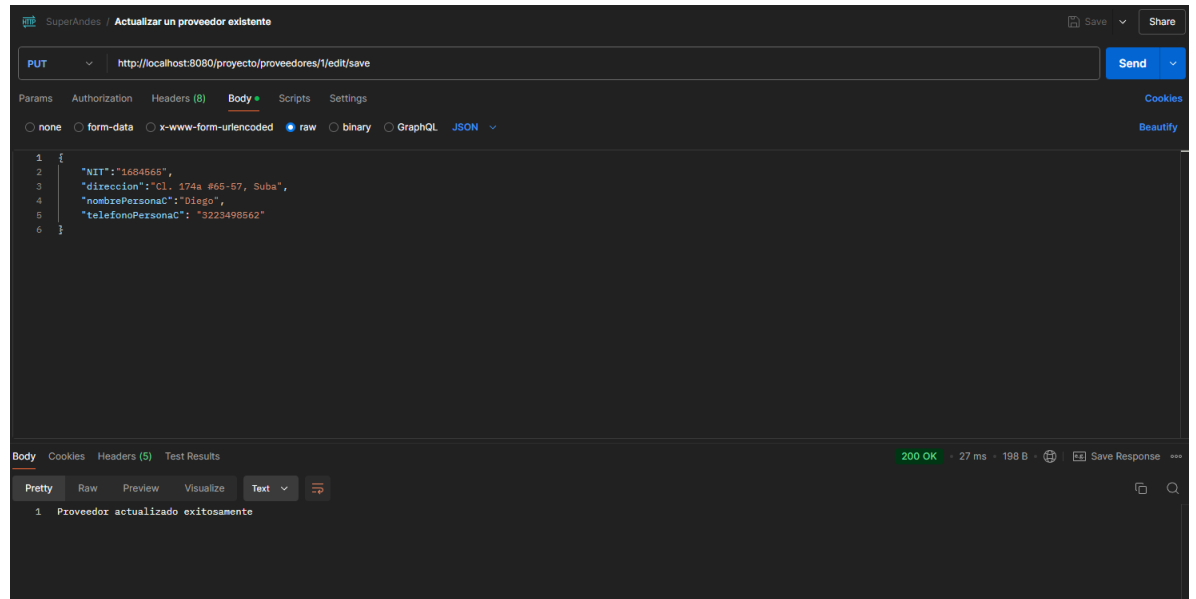
-Requerimiento funcional 4:

Crear y Actualizar Proveedores:

Crear:



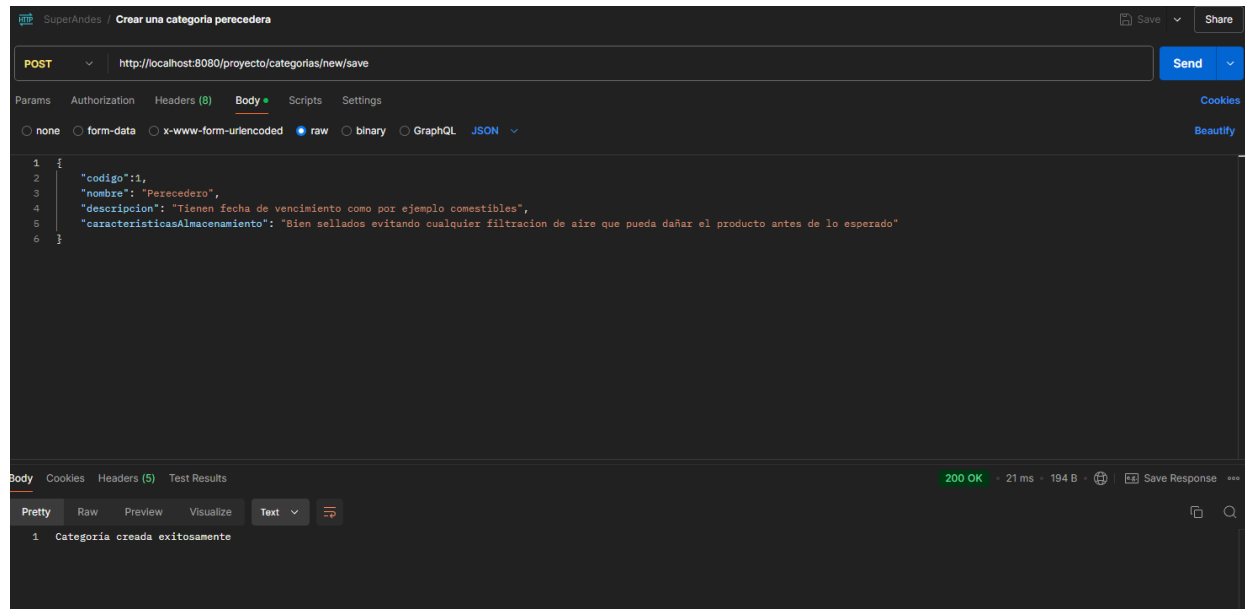
Actualizar:

