Instituto Tecnológico de Buenos Aires

72.41 - Base de datos II

Segundo Cuatrimestre 2024 - Grupo 3

Trabajo Práctico Obligatorio

Autores:

Daneri, Matias Ezequiel
Limachi, Desiree Melisa
Baron, María Mercedes

Índice

Introducción

El presente trabajo práctico tiene como objetivo desarrollar un sistema de Backoffice orientado al control y gestión de los productos pedidos a proveedores. El sistema debe ser capaz de validar el estado de los proveedores (activo y habilitado) al momento de generar un pedido, calcular el monto total con y sin IVA, y actualizar el stock futuro de los productos, entre otras operaciones.

Para satisfacer los requerimientos funcionales del sistema, se optó por una arquitectura de persistencia basada en dos bases de datos NoSQL complementarias: **MongoDB** y **Redis**.

Decisiones tomadas

MongoDB fue elegida como base de datos principal para la persistencia de la información estructurada relacionada con los proveedores, productos, órdenes y detalles de pedido. Su modelo de documentos flexible permite representar las entidades y relaciones del sistema sin necesidad de esquemas rígidos, facilitando la integración de datos provenientes de múltiples fuentes como archivos `.csv`.

Adicionalmente, MongoDB fue seleccionada considerando la importancia de la consistencia en este tipo de sistema. Al manejar información crítica como stock de productos, precios, montos de órdenes y estados de proveedores, es fundamental garantizar la integridad y coherencia de los datos. MongoDB ofrece mecanismos de consistencia robustos que aseguran que las operaciones sobre inventarios, cálculos de totales con y sin IVA, y actualizaciones de stock futuro se realicen de manera confiable, evitando inconsistencias que podrían afectar la operación del negocio.

Por último, MongoDB ofrece potentes capacidades de consulta y agregación, ideales para resolver las búsquedas y cálculos complejos solicitados (por ejemplo, obtener proveedores con órdenes registradas, calcular montos totales, o filtrar por estado y razón social).

Redis fue incorporada como sistema de soporte complementario, ideal para acelerar el acceso a información crítica y de alta frecuencia de consulta. En particular, Redis se utiliza para gestionar:

- **Caché de resultados frecuentes**, minimizando el tiempo de respuesta en consultas recurrentes como el estado de proveedores o el historial de pedidos.

Gracias a su naturaleza en memoria y su estructura basada en claves, Redis nos permitió realizar estas operaciones con una velocidad superior, manteniendo al mismo tiempo la coherencia con los datos persistidos en MongoDB.

La combinación de **MongoDB** y **Redis** brinda una arquitectura de persistencia **políglota**, eficiente y preparada para escalar. Cada base se emplea en el contexto que mejor se ajusta a sus fortalezas:

- **MongoDB** se ocupa del almacenamiento persistente, consultas complejas y operaciones de agregación.

- **Redis** complementa con velocidad para operaciones en tiempo real, almacenamiento temporal y mejoras de performance.

Este diseño híbrido permite construir un sistema de Backoffice sólido, ágil y alineado con las necesidades del negocio

Estructura de Datos

El sistema de facturación se organiza en torno a cinco entidades principales: teléfono, cliente, operación, detalle_operación y producto.

Tal y como mencionamos en la introducción, se utilizó MongoDB para gestionar la información estructurada del sistema debido a que tiene la capacidad de trabajar con documentos y subdocumentos de una manera eficiente.

Para minimizar la complejidad de las consultas y evitar “joins” innecesarios, se optó por documentos embebidos donde la relación es estrictamente dependiente: los teléfonos viven dentro de su proveedor y los detalles de las órdenes dentro de cada orden. Esto permite obtener la información completa de un proveedor o de una orden en una sola lectura, reduciendo la latencia y simplificando la lógica de acceso. En el resto de las relaciones se eligió normalizar los datos. Por ejemplo, Order necesita conocer al proveedor pero no duplicar todo su contenido, para esto se utilizó una referencia mediante el campo supplierId.

