

Ximena López – 202312848

Santiago Pineda - 202023262

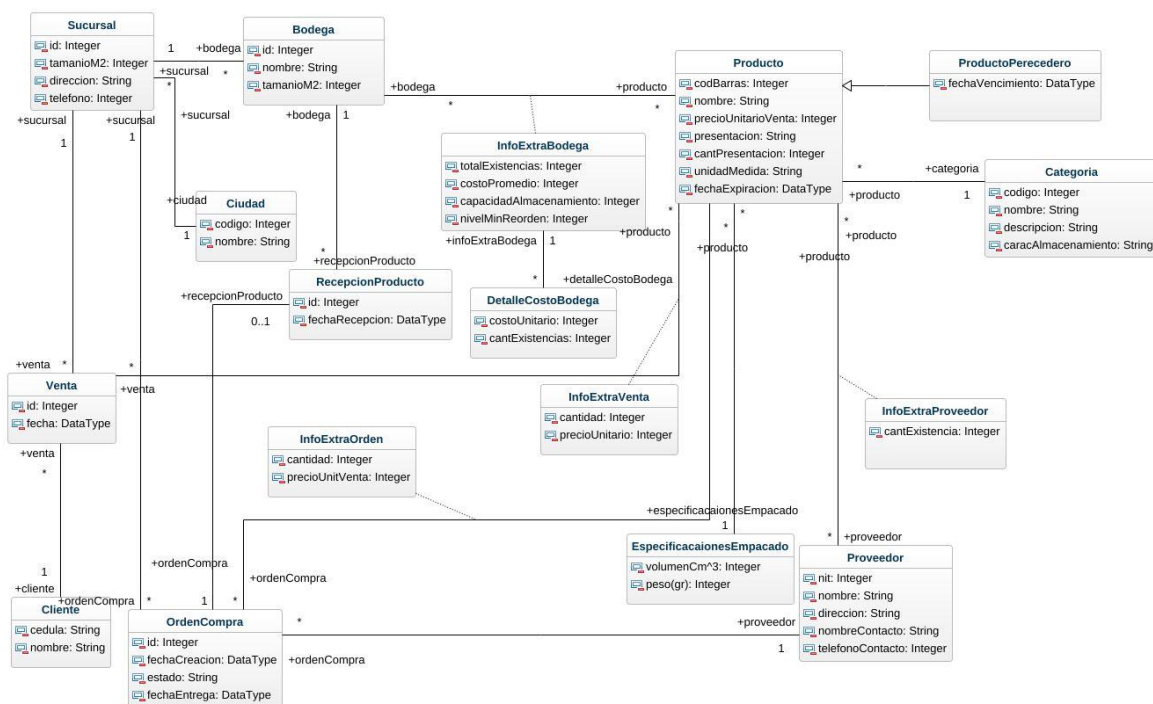
Sofia Losada - 202221008

Documentación Proyecto 1

Nombre de usuario Oracle:

ISIS2304A20202420

Diagrama UML actualizado:



Análisis UML:

- Descripción de los elementos clave:
 - Sucursal: Atributos: id, tamañoM2, dirección, teléfono.
 - Relaciones:
 - Bodega: Una sucursal puede tener múltiples bodegas (relación 1 a *), pero una bodega pertenece a una única sucursal (relación * a 1).
 - Ciudad: Una sucursal está ubicada en una ciudad (relación 1 a 1), pero una ciudad puede tener múltiples sucursales (relación * a 1).

- Bodega: Atributos: id, nombre, tamañoM2.
 - Relaciones:
 - Sucursal: Una bodega pertenece a una única sucursal (relación * a 1).
 - Producto: Una bodega puede almacenar múltiples productos (relación * a *), y cada producto puede estar presente en varias bodegas.
 - InfoExtraBodega: Relaciona una bodega con información adicional de productos específicos en dicha bodega (relación 1 a *).

- Producto: Atributos: codBarras, nombre, precioUnitarioVenta, presentación, cantidadPresentación, unidadMedida, fechaExpiración.
 - Relaciones:
 - InfoExtraBodega: Un producto puede estar almacenado en múltiples bodegas con diferentes informaciones extras, como existencias y costos (relación * a 1 con respecto a InfoExtraBodega).
 - InfoExtraProveedor: Un producto puede ser provisto por múltiples proveedores con diferentes cantidades de existencias (relación * a 1).
 - RecepcionProducto: Un producto puede estar relacionado con múltiples recepciones de productos (relación * a 1), pero una recepción está relacionada con un único producto (relación 0..1 a *).

- Producto Perecedero: Subclase de Producto, agregando el atributo fechaVencimiento.
 - Relaciones: Hereda todas las relaciones de Producto.