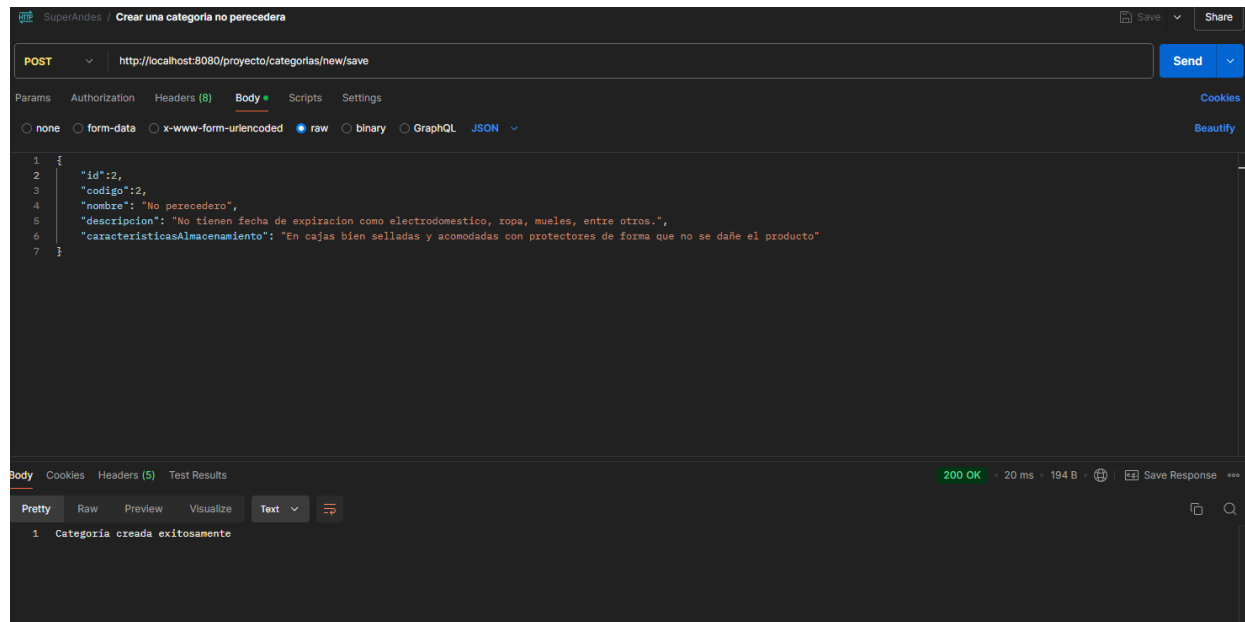


-Requerimiento funcional 5:

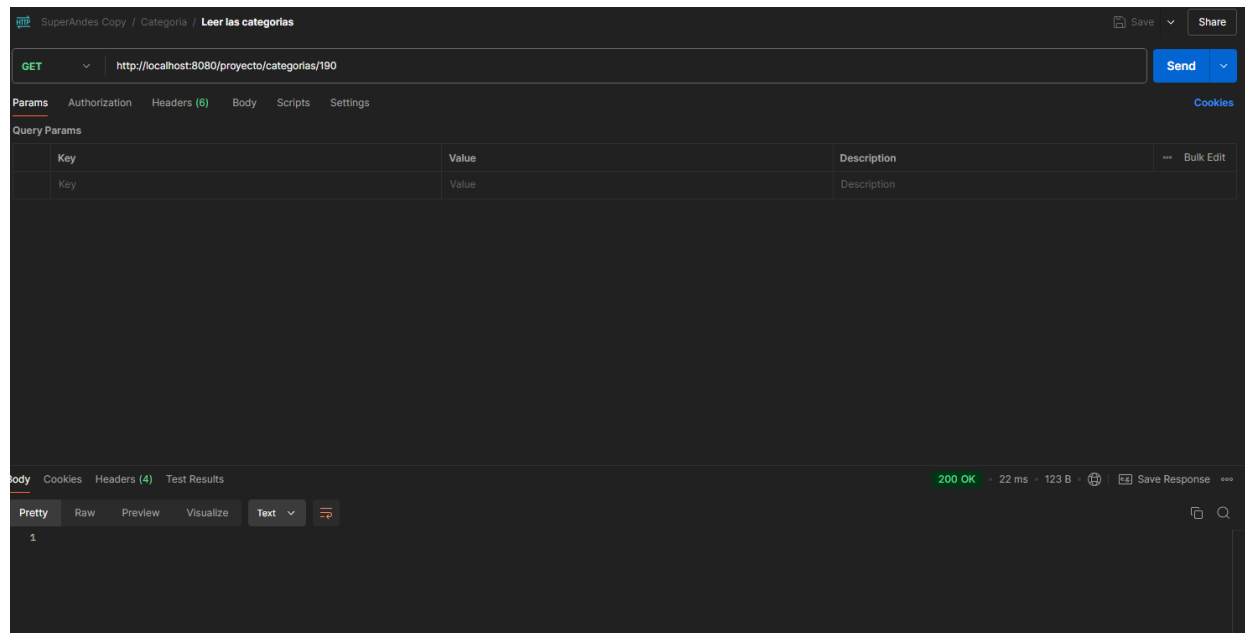
Crear y leer una categoría de producto:

Crear:





Leer:



-Requerimiento funcional 6:

Crear y leer y actualizar un producto:

Crear:

SuperAndes Copy / Producto / Crear un producto

POST

http://localhost:8080/proyecto/productos/new/save

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
2  {
3    "pk": {
4      "codigoBarras": "fASADf0f0f"
5    },
6    "nombre": "Papas fritas",
7    "costoBodega": 4200.00,
8    "precioVenta": 9000.00,
9    "presentacion": "paquetón de 5 paquetes de 200 gr",
10   "cantidadPresentacion": 1000.00,
11   "unidadMedida": 0,
12   "volumenEmpaque": 150.00,
13   "pesoEmpaque": 15.00,
14   "fechaExpiracion": "2021-02-01",
15   "categoria": {
16     "id": 1,
17     "codigo": 1,
18     "nombre": "Perecedero",
19     "descripcion": "Tienen fecha de vencimiento como por ejemplo comestibles",
20     "caracteristicasAlmacenamiento": "Bien sellados evitando cualquier filtración de aire que pueda dañar el producto antes de lo esperado"
21   }
22 }
```

Body

Cookies

Headers (5)

Test Results (1/1)

201 Created · 45 ms · 197 B · Save Response

Pretty

Raw

Preview

Visualize

Text

1

Producto creado exitosamente

Leer:

SuperAndes Copy / Producto / Leer producto por Id y código de barras

GET

http://localhost:8080/proyecto/productos/fASADf0f0f

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Query Params

Key	Value	Description
Key	Value	Description

Body

Cookies

Headers (5)

Test Results

200 OK · 112 ms · 611 B · Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "pk": {
3      "id": 191,
4      "codigoBarras": "fASADf0f0f"
5    },
6    "nombre": "Papas fritas",
7    "presentacion": "paquetón de 5 paquetes de 200 gr",
8    "categoria": {
9      "id": 1,
10     "codigo": 1,
11     "nombre": "Lácteos",
12     "descripcion": "Productos derivados de la leche",
13     "caracteristicasAlmacenamiento": "Refrigeración a 4°C"
14   },
15   "precioVenta": 9000.0,
16   "costoBodega": 4200.0,
```

Actualizar:

SuperAndes Copy / Producto / Actualizar un producto

PUT http://localhost:8080/proyecto/productos/1/ASADf0f0f/edit/save

Params Authorization Headers (8) Body Scripts Settings

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

```
1 {
2   "nombre": "Papas fritas Les frites",
3   "costoBodega": 4200.00,
4   "precioVenta": 9000.00,
5   "presentacion": "paquetón de 5 paquetes de 200 gr. cada uno",
6   "cantidadPresentación": 1000,
7   "unidadMedida (gr o ml)": true,
8   "volumenEmpaque": 150,
9   "pesoEmpaque": 15,
10  "fechaExpiracion": "2027-12-12",
11  "categoria": {
12    "id": 1,
13    "codigo": 1,
14    "nombre": "Perecedero",
15    "descripcion": "Tienen fecha de vencimiento como por ejemplo comestibles",
16    "caracteristicasAlmacenamiento": "Bien sellados evitando cualquier filtración de aire que pueda dañar el producto antes de lo esperado"
17  }
18 }
```

Body Cookies Headers (5) Test Results (1/1)

200 OK 30 ms 197 B Save Response

Pretty Raw Preview Visualize Text

1 Producto actualizado exitosamente

-Requerimiento funcional 7:

Crear una orden de compra para una sucursal:

SuperAndes Copy / Orden de compra / Crear una orden de compra para una sucursal

POST http://localhost:8080/proyecto/ordenCompra/new/save

Params Authorization Headers (8) Body Scripts Settings

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

```
1 {
2   "fechaEsperadaEntrega": "2024-10-15",
3   "precioAcordado": 1500000.00,
4   "estado": "vigente",
5   "fechaCreacion": "2024-09-15",
6   "sucursal": {
7     "id": 1,
8     "nombre": "AAAA",
9     "telefono": "3153209430",
10    "direccion": "Cl. 174a #46-37, Suba",
11    "tamano": 200.55,
12    "ciudad": {
13      "id": 1,
14      "codigo": 1,
15      "nombre": "aaaaaaaa"
16    }
17  },
18  "proveedor": {
19    "id": 1,
20    "NIT": "0",
21  }
22 }
```

Body Cookies Headers (5) Test Results (1/1)