Además, consideramos la normalización de las órdenes y los productos. Era fundamental no embeber los productos dentro de las órdenes, ya que cada producto mantiene su propio stock. De haber elegido por el embebido, la creación de cada orden habría generado una ineficiencia al requerir la actualización de múltiples subdocumentos para mantener la consistencia del inventario.

Por ende, el sistema está compuesto por las siguientes entidades principales:

- **Supplier (Proveedor)**: representa a los proveedores registrados. Contiene:
  - **id** (*String*): Identificador único del proveedor.
  - **taxId** (*String*): CUIT o número de identificación tributaria.
  - **companyName** (*String*): Razón social del proveedor.
  - **companyType** (*String*): Tipo de empresa o sociedad.
  - **address** (*String*): Dirección física del proveedor.
  - **active** (*boolean*): Indica si el proveedor se encuentra activo.
  - **authorized** (*boolean*): Indica si el proveedor está habilitado para operar.
  - **phones** (*List<Phone>*): Lista de teléfonos de contacto asociados al proveedor. Esta lista está **embebida** dentro de la entidad `Supplier`.
  - **Phone** (**`Teléfono`**) están **embebidos** dentro de la entidad **`Supplier`**, ya que forman parte de su información de contacto inmediata.
    - **supplierId** (*String*): Identificador del proveedor al que pertenece el teléfono.
    - **areaCode** (*String*): Código de área del número telefónico.
    - **phoneNumber** (*String*): Número de teléfono.
    - **type** (*String*): Tipo de teléfono (por ejemplo, fijo o celular).

- **Order (Orden de Pedido)**: representa un pedido realizado a un proveedor en una fecha determinada. Almacena los montos totales con y sin impuestos.
  - **id** (*String*): Identificador único de la orden.

    - **supplierId** (*String*): Identificador del proveedor al que se realiza la orden.
    - **date** (*String*): Fecha en la que se registró la orden.
    - **totalWithoutTax** (*Double*): Monto total de la orden sin incluir IVA.
    - **tax** (*Double*): Monto correspondiente al IVA de la orden.
    - **orderDetails** (*List<OrderDetail>*): Lista de detalles que especifican los productos pedidos.
    - **OrderDetail** (**Detalles de orden**) están **embebidos** dentro de `Order`, dado que forman una unidad lógica y no necesitan ser consultados de forma aislada.
        - **orderId** (*String*): Identificador de la orden a la que pertenece el detalle.
        - **productId** (*String*): Identificador del producto solicitado.
        - **itemNumber** (*int*): Número de ítem o secuencia dentro de la orden.
        - **quantity** (*double*): Cantidad solicitada del producto.

- **Product (Producto)**: contiene la descripción, marca, categoría y precios de cada producto, además del stock actual y el stock futuro proyectado.
    - **id** (*String*): Identificador único del producto.
    - **description** (*String*): Descripción del producto.
    - **brand** (*String*): Marca del producto.
    - **category** (*String*): Categoría a la que pertenece.
    - **price** (*double*): Precio unitario del producto.
    - **currentStock** (*int*): Stock actual disponible.
    - **futureStock** (*int*): Stock proyectado, que se incrementa con nuevos pedidos.

Las consultas de mayor frecuencia, o lo que supusimos, ayudará en la eficiencia del sistema, (proveedores activos, autorizados, o su intersección) se aceleran usando claves específicas en Redis (SUPPLIERS_ACTIVE, SUPPLIERS_UNAUTHORIZED, etc.) con un TTL corto de 30s. El flujo es simple: si la clave existe se devuelve el JSON cacheado (HIT); si no, se consulta MongoDB, se serializa con Jackson y se guarda en Redis (MISS). Este patrón esperamos reduzca significativamente el tiempo de respuesta sin comprometer el freshness de los datos, ya que toda operación de escritura sobre proveedores inválida o actualiza las mismas claves.