- Recepción de Productos: Atributos: id, fechaRecepción.
 - Relaciones:
 - Producto: Relación 0..1 a * (un producto puede tener múltiples recepciones, y una recepción puede no tener productos asociados o puede tener un solo producto).
 - DetalleCostoBodega: Múltiples productos recibidos pueden estar asociados a un único registro de costos (relación * a *).

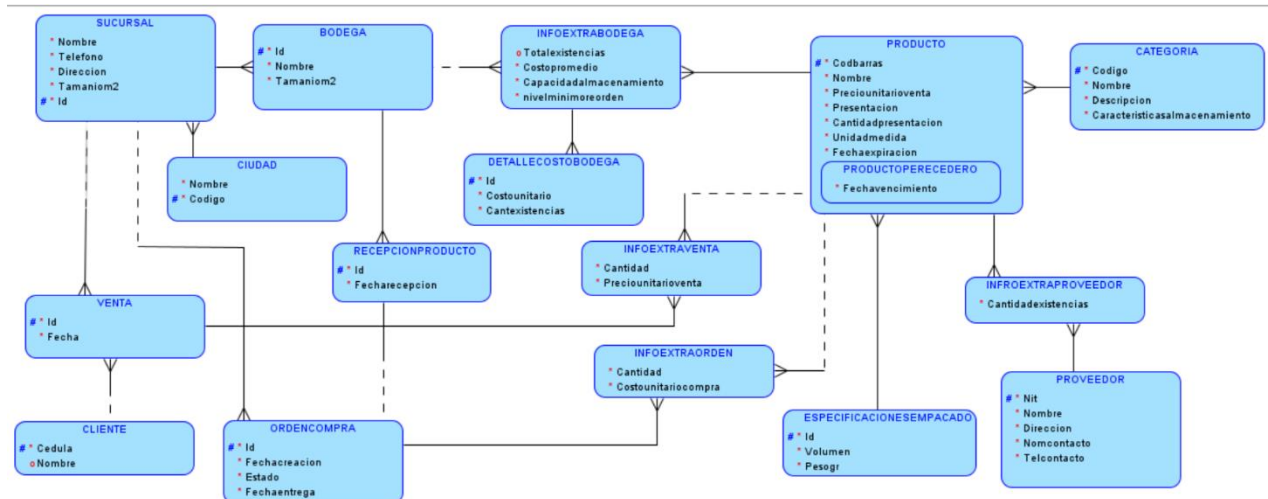
- Cliente: Atributos: cédula, nombre.
 - Relaciones:
 - Venta: Un cliente puede realizar múltiples ventas (relación 1 a *), pero una venta está relacionada con un único cliente (relación * a 1).
 - OrdenCompra: Un cliente puede hacer múltiples órdenes de compra (relación 1 a *), pero una orden de compra está asociada a un solo cliente (relación * a 1).

- Orden de Compra: Atributos: id, fechaCreación, estado, fechaEntrega.
 - Relaciones:
 - Cliente: Relación * a 1 (múltiples órdenes para un cliente).

