

SISTEMAS TRANSACCIONALES

Informe

Entrega 2 – Corrección de requerimientos entrega 1

Palacio Flórez, Alberto

Parra Réspedes, Pablo Andrés Sebastián

Hernández Paez, Juan Felipe

Universidad de Los Andes

Bogotá

03/11/2024

Tabla de contenido

Modelo conceptual: 3

Pruebas Entrega 1: 4

 RF1 – Crear una ciudad:..... 4

 RF2 – Crear una sucursal: 4

 RF3 – Crear y borrar una bodega: 5

 RF4 – Crear y actualizar proveedores: 6

 RF5 – Crear y leer una categoría de producto: 7

 RF6 – Crear, leer y actualizar un producto: 8

 RF7 – Crear una orden de compra: 9

 RF8 – Actualizar una orden de compra cambiado su estado a anulada: 10

 RF9 – Mostrar las órdenes de compra:..... 10

 RFC1 – Mostrar el índice de ocupación de cada una de las bodegas de una sucursal: 11

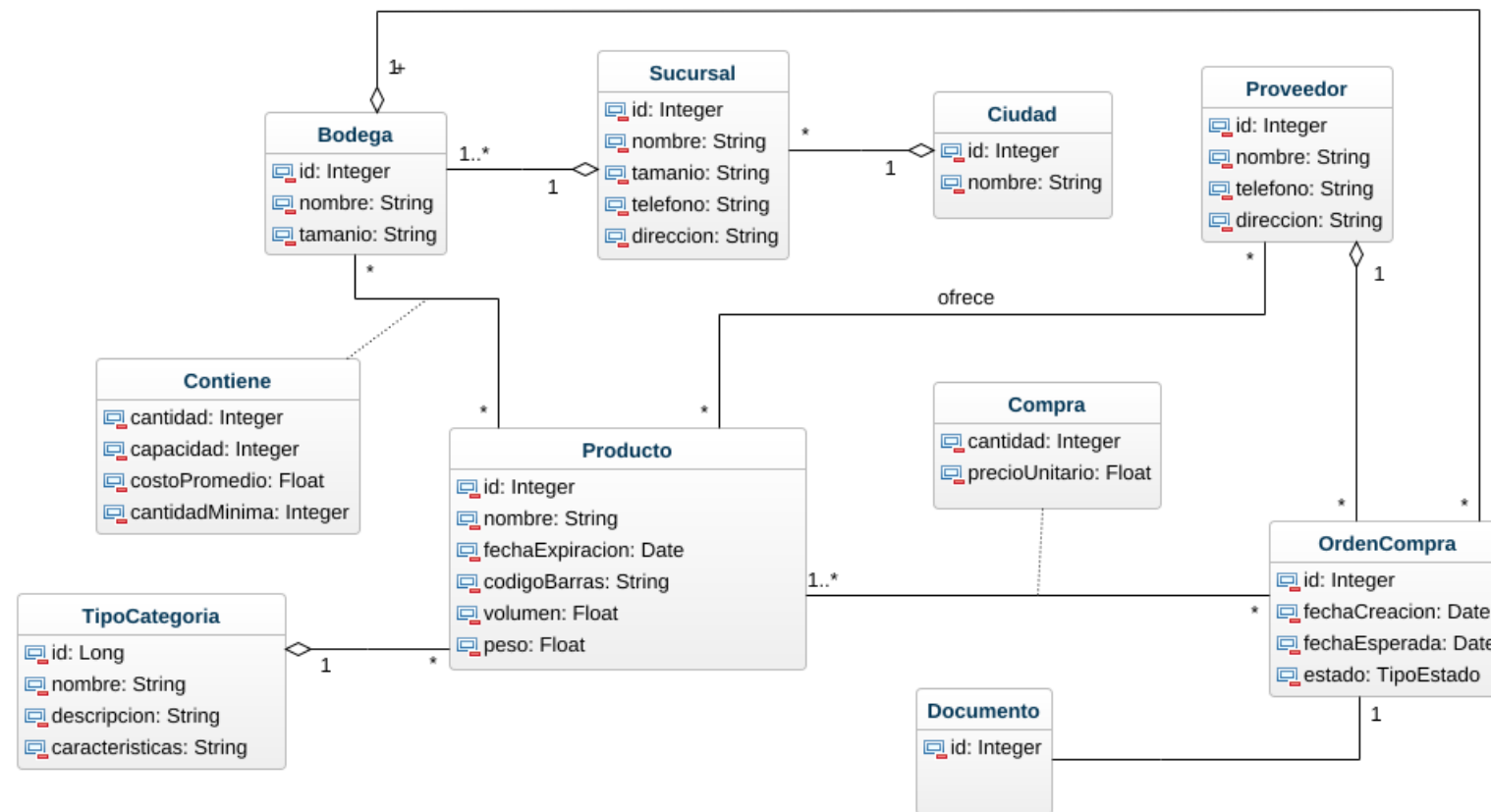
 RFC2 – Mostrar todos los productos que cumplen con cierta característica: 11

 RFC3 – Mostrar el inventario de productos en una bodega: 13

 RFC4 - Mostar las sucursales en las que está disponible un producto: 13

 RFC5 – Mostar los productos que requieren una orden de compra: 14

Modelo conceptual:



Se actualizó el modelo conceptual debido a restricciones de negocio y comentario recibidos por el monitor en la sustentación, algunos de estos cambios fueron:

- Los atributos con nombre tamaño en las clases Bodega y Sucursal ahora son Strings.
- El atributo codigoBarras en producto ahora es un String.