Requerimientos del sistema

A continuación, se describen los endpoints desarrollados para el sistema, agrupados por entidad (`Proveedores`, `Órdenes`, `Productos`). Se incluyen tanto las funcionalidades específicas solicitadas como las operaciones básicas de creación, lectura, actualización y eliminación (CRUD).

Es importante destacar que sobre las operaciones de creación de nuevas entidades de proveedores y órdenes incluyen los documentos embebidos(teléfonos y detalles de orden) correspondientes dentro del mismo documento. Esta decisión se fundamenta en el enfoque de documentos embebidos que adoptamos, ya que consideramos que no tendría sentido lógico la existencia de los subdocumentos sin su responsable.

Al mismo tiempo, es importante señalar que las operaciones de actualización pueden resultar en la eliminación de tales documentos embebidos existentes. Por lo tanto, al momento de realizar actualizaciones de proveedores y ordenes, es fundamental que todos los números telefónicos y detalles de órdenes que deban conservarse estén explícitamente contemplados en la operación, ya que cualquiera que no esté incluido en la actualización será removido del documento.

Por otro lado, se incluyen los resultados obtenidos por cada consulta en el anexo del informe.

Proveedores (Suppliers)

Consultas específicas:

- **[Ejercicio 1]** `GET /suppliers/active/phones`
  Devuelve los proveedores activos y autorizados, incluyendo sus teléfonos. Este endpoint se apoya en Redis para mejorar el rendimiento en la recuperación de estos datos.
- **[Ejercicio 2]** `GET /suppliers/tech/phones`
  Devuelve el o los teléfonos y el identificador de los proveedores cuya razón social contenga la palabra "Tecnología".
- **[Ejercicio 3]** `GET /suppliers/phones`
  Devuelve cada teléfono junto con los datos completos del proveedor al que pertenece.
- **[Ejercicio 4]** `GET /suppliers/with-orders`
  Devuelve todos los proveedores que tienen al menos una orden de pedido registrada.
- **[Ejercicio 5]** `GET /suppliers/without-orders`
  Devuelve todos los proveedores que no tienen órdenes registradas. Además, incluye información sobre si el proveedor está activo y autorizado.
- **[Ejercicio 6]** `GET /suppliers/with-orders-summary`
  Devuelve todos los proveedores, indicando la cantidad de órdenes registradas y los montos totales pedidos, con y sin IVA.
- **[Ejercicio 12]** `GET /suppliers/active-unauthorized`
  Devuelve los proveedores que se encuentran activos, pero no están autorizados.

Operaciones CRUD (Resolución del ejercicio 13):

- `GET /suppliers`
  Devuelve la lista completa de proveedores registrados.
- `GET /suppliers/{id}`
  Devuelve los datos de un proveedor específico, identificado por su ID.
- `POST /suppliers`
  Permite registrar un nuevo proveedor en el sistema.
- `PUT /suppliers/{id}`
  Permite modificar los datos de un proveedor existente.

- `DELETE /suppliers/{id}`
  Permite eliminar un proveedor del sistema a partir de su ID.

Órdenes (Orders)

Consultas específicas:

- **[Ejercicio 7]** `GET /orders/by-supplier-tax-id`
  Devuelve todas las órdenes registradas para el proveedor cuyo CUIT es `30-66060817-5`. El CUIT se debe enviar como parámetro en la consulta.
- **[Ejercicio 9]** `GET /orders/with-coto-products`
  Devuelve todas las órdenes que contienen al menos un producto de la marca "COTO".
- **[Ejercicio 10]** `GET /orders/detailed-summary`
  Devuelve todas las órdenes registradas, ordenadas por fecha, incluyendo la razón social del proveedor y los montos totales (con y sin IVA).
- **[Ejercicio 15]** `POST /orders`
  Permite registrar una nueva orden de pedido a un proveedor. La operación solo se permite si el proveedor está activo y autorizado.