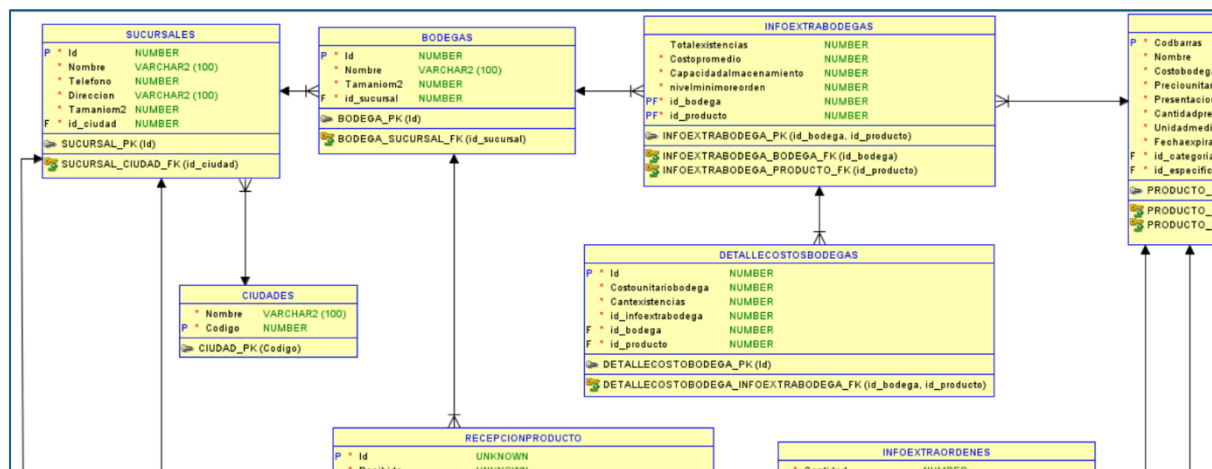
- InfoExtraOrden: Cada orden de compra puede tener múltiples detalles asociados (relación 1 a *).
- Venta: Atributos: id, fecha.
 - Relaciones:
 - Cliente: Relación * a 1 (múltiples ventas pueden estar asociadas a un cliente).
 - InfoExtraVenta: Una venta puede tener múltiples detalles adicionales (relación 1 a *).
- InfoExtraOrden: Atributos: cantidad, precioUnitVenta.
 - Relaciones:OrdenCompra: Múltiples detalles de la orden pueden estar asociados a una única orden de compra (relación * a 1).
 - Producto: Cada detalle de la orden está relacionado con un único producto (relación * a 1).
- InfoExtraVenta: Atributos: cantidad, precioUnitario.
 - Relaciones:
 - Venta: Múltiples detalles de la venta pueden estar asociados a una venta (relación * a 1).
 - Producto: Cada detalle de venta se refiere a un único producto (relación * a 1).
- Proveedor: Atributos: nit, nombre, dirección, nombreContacto, teléfonoContacto.
 - Relaciones:
 - EspecificacionesEmpacado: Un proveedor puede tener múltiples especificaciones de empaque para sus productos (relación 1 a *).
 - InfoExtraProveedor: Un proveedor puede estar relacionado con múltiples productos que proporciona (relación 1 a *).
- EspecificacionesEmpacado: Atributos: volumenCm3, pesoGr.
 - Relaciones:
 - Producto: Un producto puede tener diferentes especificaciones de empaquetado (relación * a 1).
 - Proveedor: Un proveedor puede tener múltiples especificaciones de empaquetado para sus productos (relación 1 a *).
- InfoExtraProveedor: Atributos: cantExistencia.
 - Relaciones:

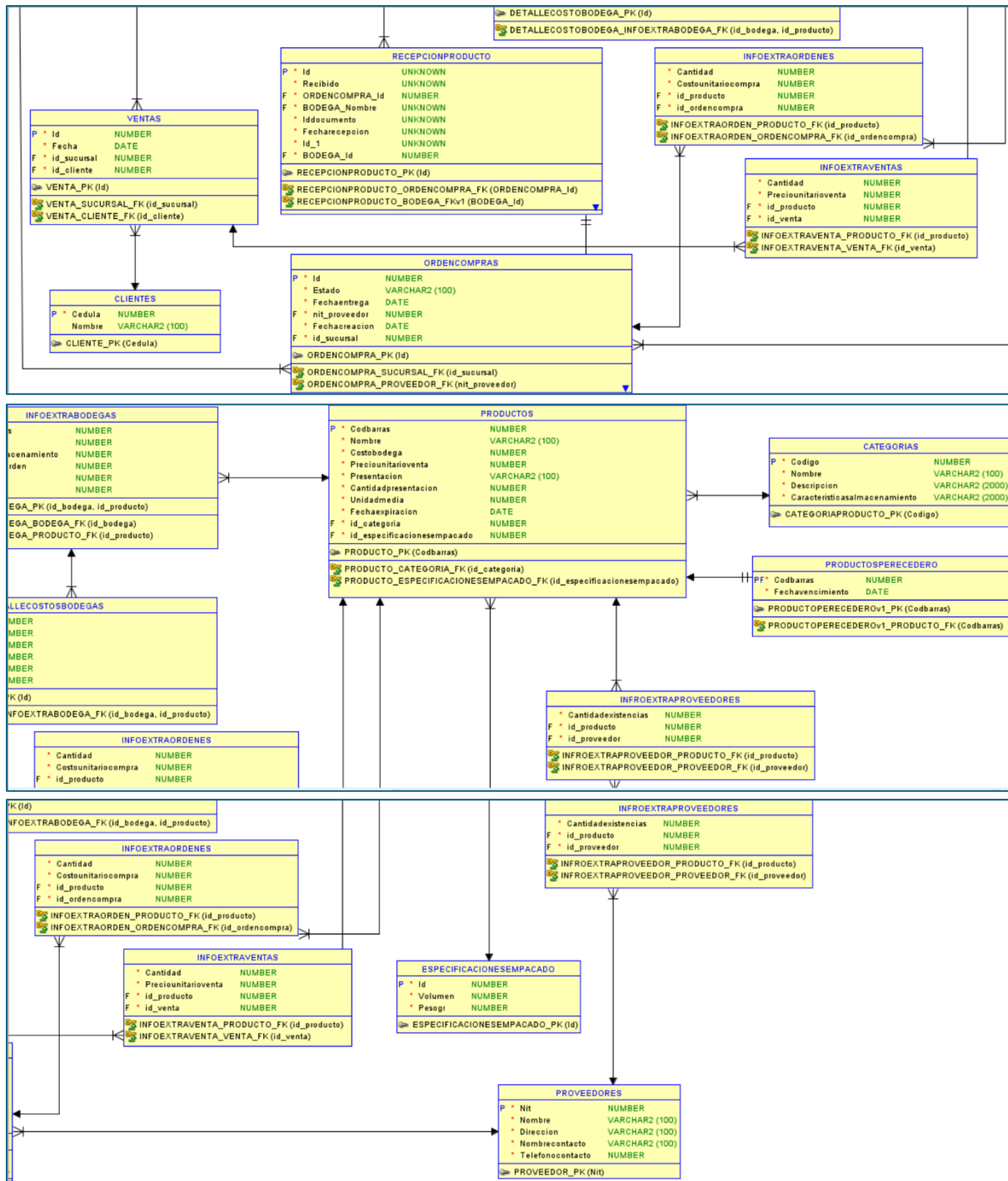
- Producto: Múltiples relaciones de proveedores a productos, con información extra como las existencias (relación * a 1).
- Proveedor: Múltiples productos pueden estar relacionados con un proveedor específico (relación * a 1).