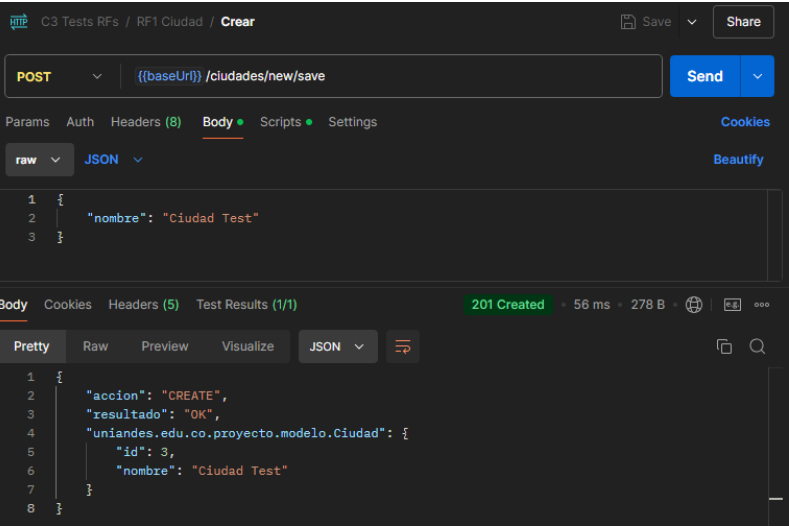
Nota: Otros cambios respecto a modelos anteriores se detallan en el documento de entrega 1 - Implementacion.

También se añadieron restricciones a la BD, en específico condiciones de unicidad, para más detalle consultar [/tablas_loop_delete.sql](#) en este archivo de sql, los atributos que se espera que no se repita su valor están marcados con unique.

Pruebas Entrega 1:

Para estos test, se recomienda abrir inicialmente el sql que se encuentra en `/sql/__restart__.sql` en sql developer con las credenciales indicadas al final del documento, y correr toda la hoja sql, esta hoja lo que hace es inicialmente borrar todas las tablas de la sesión, luego las crea de nuevo y después inserta los datos de prueba. Esta acción se puede resumir en eliminar filas creadas por los tests o otras sesiones, dejando únicamente las que se crearon como datos iniciales.

Comentado [JH1]: link



RF1 – Crear una ciudad:

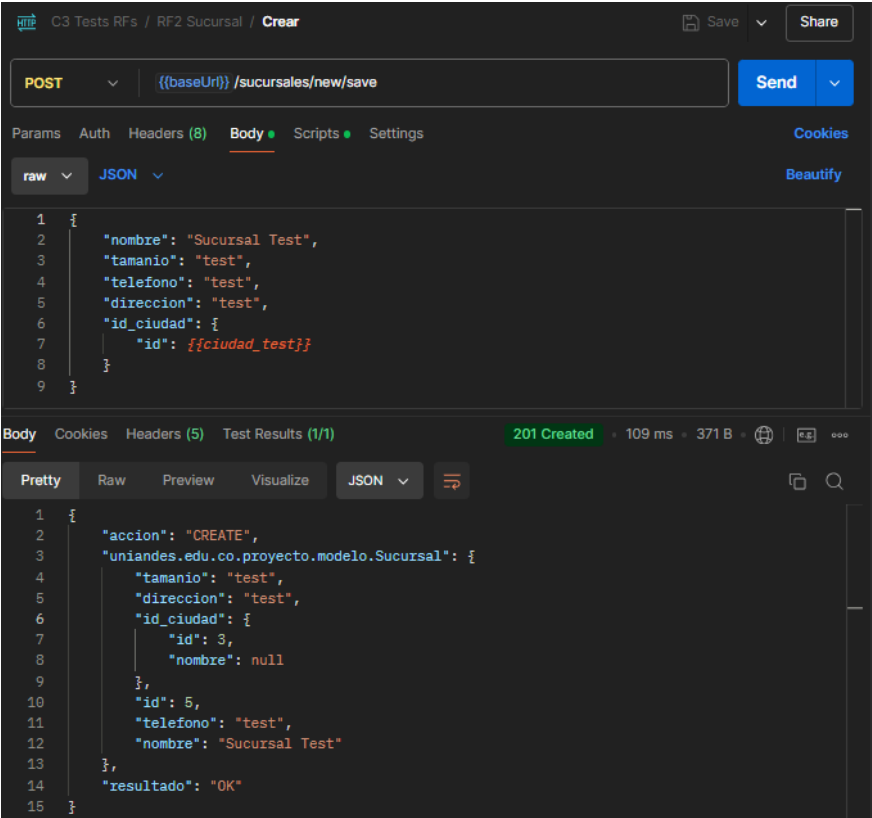
En la entrega 1 no compilaba el proyecto, ahora que, si compila, se corrigió la sentencia sql y el controller de ciudad para hacer posible el post (insertar):

La prueba es correcta si se recibe un Status code de 201. El id de la ciudad creada es guardado en la variable ciudad_test.

RF2 – Crear una sucursal:

En la entrega 1 no compilaba el proyecto, ahora que, si compila, se corrigió la sentencia sql y el controller de sucursal para hacer posible el post (insertar):

La prueba es correcta si se recibe un Status code de 201. El id de la sucursal creada es guardado en la variable sucursal_test. Note que como una sucursal tiene estrictamente una ciudad, se utilizó la variable ciudad_test para indicar este id.

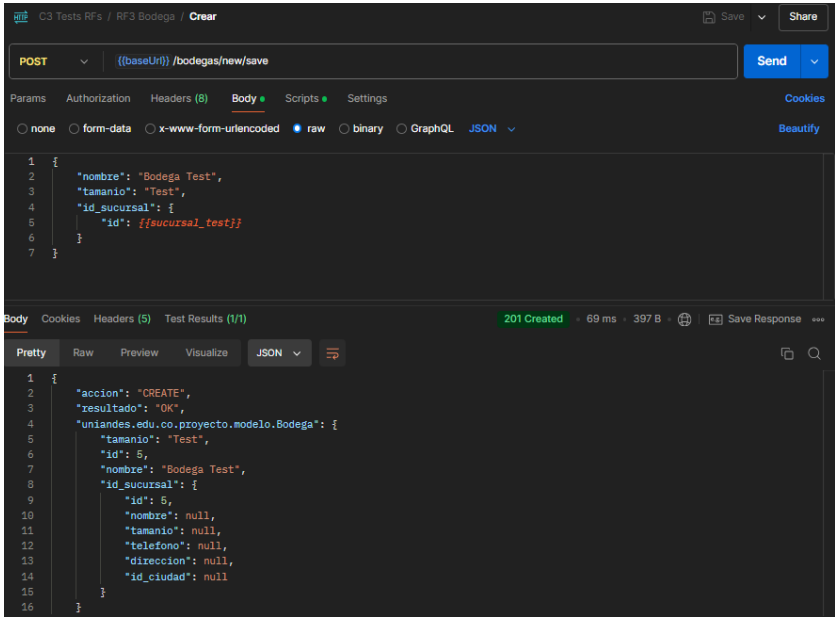


RF3 – Crear y borrar una bodega:

En la entrega 1 no compilaba el proyecto, ahora que, si compila, se corrigió la sentencia sql y el controller de bodega para hacer posible el post (insertar) y el get (eliminar), de la misma manera, se añadió la restricción:

```
ALTER TABLE ordenes_compra
  ADD CONSTRAINT orden_compra_bodega_fk FOREIGN KEY ( id_bodega )
  REFERENCES bodegas ( id )
  ON DELETE CASCADE;
```

Con el fin de no poder eliminar una bodega si esta se encuentra asociada a una orden de compra.