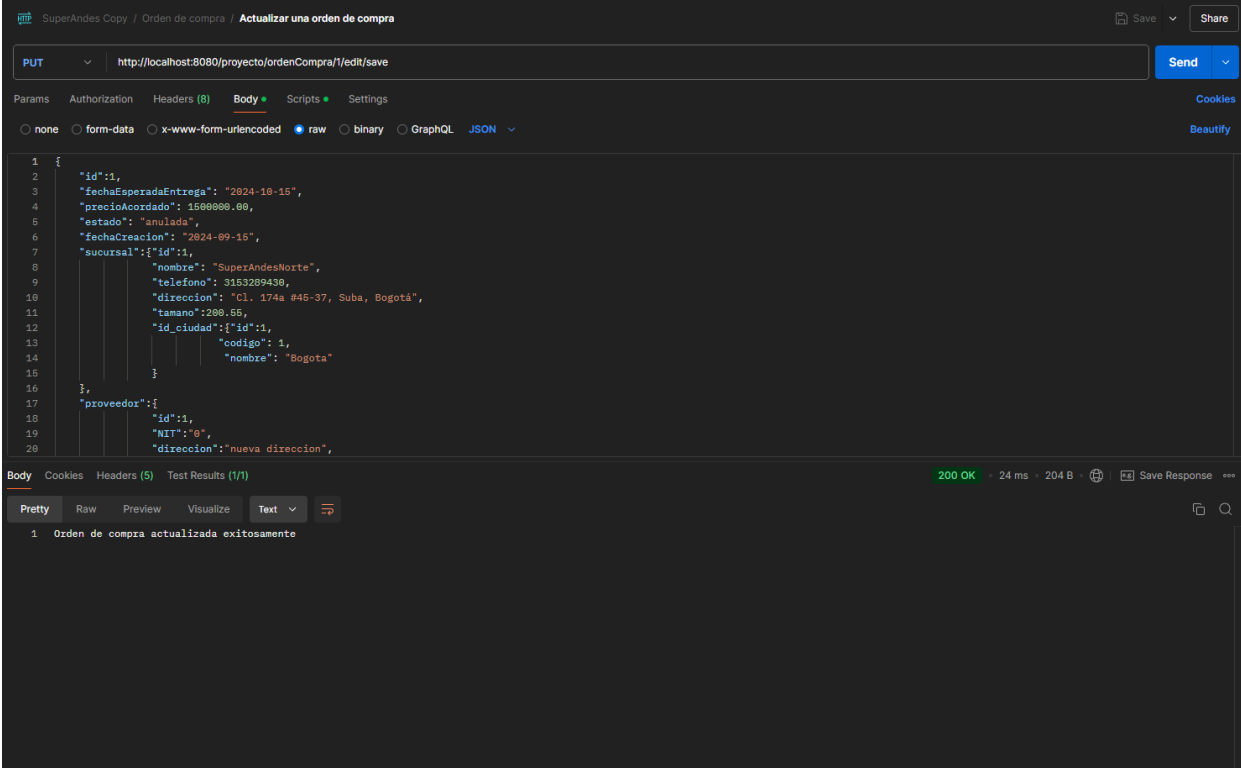
201 Created 28 ms 204 B Save Response

Pretty Raw Preview Visualize Text

1 Orden de compra creada exitosamente

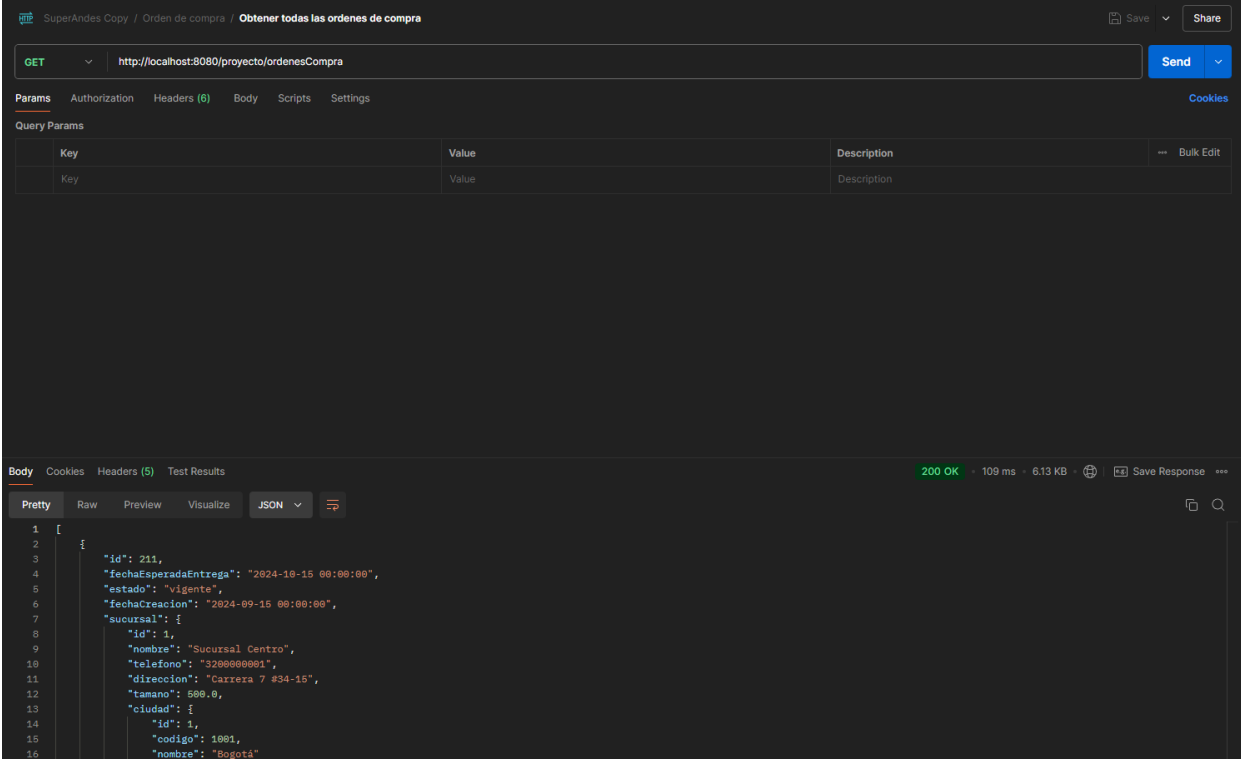
-Requerimiento funcional 8:

Actualizar una orden de compra cambiando su estado a anulada:



-Requerimiento funcional 9:

Mostrar todas las órdenes de compra:



-Requerimiento funcional 10:

Registrar ingreso de productos a bodega:

Para este requerimiento hicimos dos pruebas, una es recepción hoy la cual prueba una parte importante del requerimiento (principalmente para detectar errores y verificar la respuesta que da) y el otro es el requerimiento completo y se llama registrar productos en bodega.

Recepción hoy:

SuperAndes Copy 2 / Recepcion / **Recepcion Hoy**

GET

http://localhost:8080/proyecto/recepciones/hoy?bodega_id=1&proveedor_id=1&fechaString=2024-11-3

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	⋮	Bulk Edit
<input checked="" type="checkbox"/>	bodega_id	1			
<input checked="" type="checkbox"/>	proveedor_id	1			
<input checked="" type="checkbox"/>	fechaString	2024-11-3			

Body

Cookies

Headers (5)

Test Results

200 OK · 67 ms · 571 B · Save Response ⋮

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": 22,
3   "id_bodega": {
4     "id": 1,
5     "nombre": "Bodega Central",
6     "tamano": 1000.0,
7     "sucursal": {
8       "id": 1,
9       "nombre": "Sucursal Centro",
10      "telefono": "3200000001",
11      "direccion": "Carrera 7 #34-16",
12      "tamano": 900.0,
13      "ciudad": {
14        "id": 1,
15        "codigo": 1001,
16        "nombre": "Bogotá"
```

Registrar producto en bodega:

SuperAndes Copy 2 / Bodega / Registrar producto en bodega

GET `http://localhost:8080/proyecto/bodegas/registroProductoBodega?orden_compra_id=1&bodega_id=1` Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description
<input checked="" type="checkbox"/> orden_compra_id	1	
<input checked="" type="checkbox"/> bodega_id	1	

Body Cookies Headers (5) Test Results 200 OK · 373 ms · 206 B Save Response

Pretty Raw Preview Visualize Text

```
1 Producto registrado en bodega exitosamente
```

Requerimientos funcionales de consulta:

-Requerimiento funcional de consulta 1:

Mostrar índice de ocupación de cada una de las bodegas de una sucursal:

SuperAndes Copy / Sucursal / Índice de ocupación

GET `http://localhost:8080/proyecto/sucursales/consulta` Send

Params Authorization Headers (6) Body Scripts Settings Cookies Beautify

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

```
1 [4, 2, 3]
2
```

Body Cookies Headers (5) Test Results 200 OK · 103 ms · 192 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "porcentajeOcupacion": 66.6
3 }
```

-Requerimiento funcional de consulta 2:
Mostrar los productos que cumplen con cierta característica:

SuperAndes Copy / Producto Consultar producto por característica

GET http://localhost:8080/proyecto/productos/consultaCaracteristica?precioMinimo=10&precioMaximo=30000&fechaInicio=2001-01-01&fechaFin=2040-01-01&idCategoria=1

Params Authorization Headers (6) Body Scripts Settings

Query Params

<input checked="" type="checkbox"/> Key	Value	Description
<input checked="" type="checkbox"/> precioMinimo	10	
<input checked="" type="checkbox"/> precioMaximo	30000	
<input checked="" type="checkbox"/> fechaInicio	2001-01-01	
<input checked="" type="checkbox"/> fechaFin	2040-01-01	
<input checked="" type="checkbox"/> idCategoria	1	
Key	Value	Description

Body Cookies Headers (5) Test Results

200 OK · 51 ms · 466 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "costo_bodega": 4200.0,
3   "peso_empaque": 15.0,
4   "cantidad_presentacion": 1000.0,
5   "unidad_medida": 0,
6   "presentacion": "paquetón de 5 paquetes de 200 gr",
7   "fecha_expiration": "2021-02-01 00:00:00.0",
8   "precio_venta": 9000.0,
9   "categoria_id": 1,
10  "nombre": "Papas fritas",
11  "codigo_barras": "4ASADf0f0f",
12  "volumen_empaque": 150.0
13 }
```

-Requerimiento funcional de consulta 3:
Inventario de productos en bodega:

SuperAndes Copy 2 / Bodega / Consulta productos de bodega

GET http://localhost:8080/proyecto/bodegas/consultaBodega?bodega_id=2&sucursal_id=2 Send

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description
<input checked="" type="checkbox"/>	bodega_id	2	
<input checked="" type="checkbox"/>	sucursal_id	2	
	Key	Value	Description

Body Cookies Headers (5) Test Results 200 OK · 273 ms · 257 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "nivel_minimo": 30,
3   "costo_promedio": 8800.0,
4   "producto": "Carne de Res",
5   "cantidad_existente": 60
6 }
```