En base al UML se actualizo el Modelo Entidad-Relación:



Con esto llegamos al Modelo Relacional siguiente (para ver mejor el modelo mirar el archivo ModeloRelacionalProyecto1.dmd):





Asimismo, por el cambio en el UML, se realizaron las debidas actualizaciones a las tablas relacionales. A las cuales, se le añadieron las nuevas clases como venta, y se corrigieron las relaciones intermedias. Para poder visualizarlas mejor se adjunta el archivo en formato Excel.

| Sucursales | | | | | |
|------------|----------------|-----------|------------|----------|-------------------|
| id | nombre | direccion | telefono | tamanoM2 | id_ciudad |
| SA, PK | NN | NN | NN | NN | FKciudades.codigo |
| Suc-01 | sucursal norte | Carrera 4 | 322345931 | 500 | 1001 |
| Suc-01 | sucursa sur | calle 55 | 3112018882 | 100 | 1001 |

| Proveedores | | | | |
|-------------|--------|--------------|-------------|-------------|
| nit | nombre | direccion | nomContacto | telContacto |
| PK, UA | NN | NN | NN | NN |
| 1302 | Lay | calle 4#2-34 | Juan | 3112018888 |

| Productos | | | | | | | | |
|-----------|-------------------------|----------------|--|------------------|--------------|-----------------|----------------------------|-----------------------------------|
| codBarras | nombre | precioUnitario | presentacion | cantPresentacion | unidadMedida | fechaExpiracion | id_categoria | id_especificaciones |
| PK, SA | NN | NN | NN | NN | NN | NN, NC | NN, FKcategoriaProducto.id | NN, FKespecificacionesEmpacado.id |
| Pro-01 | Papas fritas les frites | 9000 | paquetón de 5 paquetes de 200 gr. cada uno | 1000 | gr | 12/12/2027 | 123 | 123 |

| Bodegas | | | |
|---------|-----------|----------|-------------------|
| id | nombre | tamanoM2 | id_sucursal |
| PK, SA | NN | NN | NN, FKsucursal.id |
| Bod-01 | Bodeguita | 100 | Suc-01 |

| Ciudades | |
|----------|--------|
| codigo | nombre |
| PK, UA | NN |
| 1001 | Bogotá |
| 1001 | |

| RecepcionProductos | | | |
|--------------------|----------------|------------------|-------------|
| idDocumento | fechaRecepcion | id_OrdenCompra | id_bodega |
| PK, UA | NN, NC | FKordenCompra.id | Fkbodega.id |
| recepcion-01 | 17/02/2006 | Orden-01 | Bod-01 |

| OrdenCompras | | | | | |
|--------------|-----------|---------------|-------------------|---------------|----------------------|
| id | estado | fecha_entrega | id_sucursal | fechaCreacion | id_proveedor |
| PK, SA | NN | NN | FKsucursal.id, NN | NN | FKproveedores.id, NN |
| Orden-01 | Entregada | 12/02/2020 | Suc-01 | 12/01/2992 | 1302 |
| Orden-03 | Entregada | | Suc-70 | | 2029 |

| Categorias |
|------------|
|------------|

| codigo | nombre | descripción | caracAlmacenamiento |
|--------|---------------|----------------|---------------------|
| PK,UA | NN | NN | NN |
| 124 | No-perecedero | No vencimiento | No frio |

| cedula | |
|--------|---------|
| cedula | nombre |
| PK,UA | NN |
| 1001 | Juliana |

| Ventas | | | |
|--------|------------|------------------|------------------|
| id | fecha | id_sucursal | id_cliente |
| PK,SA | NN | Fksucursal.id,NN | NN,Fkclientes.id |
| 124 | 17/02/2002 | SUC11 | 1001 |