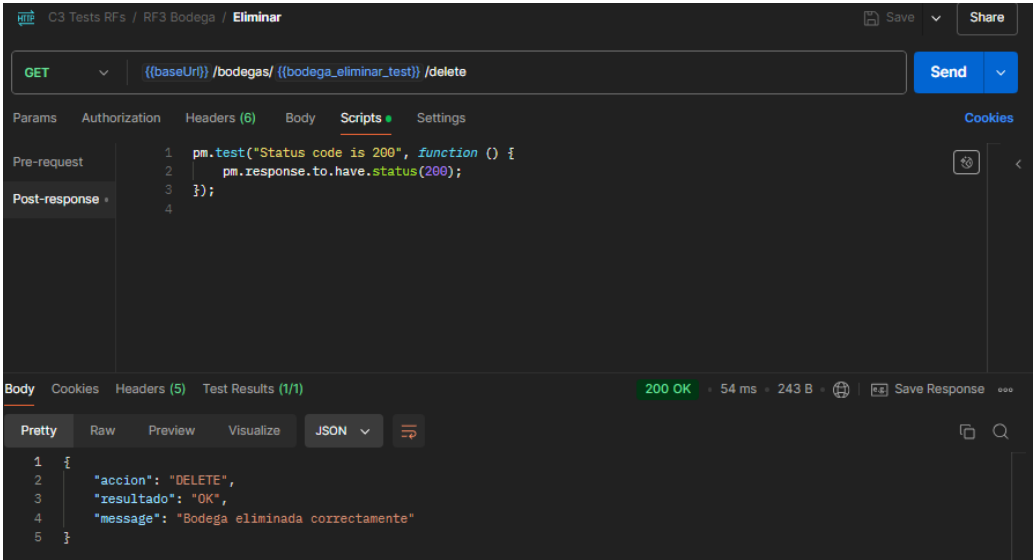


Crear una bodega:

La prueba es correcta si se recibe un Status code de 201. El id de la bodega creada es guardado en la variable bodega_test. Note que como una bodega tiene estrictamente una sucursal, se utilizó la variable sucursal_test para indicar este id.

Borrar una bodega:

La prueba es correcta si se recibe un Status code de 200. Note que se envía el id de la bodega que se quiere eliminar, si esta está asociada a una orden de compra o a un producto, esta no se podrá eliminar y se lanzará una excepción (resultado = not_ok).

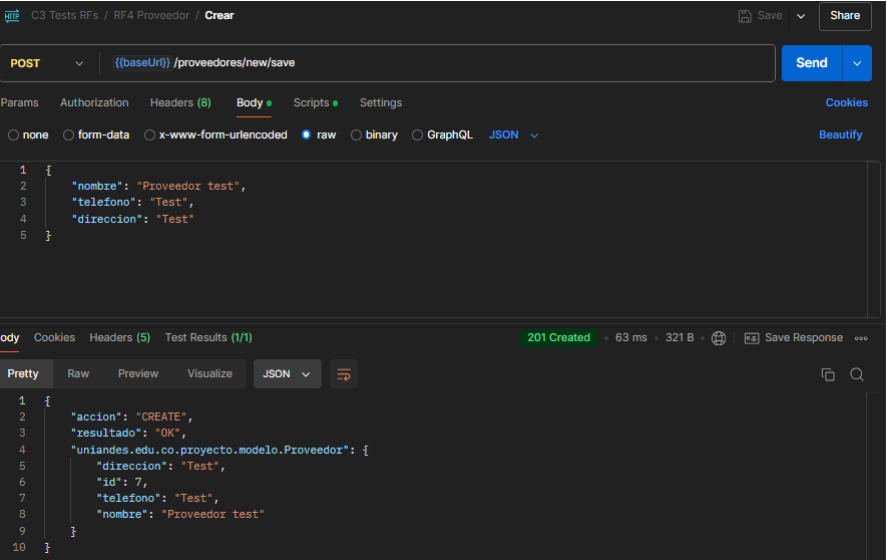
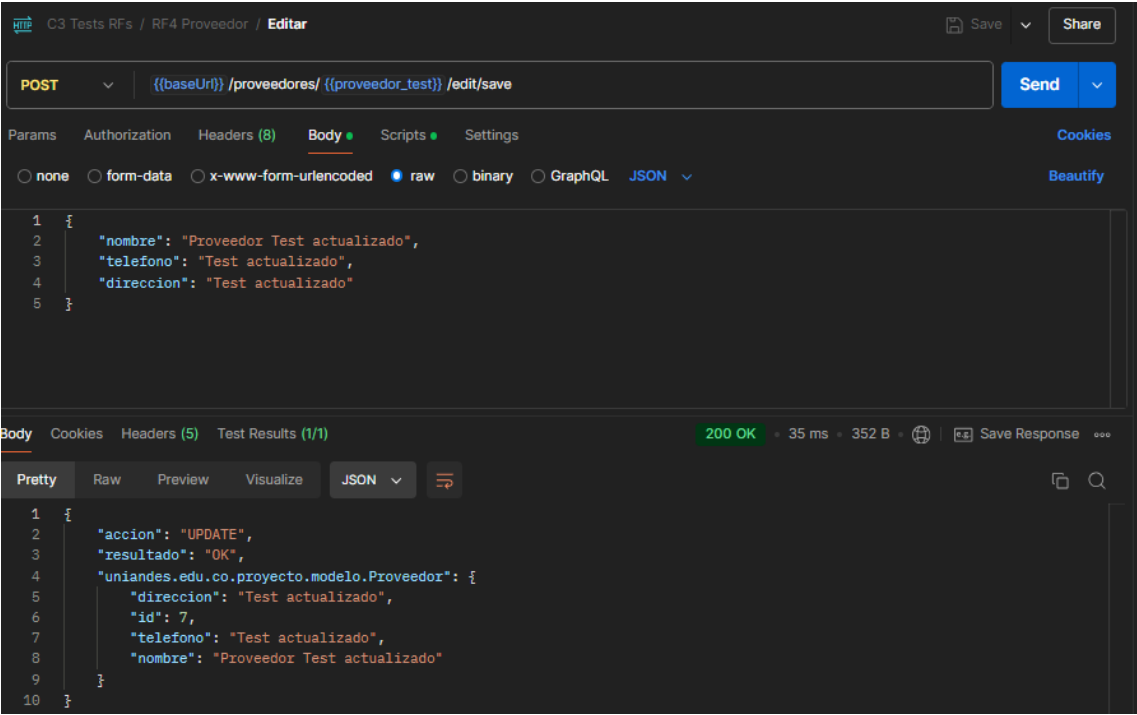