-Requerimiento funcional de consulta 4:
Mostrar sucursales en las que hay disponibilidad de un producto:

SuperAndes Copy / Bodega / Consulta de sucursal producto

GET http://localhost:8080/proyecto/bodegas/consultarSucursalProducto?producto_id=2 Send

Params • Authorization Headers (6) Body Scripts Settings Cookies

Query Params

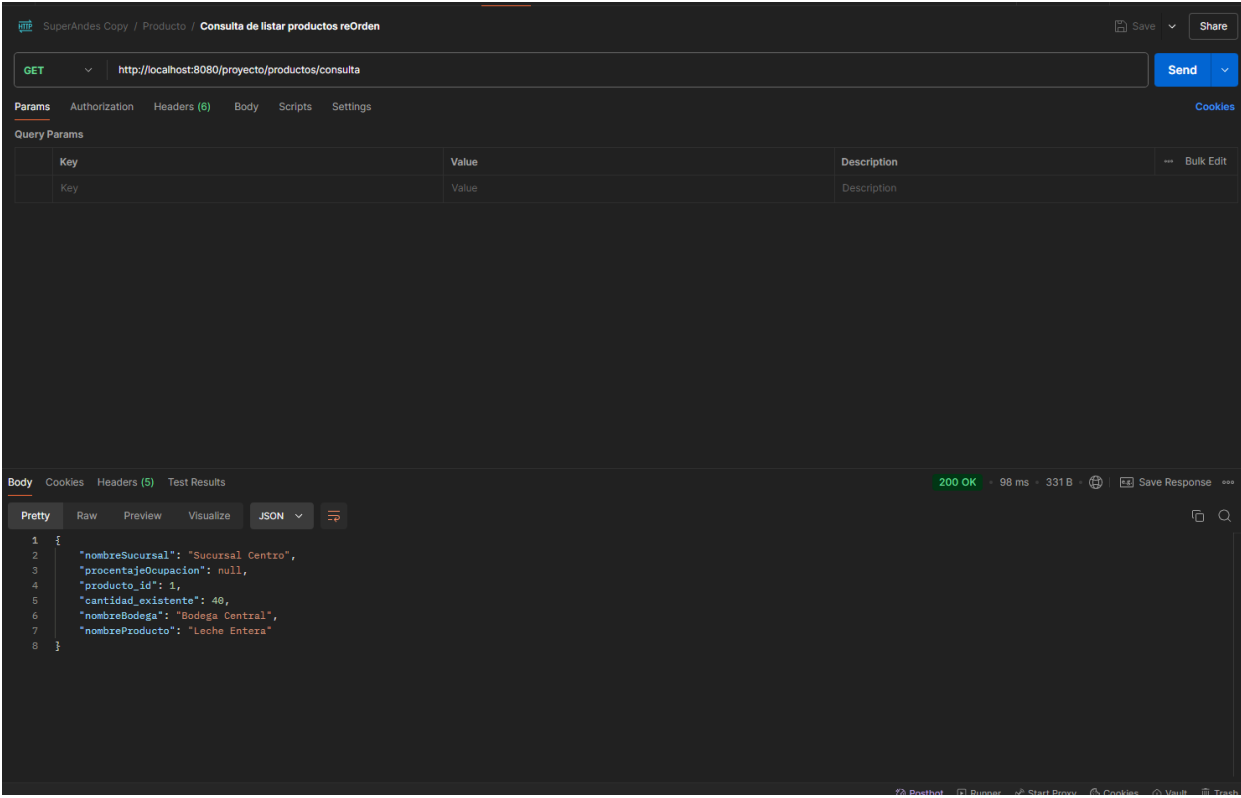
<input checked="" type="checkbox"/>	Key	Value	Description
<input checked="" type="checkbox"/>	producto_id	2	
	Key	Value	Description