Esta clase se añadió, debido a que no la teníamos contemplada y estábamos dejando de lado información necesaria al relacionarla.

| ProductosPerecederos | | | | | | | | | |
|----------------------|-------------------------------|--------------------|--|----------------------|------------------|---------------------|---------------------------------------|--|----------------------|
| codB arras | nombre | precio Unitario | presentacion | cantPres entacion | unidad Medida | fechaEx piracion | id_categori a | id_especificacio nes | fechaVe ncimiento |
| PK,S A | NN | NN | NN | NN | NN | NN,NC | NN, FKcategori aProducto.i d | NN, FKEspecificacio nesEmpacado.i d | NN |
| Pro- 01 | Papas fritas les frites | 9000 | paquetón de 5 paquetes de 200 gr. cada uno | 1000 | gr | 12/12/2 027 | 123 | 123 | 12/02/20 02 |

| InfoExtraVentas | | | |
|-----------------|-------------------|----------|----------------|
| id_venta | id_producto | cantidad | precioUnitario |
| PK,Fkventas.id | PK,Fkproductos.id | NN | NN |
| VENTA100 | PRO 12 | 12 | 1000 |

tabla se añadió para poder tener datos que se necesitan de la relacion que ya estaba.

| InfoExtraOrdenes | | | |
|----------------------|---------------------|----------|---------------|
| id_ordenCompra | id_producto | cantidad | costoUnitario |
| PK,Fkordencompras.id | PK, FK productos.id | NN | NN |
| ORDEN100 | PRO 12 | 12 | 1000 |

Esta tabla se añadió para poder tener datos que se necesitan de la relacion que ya estaba.

| InfoExtraProveedores | | |
|----------------------|-------------------|---------------------|
| id_proveedor | id_producto | cantidadExistencias |
| PK,Fkproveedore.id | PK,Fkproductos.id | NN |
| VENTA100 | PRO 12 | 12 |

Esta tabla se añadió para poder tener datos que se necesitan de la relacion que ya estaba.

| InfoExtraBodegas | | | | | |
|------------------|-------------------|------------------|---------------|-------------------------|--------------------|
| id_bodega | id_producto | totalExistencias | costoPromedio | capacidadAlmacenamiento | nivelMinimoReorden |
| PK,Fjbodegas.id | PK,Fkproductos.id | NN | NN | NN | NN |
| VENTA100 | PRO 12 | 12 | 1000 | 1000 | 1000 |

Esta tabla se añadió para poder tener datos que se necesitan de la relacion que ya estaba.

| DetalleCostosBodegas | | | | | |
|----------------------|---------------------|---------------------|--------------|-----------------------|---------------------------|
| id | costoUnitarioBodega | cantidadExistencias | id_bodega | id_InfoExtra | id_producto |
| PK | NN | NN | Fkbodegas.id | PKinfoExtraBodegas.pk | Fkproductos.codbarra s |
| VENTA100 | PRO 12 | 12 | 1000 | 1000 | 1000 |

| EspecificacionesEmpacado | | |
|--------------------------|---------|------|
| id | Volumen | peso |
| PK,SA | NN | NN |
| es-01 | 1001 | 5 |

La clase especificaciones empacado se añadió debido a que sin ella se violaba la 3FN al no depender de la llave primaria, es decir el peso era información que se obtenía mediante el volumen.

Arquitectura de la aplicación:

La implementación se desarrolla bajo la arquitectura de modelo vista controlador. De esta manera, es posible gestionar múltiples entidades y aspectos del negocio.

La separación en capas permite que cada parte del sistema tenga una responsabilidad bien definida, facilitando el mantenimiento y la escalabilidad del sistema. Las principales capas son: Capa de Lógica de Negocio, Capa de Persistencia: Mapea las clases del diagrama UML a tablas en la base de datos y maneja la persistencia de los datos. Capa de Base de Datos: Almacena los

datos de forma estructurada y relacional. De esta manera, las entidades y relaciones previamente descritas se contienen en estas capas y permiten un flujo de procesos más eficiente en la ejecución de operaciones.

Traza de la ejecución del plan de pruebas

- **RF1:**

Acá estamos creando una ciudad con el nombre de “Barranquilla”, y el código de la ciudad se genera automáticamente por la secuencia creada en el script SQL.