RF4 – Crear y actualizar proveedores:

En la entrega 1 no compilaba el proyecto, ahora que, si compila, se corrigió la sentencia sql y el controller de proveedor para hacer posible el post (insertar) y el post (actualizar):

Crear un proveedor:

La prueba es correcto si se recibe un Status code de 201. El id del proveedor creado es guardado en la variable proveedor_test.

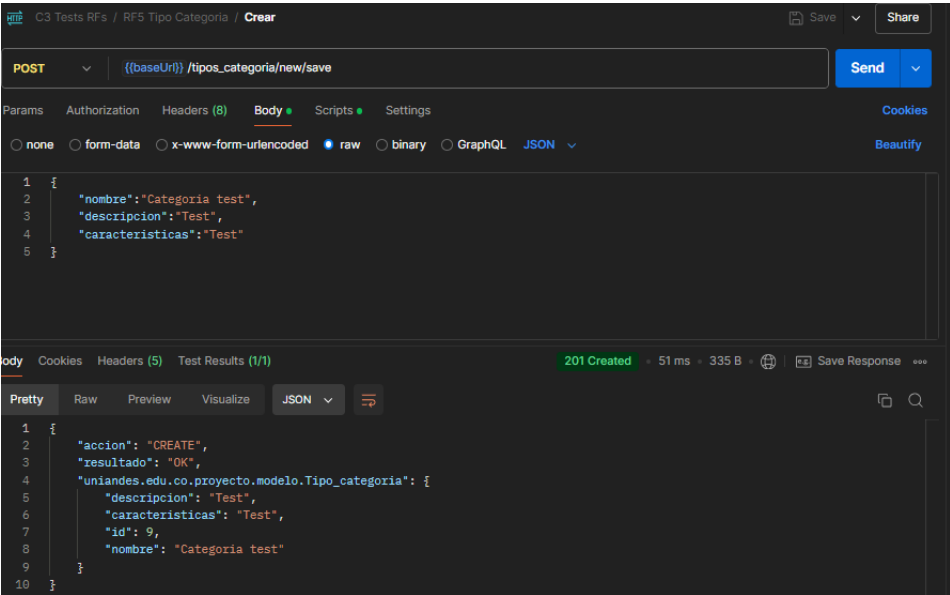


Actualizar un proveedor:

El test es correcto si se recibe un Status code de 200. Note que se envía el id del proveedor que se quiere mediante parámetro, y los datos actualizados mediante el body. De la misma manera, la tabla proveedores tiene la restricción de unicidad en los atributos nombre, teléfono y dirección, por lo que si esta (restricción) no se cumple, se lanzará una excepción directamente de la base de datos.

RF5 – Crear y leer una categoría de producto:

En la entrega 1 no compilaba el proyecto, ahora que, si compila, se corrigió la sentencia sql y el controller de tipo_categoria para hacer posible el post (insertar) y get (leer por id o nombre):

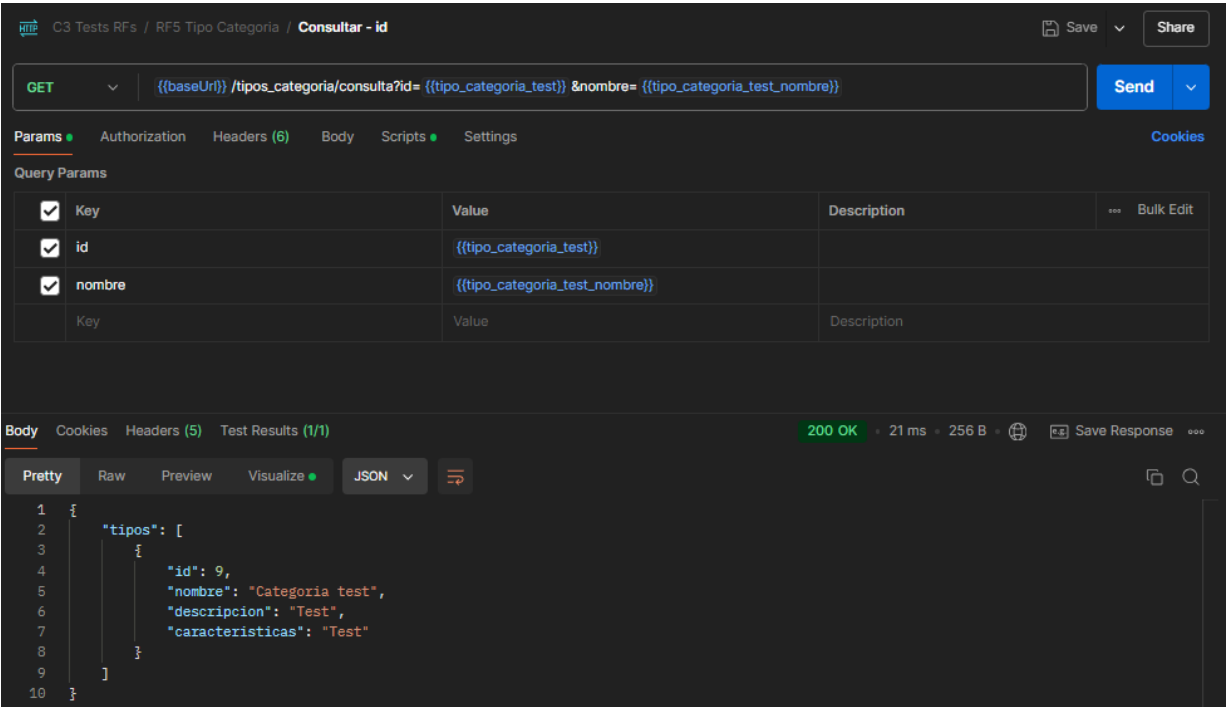


Crear categoría:

El test es correcto si se recibe un Status code de 201. El id del tipo de categoria creado es guardado en la variable tipo_categoria_test.

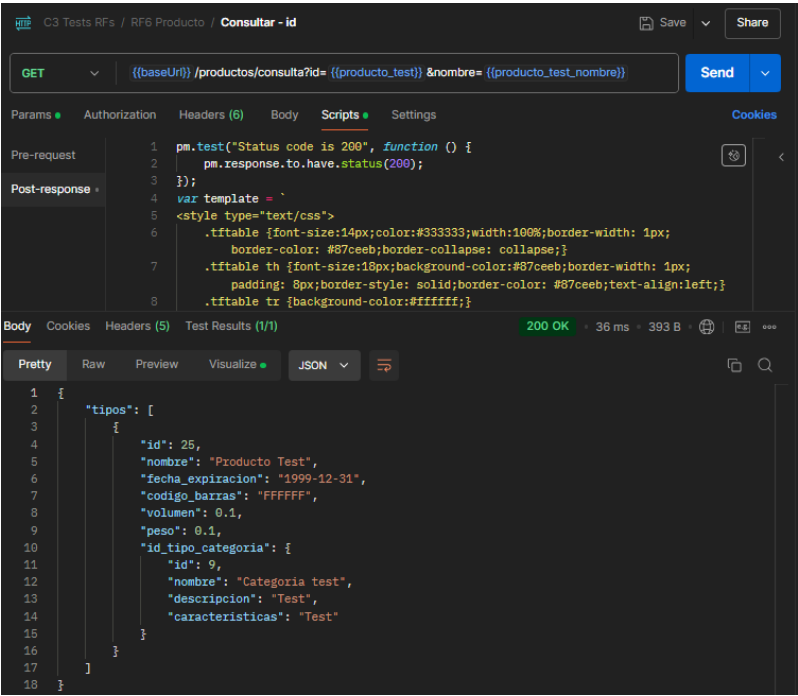
Leer una categoría por id o nombre:

El test es correcto si se recibe un Status code de 200. El controller (de tipo_categoria) está implementado para que, si solo se recibe un parámetro de los dos posibles (id o nombre), haga la búsqueda únicamente respecto a este parámetro. Hay otros tests de esta prueba en la collection.

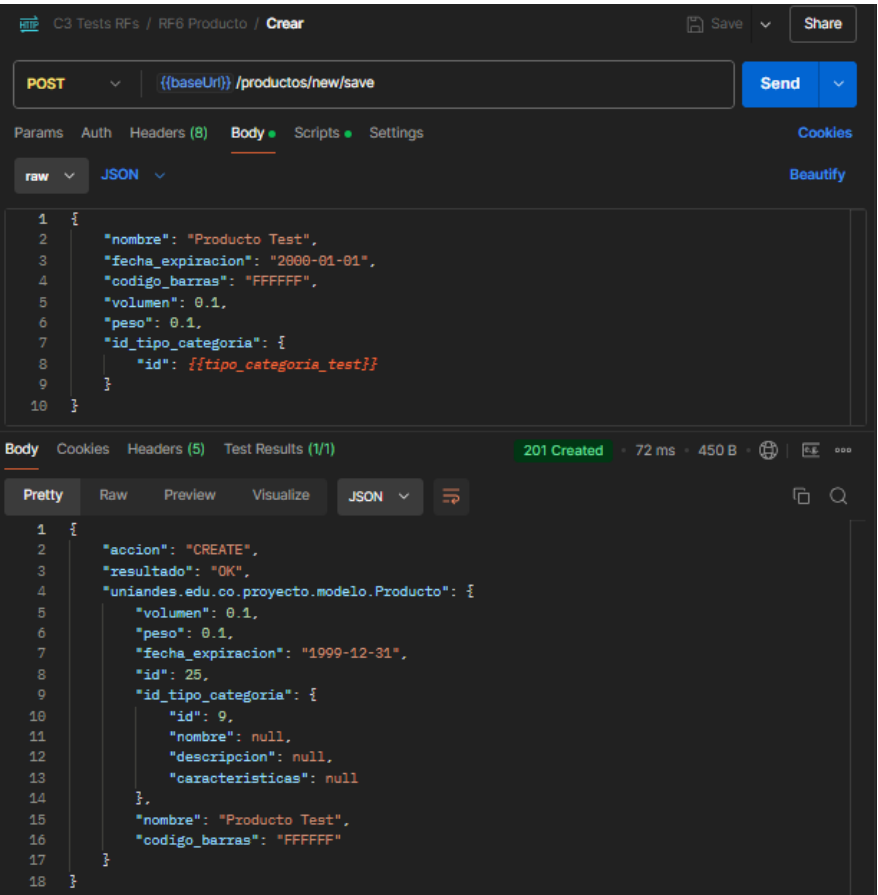


RF6 – Crear, leer y actualizar un producto:

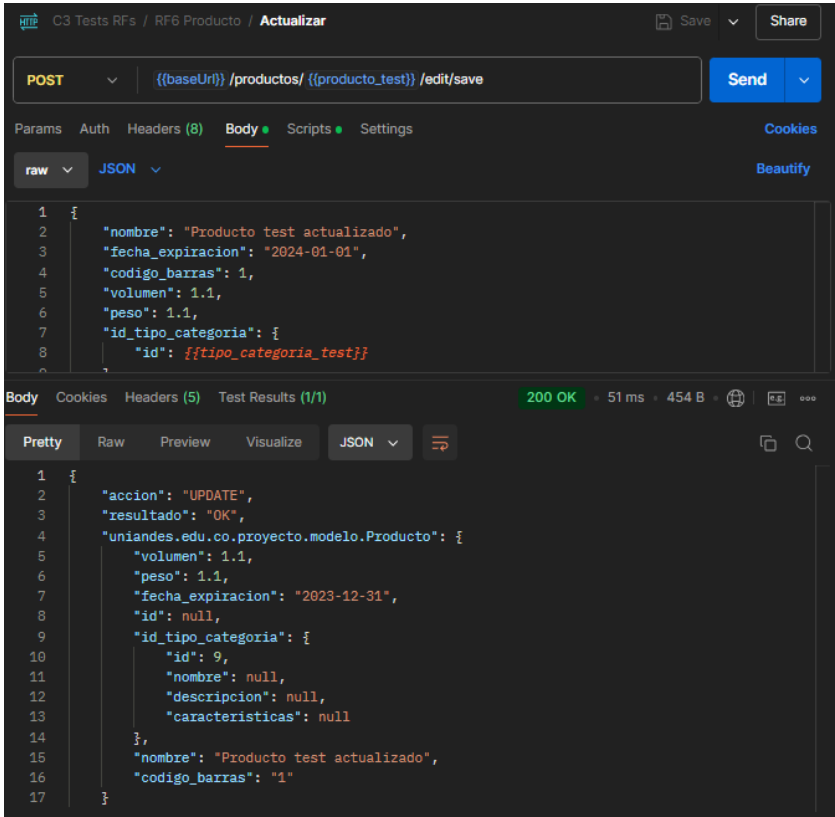
En la entrega 1 no compilaba el proyecto, ahora que, si compila, se corrigió la sentencia sql y el controller de producto para hacer posible el post (insertar), get (leer por id o nombre) y post (actualizar). Note que se cambió el tipo de dato de el atributo codigoBarras a String y se añadió la restricción de unicidad a este atributo (no se pueden repetir los códigos de barras):



Leer un producto por id o nombre



Crear producto



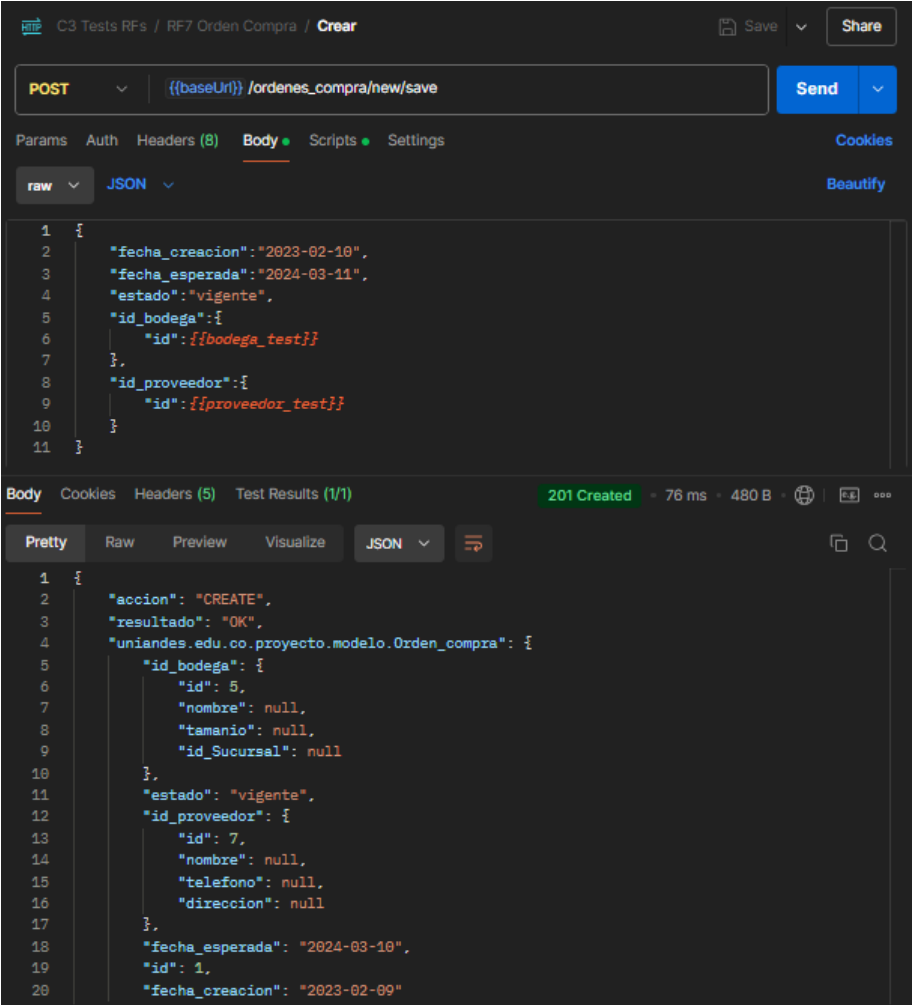
Actualizar un producto

RF7 – Crear una orden de compra:

En la entrega 1 no compilaba el proyecto, ahora que, si compila, se corrigió la sentencia sql y el controller de orden_compra para hacer posible el post (insertar). Note que una orden de compra tiene encabezado y detalle, los datos del encabezado son los mostrados en los atributos de la clase mientras que el detalle se hace mediante la tabla Detalle (ver el UML para entender mejor):

El test es correcto si se recibe un Status code de 201. Note que aquí se crea una orden de compra sin productos, para añadir productos, estos de deben añadir a la tabla Detalle la cual tiene el id de la orden de compra, el producto, la cantidad y el precio unitario, pero esto se hace mediante una operación aparte.

El test es correcto si se recibe un Status code de 200. Si se incumple una regla de negocio (Ej: el código de barras actualizado ya existe), se lanzará una excepción.