Operaciones CRUD:

- `GET /orders`
  Devuelve todas las órdenes registradas en el sistema.
- `GET /orders/{id}`
  Devuelve los datos de una orden específica, identificada por su ID.
- `POST /orders`
  Permite registrar una nueva orden.
- `PUT /orders/{id}`
  Permite modificar una orden existente.
- `DELETE /orders/{id}`
  Permite eliminar una orden del sistema.

Productos (Products)

Consultas específicas:

- **[Ejercicio 8]** `GET /products/with-orders`
  Devuelve todos los productos que han sido pedidos al menos una vez.
- **[Ejercicio 11]** `GET /products/without-orders`
  Devuelve todos los productos que no han sido incluidos en ninguna orden hasta el momento.

Operaciones CRUD (Resolución del ejercicio 14):

- `GET /products`
  Devuelve la lista completa de productos registrados.

- `GET /products/{id}`
  Devuelve los datos de un producto específico, identificado por su ID.
- `POST /products`
  Permite registrar un nuevo producto en el sistema.
- `PUT /products/{id}`
  Permite modificar los datos de un producto existente.
- `DELETE /products/{id}`
  Permite eliminar un producto del sistema.

Conclusiones

El desarrollo del sistema de Backoffice propuesto permitió abordar, de forma integral, la gestión de proveedores, productos y órdenes de pedido, utilizando tecnologías modernas de persistencia NoSQL. A lo largo del trabajo se logró:

- **Modelar correctamente los datos** involucrados en el proceso de compra a proveedores, utilizando estructuras embebidas donde fue pertinente (por ejemplo, teléfonos dentro de proveedores y detalles dentro de órdenes), lo cual optimiza el acceso y la coherencia de los datos
- **Implementar una capa de persistencia políglota**, utilizando **MongoDB** para consultas complejas y almacenamiento estructurado, y **Redis** como sistema de caché para mejorar el rendimiento en accesos frecuentes, como el caso de proveedores activos y autorizados.
- **Desarrollar una API RESTful clara y coherente**, exponiendo endpoints significativos y organizados por entidad, que permiten realizar operaciones tanto CRUD como consultas específicas requeridas por el sistema.

En conjunto, el sistema implementado cumple satisfactoriamente con los requisitos funcionales y técnicos planteados, brindando una base sólida y escalable para su posible evolución futura. La elección de una arquitectura basada en servicios REST y persistencia distribuida lo posiciona favorablemente ante escenarios reales de operación, donde la eficiencia y flexibilidad son clave.

Anexo

Se anexan las respuestas de cada consulta, se aclara que las mismas son una parte de la respuesta de forma representativa.

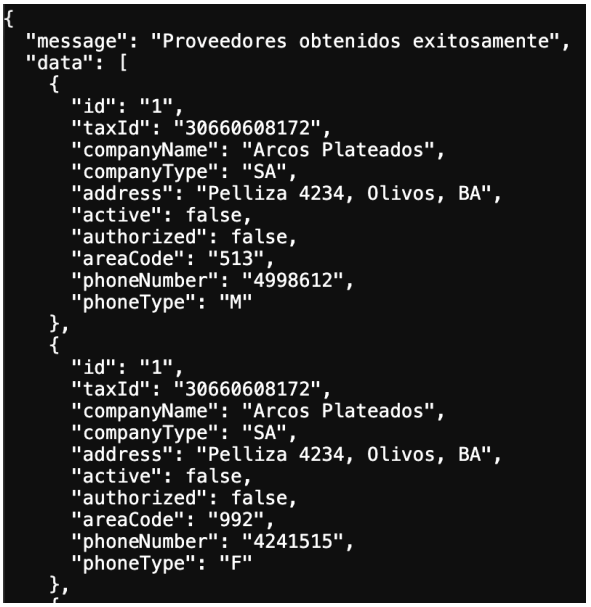
- **[Ejercicio 1]** `GET /suppliers/active/phones`
  Devuelve los proveedores activos y autorizados, incluyendo sus teléfonos. Este endpoint se apoya en Redis para mejorar el rendimiento en la recuperación de estos datos.
- **[Ejercicio 2]** `GET /suppliers/tech/phones`
  Devuelve el o los teléfonos y el identificador de los proveedores cuya razón social contenga la palabra "Tecnología".