The screenshot shows the Postman interface for a new collection named "Crear una ciudad". The request is a POST to `http://localhost:8080/ciudades/new/save` with a JSON body: `{ "nombre": "Barranquilla" }`. The response status is 201 Created, with a message "Ciudad creada exitosamente".

```
1 {
2   "nombre": "Barranquilla"
3 }
```

Body: Ciudad creada exitosamente

Y así queda añadida en la tabla Ciudades:

| | CODIGO | NOMBRE |
|---|--------|--------------|
| 1 | 1 | Barranquilla |

- **RF2:**

Ahora creamos una nueva sucursal la cual esta relacionada con la ciudad anteriormente insertada, por lo cual se crean los atributos que tendrá esta sucursal, y además se pone la ciudad existente con la que se relaciona.

The screenshot shows the Postman interface for a new collection named "Crear una sucursal". The request is a POST to `http://localhost:8080/sucursales/new/save` with a JSON body: `{ "nombre": "sucursalprueba", "tamanoM2": 100, "direccion": "Carrea 1 #1-2", "telefono": 3841234, "id_ciudad": { "codigo": 1, "nombre": "Barranquilla" } }`. The response status is 201 Created, with a message "Sucursal creada exitosamente".

```
2 "nombre": "sucursalprueba",
3 "tamanoM2": 100,
4 "direccion": "Carrea 1 #1-2",
5 "telefono": 3841234,
6 "id_ciudad": {
7   "codigo": 1,
8   "nombre": "Barranquilla"
9 }
```

Body: Sucursal creada exitosamente

Y así es que como queda ya en la tabla de Sucursales en la base de datos:

| | ID | NOMBRE | TAMANIOM2 | DIRECCION | TELEFONO | ID_CIUADAD |
|---|----|----------------|-----------|---------------|----------|------------|
| 1 | 23 | sucursalprueba | 100 | Carrea 1 #1-2 | 3841234 | 1 |

- **RF3:**

Ahora bien, para esta parte se necesita crear una nueva bodega que igualmente tiene una relación. Esta vez con una sucursal, entonces al crear sus atributos dentro de la tabla, también se pone la sucursal a la cual estaría relacionada.

The screenshot shows a REST client interface with the following details:

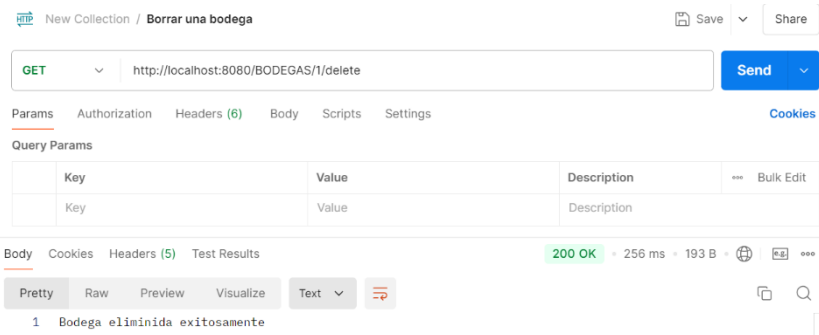
- Method:** POST
- URL:** http://localhost:8080/BODEGAS/new/save
- Body (JSON):**

```
{
  "nombre": "Bodega Prueba",
  "tamaniom2": 25,
  "id_sucursal": {
    "id": 23,
    "nombre": "sucursalprueba",
    "tamaniom2": 100,
    "direccion": "Carrea 1 #1-2",
    "telefono": 3841234,
    "id_ciudad": {
      "codigo": 1,
      "nombre": "Barranquilla"
    }
  }
}
```
- Status:** 201 Created
- Response:** Bodega creada existosamente

En la tabla se ve la bodega creada con nombre “Bodega Prueba” de esta manera:

| | ID | NOMBRE | TAMANIOM2 | ID_SUCURSAL |
|---|------|---------------|-----------|-------------|
| 1 | 1000 | Bodegal | 20 | 1 |
| 2 | 1 | Bodega Prueba | 25 | 23 |

Y para la segunda parte, se tiene que eliminar una bodega. Así que se eliminara la bodega previamente creada, la cual tiene como id, 1.