Body Cookies Headers (5) Test Results 200 OK · 33 ms · 261 B Save Response

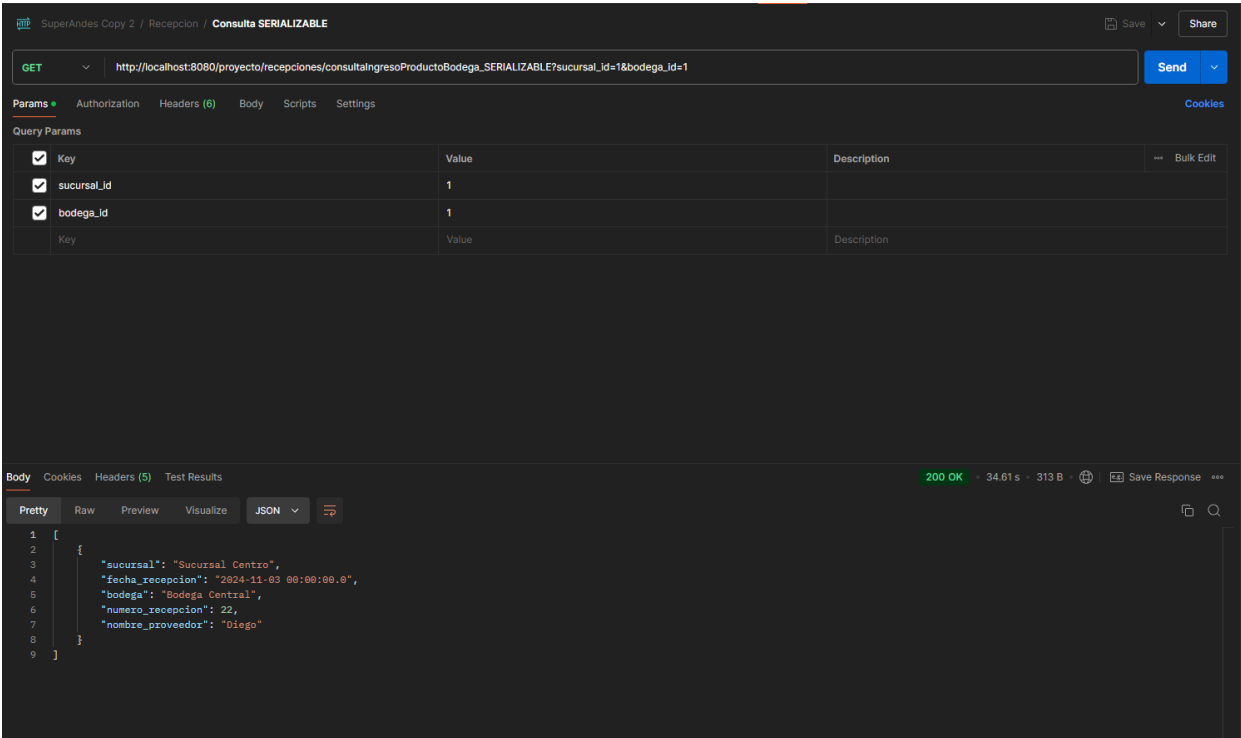
Pretty Raw Preview Visualize JSON

```
1 {
2   "direccion": "Avenida 19 #153-30",
3   "telefono": "3100000002",
4   "nombre": "Bodega Norte",
5   "tamano": 800.0
6 }
```

-Requerimiento funcional de consulta 5:
Mostrar todos los productos que requieren una orden de compra:



-Requerimiento funcional de consulta 6:
Consulta de documentos de ingreso de productos a bodega - Serializable:



-Requerimiento funcional de consulta 7:

Consulta de documentos de ingreso de productos a bodega – Read Committed:

SuperAndes Copy 2 / Recepcion / Consulta READ_COMMITTED

GET

http://localhost:8080/proyecto/recepciones/consultaIngresoProductoBodega_READ_COMMITTED?bodega_id=1&sucursal_id=1

Send

Params

AuthorizationHeaders (6)BodyScriptsSettings

Query Params

Key	Value	Description
bodega_id	1	
sucursal_id	1	

Body

CookiesHeaders (5)Test Results

200 OK · 30.04 s · 313 B · Save Response

PrettyRawPreviewVisualizeJSON

```
1 [
2   {
3     "bodega": "Bodega Central",
4     "fecha_recepcion": "2024-11-03 00:00:00.0",
5     "nombre_proveedor": "Diego",
6     "numero_recepcion": 22,
7     "sucursal": "Sucursal Centro"
8   }
9 ]
```

3. Escenarios de prueba

```
1 row inserted.

Error starting at line : 8 in command -
INSERT INTO proveedores (id, NIT, direccion, nombrePersonaC, telefonoPersonaC)
VALUES (1, '987654321', 'Av. Secundaria 200', 'Maria Garcia', '555-5678')
Error report -
ORA-00001: unique constraint (ISIS2304A01202420.SYS_C001329426) violated

1 row inserted.

Error starting at line : 16 in command -
INSERT INTO productos (id, codigoBarras, nombre, costoBodega, precioVenta, presentacion, cantidadPresentacion, unidadMedida, volumenEmpaque, pesoEmpaque)
VALUES (1, 'a0a0a0a0a0', 'Cereal Chocoflakes', 3000, 5000, 'Caja', 500, 1, 100, 10, TO_DATE('2026-12-12', 'YYYY-MM-DD'), 2)
Error report -
ORA-00001: unique constraint (ISIS2304A01202420.SYS_C001329441) violated

1 row inserted.
```

```

1 row inserted.

1 row inserted.

1 row inserted.

Error starting at line : 31 in command -
INSERT INTO bodegas (nombre, tamano, id_sucursal)
VALUES ('Bodega Secundaria', 150, 999)
Error report -
ORA-02291: integrity constraint (ISIS2304A01202420.FK_BODEGA_SUCURSAL) violated - parent key not found

1 row inserted.

1 row inserted.

1 row inserted.

Error starting at line : 50 in command -
INSERT INTO info_recepcion (id_recepcion, id_producto, cantidadRecibida, costoRecibido)
VALUES (1, 1, 10, 9000)
Error report -
ORA-00001: unique constraint (ISIS2304A01202420.PK_INFO_RECEPCION) violated

1 row inserted.

1 row inserted.

1 row inserted.

1 row updated.

1 row inserted.