Ejercicio 1

Ejercicio 2

- **[Ejercicio 3]** `GET /suppliers/phones`
  Devuelve cada teléfono junto con los datos completos del proveedor al que pertenece.
- **[Ejercicio 4]** `GET /suppliers/with-orders`
  Devuelve todos los proveedores que tienen al menos una orden de pedido registrada.

Ejercicio 3




```
{
  "message": "Proveedores obtenidos exitosamente",
  "data": [
    {
      "id": "1",
      "taxId": "30660608172",
      "companyName": "Arcos Plateados",
      "companyType": "SA",
      "address": "Pelliza 4234, Olivos, BA",
      "active": false,
      "authorized": false,
      "areaCode": "513",
      "phoneNumber": "4998612",
      "phoneType": "M"
    },
    {
      "id": "1",
      "taxId": "30660608172",
      "companyName": "Arcos Plateados",
      "companyType": "SA",
      "address": "Pelliza 4234, Olivos, BA",
      "active": false,
      "authorized": false,
      "areaCode": "992",
      "phoneNumber": "4241515",
      "phoneType": "F"
    },
```


Ejercicio 4

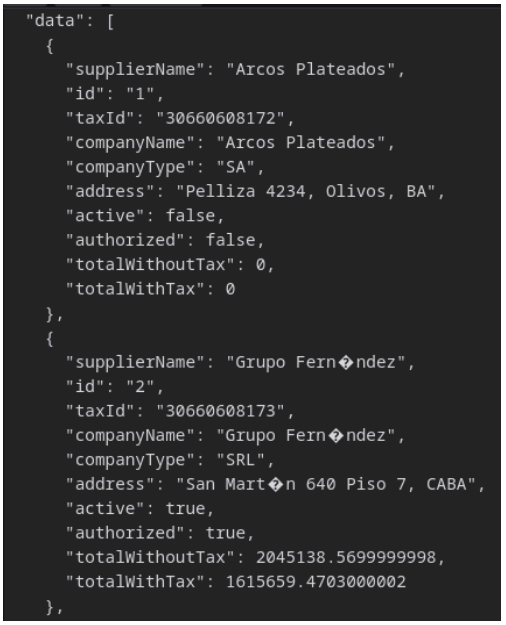

```
{
  "message": "Proveedores obtenidos exitosamente",
  "data": [
    {
      "id": "1",
      "taxId": "30660608172",
      "companyName": "Arcos Plateados",
      "companyType": "SA",
      "address": "Pelliza 4234, Olivos, BA",
      "active": false,
      "authorized": false,
      "phones": [
        {
          "supplierId": "1",
          "areaCode": "513",
          "phoneNumber": "4998612",
          "type": "M"
        },
        {
          "supplierId": "1",
          "areaCode": "992",
          "phoneNumber": "4241515",
          "type": "F"
        }
      ]
    },
    {
```


- **[Ejercicio 5]** `GET /suppliers/without-orders`
  Devuelve todos los proveedores que no tienen órdenes registradas. Además, incluye información sobre si el proveedor está activo y autorizado.
- **[Ejercicio 6]** `GET /suppliers/with-orders-summary`
  Devuelve todos los proveedores, indicando la cantidad de órdenes registradas y los montos totales pedidos, con y sin IVA.