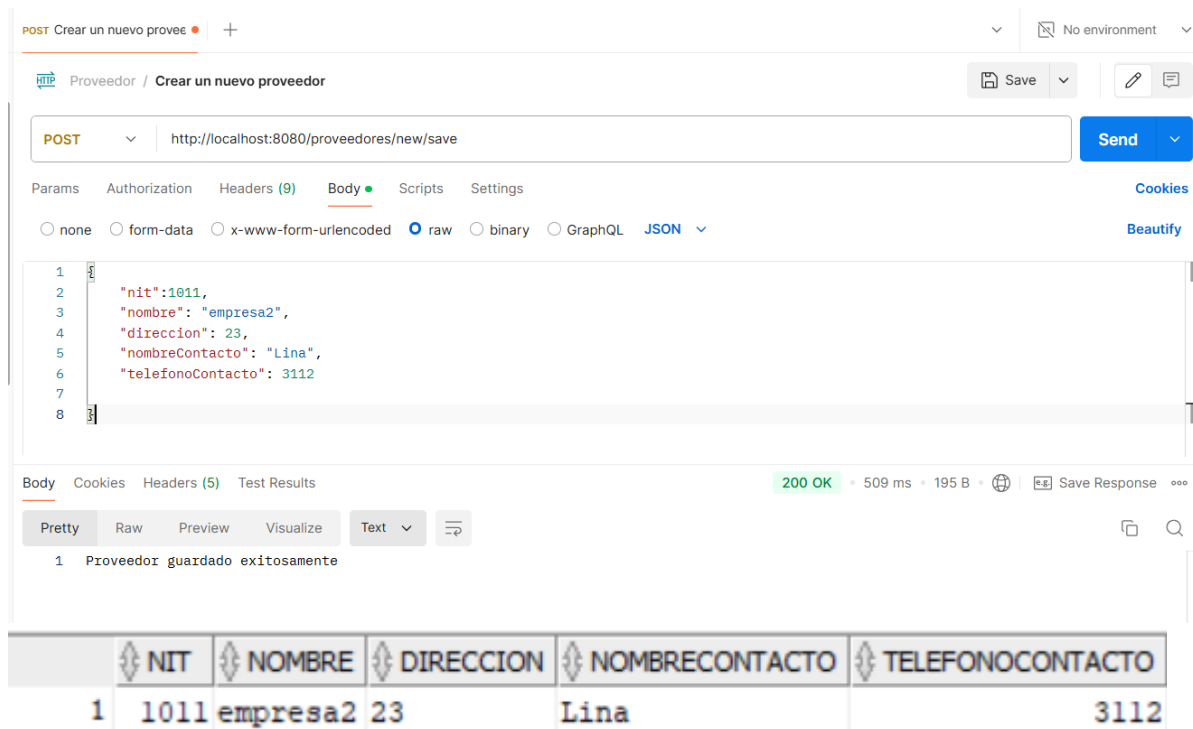


Y ahora si miramos en la tabla de la base de datos en BODEGAS, veremos que la bodega ya no estará ahí:

| ID | NOMBRE | TAMANIOM2 | ID_SUCURSAL |
|----|--------------|-----------|-------------|
| 1 | 1000 Bodega1 | 20 | 1 |

- RF4:**

Acá se necesita crear y actualizar un proveedor, en el ejemplo, vamos a crear uno de nombre “empresa2”. En donde se confirma que no hubo errores al crearse con el mensaje adjunto



Además, es necesario poder actualizarlo, en este caso, se le hace un cambio a el nombre, el cual va a ser “empresa5”. Esta actualización se confirma con el mensaje adjunto.

Proveedor / Actualizar un proveedor

POST http://localhost:8080/proveedores/1011/edit/save

Params Authorization Headers (9) Body Scripts Settings

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

```

1 {
2   "nit":1011,
3   "nombre": "empresa5",
4   "direccion": 23,
5   "nombreContacto": "Lina",
6   "telefonoContacto": 3112
7 }
8

```

Body Cookies Headers (5) Test Results

200 OK • 597 ms • 198 B

Save Response

Pretty Raw Preview Visualize Text

1 Proveedor actualizado exitosamente

| | NIT | NOMBRE | DIRECCION | NOMBRECONTACTO | TELEFONOCONTACTO |
|---|------|----------|-----------|----------------|------------------|
| 1 | 1011 | empresa5 | 23 | Lina | 3112 |

RF5:

En este requerimiento hay que crear una nueva Categoría. Como la categoría no contiene llaves foráneas, simplemente se ponen sus atributos propios de la tabla y se crea.

New Collection / Crear una categoria

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

Params Authorization Headers (8) Body Scripts Settings

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

```

1 {
2   "nombre":"Enlatados",
3   "descripcion":"Preservados en recipientes herméticos ",
4   "caracteristicasAlmacenamiento":"Almacenarse en un lugar fresco y seco"
5 }

```

Body Cookies Headers (5) Test Results

201 Created • 778 ms • 199 B

Save Response

Pretty Raw Preview Visualize Text

1 Categoria creada exitosamente

Ya en la tabla de categorías se guarda y quedaría así:

| | CODIGO | NOMBRE | DESCRIPCION | CARACTERISTICASALMACENAMIENTO |
|---|--------|---------------|---------------------------------------|---------------------------------------|
| 1 | 3 | categoryatest | descripcion producto | varias cosas |
| 2 | 43 | Enlatados | Preservados en recipientes herméticos | Almacenarse en un lugar fresco y seco |