RF8 – Actualizar una orden de compra cambiado su estado a anulada:

orden	fecha creacion	fecha esperada	estado	bodega	proveedor	cantidad	precio unitario	producto
1	2023-02-10T00:00:00.000+00:00	2024-03-11T00:00:00.000+00:00	anulada	5	7	50	4500	1
1	2023-02-10T00:00:00.000+00:00	2024-03-11T00:00:00.000+00:00	anulada	5	7	30	8500	5
1	2023-02-10T00:00:00.000+00:00	2024-03-11T00:00:00.000+00:00	anulada	5	7	20	6800	8
2	2024-10-30T05:00:00.000+00:00	2024-11-25T05:00:00.000+00:00	vigente	3	6	100	3500	2
2	2024-10-30T05:00:00.000+00:00	2024-11-25T05:00:00.000+00:00	vigente	3	6	45	15000	10
3	2024-10-30T05:00:00.000+00:00	2024-11-25T05:00:00.000+00:00	vigente	3	6	75	180000	15
3	2024-10-30T05:00:00.000+00:00	2024-11-25T05:00:00.000+00:00	vigente	3	6	60	45000	20
3	2024-10-30T05:00:00.000+00:00	2024-11-25T05:00:00.000+00:00	vigente	3	6	25	150000	24
4	2024-10-30T05:00:00.000+00:00	2024-11-25T05:00:00.000+00:00	vigente	3	6	40	5500	3
4	2024-10-30T05:00:00.000+00:00	2024-11-25T05:00:00.000+00:00	vigente	3	6	35	22000	7
5	2024-10-15T05:00:00.000+00:00	2024-10-30T05:00:00.000+00:00	vigente	1	3	90	12000	12
5	2024-10-15T05:00:00.000+00:00	2024-10-30T05:00:00.000+00:00	vigente	1	3	65	1200000	18
6	2024-10-20T05:00:00.000+00:00	2024-11-05T05:00:00.000+00:00	vigente	2	1	55	7500	4
7	2024-10-25T05:00:00.000+00:00	2024-11-10T05:00:00.000+00:00	vigente	3	4	70	4200	6
7	2024-10-25T05:00:00.000+00:00	2024-11-10T05:00:00.000+00:00	vigente	3	4	40	18000	11
7	2024-10-25T05:00:00.000+00:00	2024-11-10T05:00:00.000+00:00	vigente	3	4	30	800000	16
8	2024-09-01T05:00:00.000+00:00	2024-09-15T05:00:00.000+00:00	entregada	1	2	85	15500	9
8	2024-09-01T05:00:00.000+00:00	2024-09-15T05:00:00.000+00:00	entregada	1	2	50	120000	14

C3 Tests RFs / RF8 Orden Compra / Actualizar - estado a anulada

POST{{baseUri}}/ordenes_compra/{{orden_compra_test}}/anularSend

ParamsAuthHeaders (7)BodyScriptsSettingsCookies

BodyCookiesHeaders (5)Test Results (1/1)200 OK · 98 ms · 665 B

PrettyRawPreviewVisualizeJSON

```
1 {
2   "accion": "UPDATE",
3   "resultado": "OK",
4   "uniandes.edu.co.proyecto.modelo.Orden_compra": {
5     "id_bodega": {
6       "id": 5,
7       "nombre": "Bodega Test",
8       "tamanio": "Test",
9       "id_Sucursal": {
10        "id": 5,
11        "nombre": "Sucursal Test",
12        "tamanio": "test",
13        "telefono": "test",
14        "direccion": "test",
15        "id_ciudad": {
16          "id": 3,
17          "nombre": "Ciudad Test"
18        }
19      }
20    },
21    "estado": "vigente",
22    "id_proveedor": {
23      "id": 7,
24      "nombre": "Proveedor Test actualizado",
25      "telefono": "Test actualizado",
26      "direccion": "Test actualizado"
27    },
28    "fecha_esperada": "2024-03-10",
29    "id": 1,
30    "fecha_creacion": "2023-02-09"
31  }
32 }
```

Comentado [JH2]: Preguntar monitor comentario entrega 1

RF9 – Mostrar las órdenes de compra:

En la entrega 1 no compilaba el proyecto, ahora que, si compila, se corrigió la sentencia sql y el controller de orden_compra para hacer posible el get (leer todos). Se modificó el sql para que ahora tambien se muestren los datos del detalle de las ordenes de compra:

La prueba es correcta si se recibe un Status code de 200. Note que los ids de las ordenes se repiten, esto se debe a que hay varios productos ligados a una misma orden.

RFC1 – Mostrar el índice de ocupación de cada una de las bodegas de una sucursal:

En la entrega 1 no se desarrolló este requerimiento. Ya se desarrolló y funciona bajo el controller de productos:

Este requerimiento recibe una lista de ids de productos por parámetro, y muestra en que bodega están y cual es su porcentaje de ocupación en esta. El test es correcto con Status code 200.

C3 Tests RFCs / RFC1 / **Mostrar el indice de ocupacion de productos de bodegas**

Save

Share

GET

{{baseUrl}} /productos/consulta?ids=1,4

Send

ParamsAuthHeaders (6)BodyScriptsSettingsCookies

BodyCookiesHeaders (5)Test Results (1/1)

200 OK24 ms215 B

PrettyRawPreviewVisualize

ID Bodega	ID Producto	Ocupación
1	1	0.5
1	4	0.5
2	4	0.5
3	4	0.5
4	4	0.5

C3 Tests RFCs / RFC2 / Rango de precio / **Mostrar**

Save

Share

GET

{{baseUrl}} /productos/consulta?minPrice=3000&maxPrice=10000

Send

ParamsAuthorizationHeaders (6)BodyScriptsSettingsCookies

BodyCookiesHeaders (5)Test Results (1/1)

200 OK31 ms850 B

PrettyRawPreviewVisualize

Bodega	ID	Nombre	Fecha de Expiración	Código de Barras	Volumen	Peso	Categoría	Precio
1 Bodega Norte Bucaramanga	1	Leche Entera	2024-12-31T05:00:00.000+00:00	PRD001	1	1.1	1 Perecederos	3500
2 Bodega Sur Bucaramanga	2	Yogurt Natural	2024-11-30T05:00:00.000+00:00	PRD002	0.5	0.6	1 Perecederos	4000
3 Bodega Occidente Bogota	3	Pan Integral	2024-10-15T05:00:00.000+00:00	PRD003	0.8	0.4	1 Perecederos	3000
4 Bodega Oriente Bogota	9	Limpiador Multiusos		PRD009	1.5	1.6	3 Aseo	8000
1 Bodega Norte Bucaramanga	10	Helado Vainilla	2025-06-30T05:00:00.000+00:00	PRD010	2	1	4 Congelados	8000
3 Bodega Occidente Bogota	12	Verduras Mixtas Congeladas	2025-04-30T05:00:00.000+00:00	PRD012	1	0.8	4 Congelados	6000