```

4. ESCENARIO DE PRUEBA DE CONCURRENCIA 2:

1. Pasos para la Ejecución Concurrente de RFC7 y RF10 a través de la Línea de Tiempo

- **Paso 1:** Se inició RFC7 con el nivel de aislamiento **READ COMMITTED**, lo cual comenzó una consulta de documentos de ingreso de productos a bodega, incluyendo un temporizador de 30 segundos.
- **Paso 2:** Mientras RFC7 estaba en ejecución, se inició RF10 para registrar un nuevo ingreso de producto en la bodega.
- **Paso 3:** RF10 finalizó rápidamente y confirmó la transacción de inserción, mientras RFC7 aún estaba en el proceso de espera.
- **Paso 4:** Al finalizar los 30 segundos, RFC7 completó su consulta y devolvió los documentos de ingreso de los últimos 30 días.

2. Descripción de lo Sucedido

- **¿RF10 Debíó Esperar a RFC7?**

No, RF10 no tuvo que esperar a que terminara RFC7. Dado que el nivel de aislamiento de RFC7 es **READ COMMITTED**, no bloquea la escritura concurrente de RF10. Esto significa que RF10 pudo registrar el ingreso del producto de manera simultánea sin interferir con la consulta en curso de RFC7.

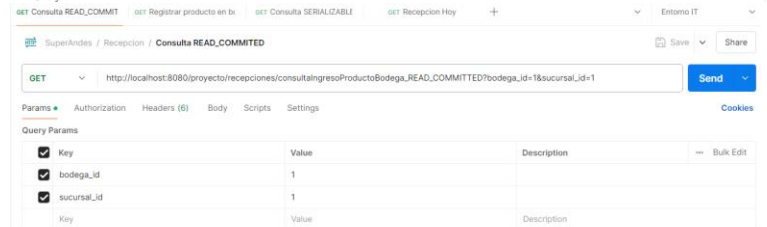
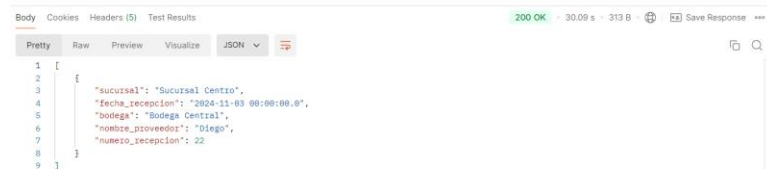
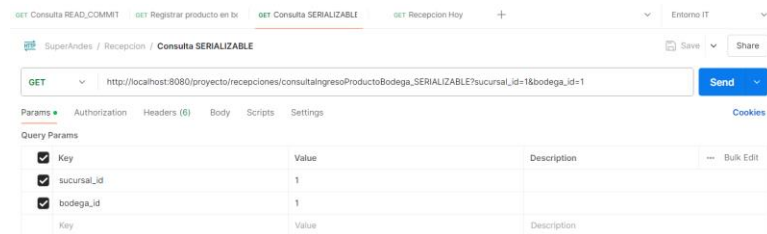
3. Resultado Presentado por RFC7

- **¿Apareció el Nuevo Documento de Ingreso?**

Sí, el nuevo documento de ingreso creado por RF10 apareció en los resultados de la consulta de RFC7. Esto se debe a que RFC7, con el nivel de aislamiento **READ COMMITTED**, puede ver las transacciones confirmadas durante su ejecución. Al completar el temporizador de 30 segundos, RFC7 incluyó el nuevo documento de ingreso registrado por RF10 en sus resultados.

Conclusión

El nivel de aislamiento **READ COMMITTED** permitió que la consulta de RFC7 incluyera el documento de ingreso de productos realizado por RF10 de manera simultánea, ya que la transacción de RF10 se confirmó antes de que finalizara la ejecución de RFC7. Esto demuestra que **READ COMMITTED** permite ver los datos confirmados sin bloquear las transacciones de escritura concurrentes.



GET Consulta READ_COMMIT

GET Registrar producto en b

GET Consulta SERIALIZABLE

GET Recepcion Hoy

+

Entorno IT

SuperAndes / Recepcion / Consulta SERIALIZABLE

SaveShare

GEThttp://localhost:8080/proyecto/recepciones/consultalingresoProductoBodega_SERIALIZABLE?sucursal_id=1&bodega_id=1Send

ParamsAuthorizationHeaders (6)BodyScriptsSettingsCookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> sucursal_id	1		
<input checked="" type="checkbox"/> bodega_id	1		
Key	Value	Description	

BodyCookiesHeaders (5)Test Results200 OK · 30.09 s · 313 B · Save Response

PrettyRawPreviewVisualizeJSON

```
1 [
2   {
3     "sucursal": "Sucursal Centro",
4     "fecha_recepcion": "2024-11-03 00:00:00.0",
5     "bodega": "Bodega Central",
6     "nombre_proveedor": "Diego",
7     "numero_recepcion": 22
8   }
9 ]
```

PostbotCtrlAltP