Ahora, también se requiere consultar la categoría por su nombre o código para obtener su información. Así que, primero tenemos la consulta de la categoría con código 1:

HTTP New Collection / Consultar categoria por codigo Save Share

GET Send http://localhost:8080/Categorias/consulta?codigo=1

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

| <input checked="" type="checkbox"/> | Key | Value | Description | ... | Bulk Edit |
|-------------------------------------|--------|-------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | codigo | 1 | | | |
| | Key | Value | Description | | |

Body Cookies Headers (5) Test Results 200 OK 375 ms 285 B 🌐 🔍 ⋮

Pretty Raw Preview Visualize JSON 🔍 🔍

```

1 {
2   "codigo": 1,
3   "nombre": "categoriatest",
4   "descripcion": "descripcion producto",
5   "caracteristicasAlmacenamiento": "varias cosas"
6 }
```

También tenemos la consulta de la Categoría con nombre “Enlatados”:

HTTP New Collection / Consultar categoria por nombre Save Share

GET Send http://localhost:8080/Categorias/consulta?nombre=Enlatados

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

| <input checked="" type="checkbox"/> | Key | Value | Description | ... | Bulk Edit |
|-------------------------------------|--------|-----------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | nombre | Enlatados | | | |
| | Key | Value | Description | | |

Body Cookies Headers (5) Test Results 200 OK 378 ms 326 B 🌐 🔍 ⋮

Pretty Raw Preview Visualize JSON 🔍 🔍

```

1 {
2   "codigo": 43,
3   "nombre": "Enlatados",
4   "descripcion": "Preservados en recipientes herméticos ",
5   "caracteristicasAlmacenamiento": "Almacenarse en un lugar fresco y seco"
6 }
```



RFC1:

En este caso se quiere conocer el porcentaje de ocupación de un producto en una bodega por loque la salida presenta el porcentaje de ocupación: volumen ocupado / capacidad de la bodega o estante para los productos indicados. Así que para la lista de productos siguiente

```

1 {
2   {
3     "codigoBarras":1223,
4     "nombre":"Jugo de lulo",
5     "precioUnitarioVenta":2000,
6     "presentacion":"Caja",
7     "cantidadPresentacion":4,
8     "unidadMedia":2,
9     "Date":"10-OCT-2002"
10  },
11  {
12    "codigoBarras":8765,
13    "nombre":"Kola Roman",
14    "precioUnitarioVenta":2500,
15    "presentacion":"Botella",
16    "cantidadPresentacion":1,
17    "unidadMedia":2,
18    "Date":"10-SEP-2002"
19  }
}
```

Se obtuvo el siguiente resultado:

 http://localhost:8080/BODEGAS/ocupacion  S

GET

http://localhost:8080/BODEGAS/ocupacion

ParamsAuthorizationHeaders (8)BodyScriptsSettings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw



☐ binary

☐ GraphQL

JSON

```
1 [
2   {
3     "codigoBarras":1223,
4     "nombre":"Jugo de lulo",
5     "precioUnitarioVenta":2000,
6     "presentacion":"Caja",
7   },
8   {
9     "codigoBarras":8765,
10    "nombre":"Kola Roman",
11    "precioUnitarioVenta":2200,
12    "presentacion":"Botella",
13  }
14 ]
```

BodyCookiesHeaders (5)Test Results

200 OK • 662 ms • 253 B •  

PrettyRawPreviewVisualizeJSON

```
1 [
2   {
3     "id_producto": 8765,
4     "IndiceOcupacion": 0.38
5   },
6   {
7     "id_producto": 1223,
8     "IndiceOcupacion": 0.45
9   }
10 ]
```

Mostrando que el índice de ocupación para el producto “Kola Roman” es igual a 0.38 o el 38%. Igualmente, para el producto “Jugo de lulo” se obtuvo un índice de ocupación de 0.45 o el 45%. Estos productos se ven referenciados en la tabla INFOEXTRABODEGAS de la siguiente manera:

| | ID_BODEGA | ID_PRODUCTO | TOTALEXISTENCIAS | COSTOPROMEDIO | CAPACIDADALMACENAMIENTO | NIVELMINIMOREORDEN |
|---|-----------|-------------|------------------|---------------|-------------------------|--------------------|
| 1 | 1000 | 8765 | 76 | 2200 | 200 | 2 |
| 2 | 1000 | 1223 | 54 | 2000 | 120 | 10 |
| 3 | 1000 | 1226 | 54 | 2000 | 120 | 10 |