RFC2 – Mostrar todos los productos que cumplen con cierta característica:

En la entrega 1 no se desarrolló este requerimiento. Ya se desarrolló y funciona bajo el controller de productos, bajo este se pueden hacer filtros independientes por rango de precios, rango de fecha, sucursal o categoría:

GET{{baseUrl}} /productos/consulta?fechaPosteriorA=2025-01-01&fechaInferiorA=2027-01-01Send

ParamsAuthorizationHeaders (6)BodyScriptsSettingsCookies

BodyCookiesHeaders (5)Test Results (1/1)200 OK33 ms1.09 KBSave Response

PrettyRawPreviewVisualize

ID	Nombre	Fecha de Expiración	Código de Barras	Volumen	Peso	Tipo de Categoría
10	Helado Vainilla	2025-06-30	PRD010	2	1	4 Congelados
11	Pollo Congelado	2025-05-30	PRD011	1.5	1.2	4 Congelados
12	Verduras Mixtas Congeladas	2025-04-30	PRD012	1	0.8	4 Congelados

GET{{baseUrl}} /productos/consulta?id_sucursal=1Send

ParamsAuthorizationHeaders (6)BodyScriptsSettingsCookies

BodyCookiesHeaders (5)Test Results (1/1)200 OK48 ms547 BSave Response

PrettyRawPreviewVisualize

ID	Nombre	Fecha de Expiración	Código de Barras	Volumen	Peso	Categoría	ID Sucursal
1	Leche Entera	2024-12-31T05:00:00.000+00:00	PRD001	1	1.1	1	1
4	Arroz Premium		PRD004	1	1	2	1
7	Detergente Líquido		PRD007	2	2.2	3	1
10	Helado Vainilla	2025-06-30T05:00:00.000+00:00	PRD010	2	1	4	1
13	Camiseta Básica		PRD013	0.3	0.2	5	1
17	Silla Ergonómica		PRD017	50	8	6	1
21	Taladro Eléctrico		PRD021	2	2.5	7	1

GET{{baseUrl}} /productos/consulta?id_tipo_categoria=3Send

ParamsAuthorizationHeaders (6)BodyScriptsSettingsCookies

BodyCookiesHeaders (5)Test Results (1/1)200 OK49 ms1.06 KBSave Response

PrettyRawPreviewVisualize

ID	Nombre	Fecha de Expiración	Código de Barras	Volumen	Peso	Categoría (ID Nombre)
7	Detergente Líquido		PRD007	2	2.2	3 Aseo
8	Jabón de Baño		PRD008	0.2	0.15	3 Aseo
9	Limpiador Multiusos		PRD009	1.5	1.6	3 Aseo

Note que cada búsqueda es y debe ser independiente de las demás, por lo que, no puedo filtrar por 2 características a la vez, todos los test son marcados correctos con Status code 200.

RFC3 – Mostrar el inventario de productos en una bodega:

En la entrega 1 no compilaba el proyecto, ahora que, si compila, se corrigió la sentencia sql y el controller de producto:

C3 Tests RFCs / RFC3 / **Mostrar - pertenecientes a una bodega y sucursal**

GET

{{baseUrl}} /productos/consulta?id_sucursal=2&id_bodega=2

Send

Params Authorization Headers (6) Body Scripts Settings

Body Cookies Headers (5) Test Results (1/1) 200 OK · 28 ms · 428 B Save Response

Pretty Raw Preview Visualize

ID	Nombre	Cantidad Actual	Cantidad Mínima	Costo Promedio	ID Sucursal	ID Bodega
2	Yogurt Natural	80	15	4000	2	2
4	Arroz Premium	100	20	2800	2	2
7	Detergente Líquido	90	15	15000	2	2
11	Pollo Congelado	60	12	12000	2	2
14	Pantalón Jean	150	30	45000	2	2
18	Sofá 3 Puestos	8	2	850000	2	2
22	Licuada	25	5	120000	2	2

C3 Tests RFCs / RFC4 / **Mostrar - sucursales con un producto**

GET

{{baseUrl}} /sucursales/consulta?id_producto=7

Send

Params Authorization Headers (6) Body Scripts Settings

Body Cookies Headers (5) Test Results (1/1) 200 OK · 31 ms · 369 B Save Response

Pretty Raw Preview Visualize

ID	Nombre	Tamaño	Teléfono	Dirección	ID Ciudad	ID Producto	Nombre Producto
1	Sucursal Norte Bucaramanga	300 m²	6376001	Calle 45 Norte #23-45	2	7	Detergente Líquido
2	Sucursal Sur Bucaramanga	350 m²	6376002	Carrera 33 Sur #12-67	2	7	Detergente Líquido

RFC4 - Mostar las sucursales en las que está disponible un producto:

En la entrega 1 no compilaba el proyecto, ahora que, si compila, se corrigió la sentencia sql y el controller de sucursal:

RFC5 – Mostar los productos que requieren una orden de compra:

En la entrega 1 no se desarrolló este requerimiento. Ya se desarrolló y funciona bajo el controller de productos:

HTTP

C3 Tests RFCs / RFC5 / **Mostrar - requieren orden de compra**

Save

Share

GET

{{baseUri}} /productos/consulta/reqOrdenCompra

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Cookies

Body

Cookies

Headers (5)

Test Results (1/1)

200 OK · 21 ms · 233 B ·

Save Response

...

Pretty

Raw

Preview

Visualize

ID	Nombre	Bodega	Sucursal	Cantidad	Cantidad Mínima	Proveedor
11	Pollo Congelado	3	3	5	20	6
11	Pollo Congelado	3	3	5	20	1

Note que sale 2 veces el producto Pollo congelado ya que este es ofrecido por 2 proveedores.