Ejercicio 5


```
[
  {
    "id": "58",
    "taxId": "30660608240",
    "companyName": "Cole",
    "companyType": "SA",
    "address": "Av del Libertador 20563, CABA",
    "active": true,
    "authorized": true,
    "phones": [
      {
        "supplierId": "58",
        "areaCode": "428",
        "phoneNumber": "4354331",
        "type": "M"
      },
      {
        "supplierId": "58",
        "areaCode": "684",
        "phoneNumber": "4541672",
        "type": "M"
      },
      {
        "supplierId": "58",
        "areaCode": "776",
        "phoneNumber": "4552842",
        "type": "F"
      }
    ]
  },
  {
```


Ejercicio 6



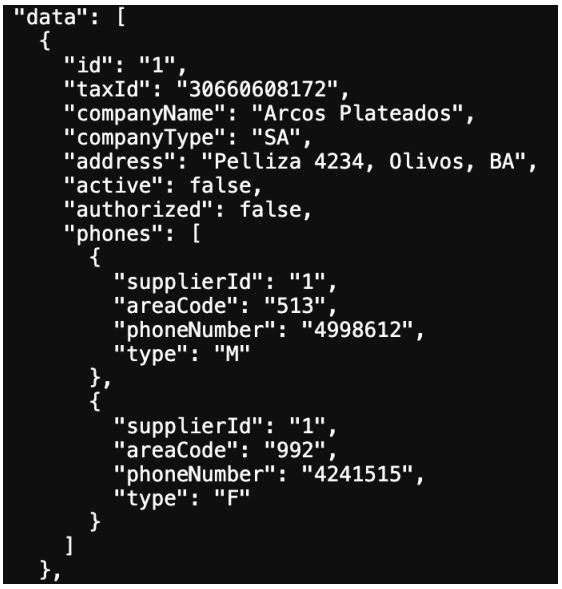

```
"data": [
  {
    "supplierName": "Arcos Plateados",
    "id": "1",
    "taxId": "30660608172",
    "companyName": "Arcos Plateados",
    "companyType": "SA",
    "address": "Pelliza 4234, Olivos, BA",
    "active": false,
    "authorized": false,
    "totalWithoutTax": 0,
    "totalWithTax": 0
  },
  {
    "supplierName": "Grupo Fern\ufffdndez",
    "id": "2",
    "taxId": "30660608173",
    "companyName": "Grupo Fern\ufffdndez",
    "companyType": "SRL",
    "address": "San Mart\ufffdn 640 Piso 7, CABA",
    "active": true,
    "authorized": true,
    "totalWithoutTax": 2045138.5699999998,
    "totalWithTax": 1615659.4703000002
  },
```

- **[Ejercicio 12]** `GET /suppliers/active-unauthorized`
  Devuelve los proveedores que se encuentran activos, pero no están autorizados


```
"data": [
  {
    "id": "11",
    "taxId": "30660608182",
    "companyName": "Alvarado",
    "companyType": "SA",
    "address": "Av del Libertador 2560, CABA",
    "active": true,
    "authorized": false,
    "phones": [
      {
        "supplierId": "11",
        "areaCode": "115",
        "phoneNumber": "4573879",
        "type": "F"
      }
    ]
  },
  {
    "id": "12",
    "taxId": "30660608183",
    "companyName": "Hubbard",
    "companyType": "SA",
    "address": "Calle 45 452, La Plata, BA",
    "active": true,
    "authorized": false,
    "phones": [
      {
        "supplierId": "12",
        "areaCode": "146",
        "phoneNumber": "4579375",
        "type": "F"
      },
      {
        "supplierId": "12",
        "areaCode": "336",
        "phoneNumber": "4869818",
        "type": "M"
      },
      {
        "supplierId": "12",
        "areaCode": "698",
        "phoneNumber": "4653876",
        "type": "F"
      },
      {
        "supplierId": "12",
        "areaCode": "712",
        "phoneNumber": "4652912",
        "type": "F"
      }
    ]
  },
```

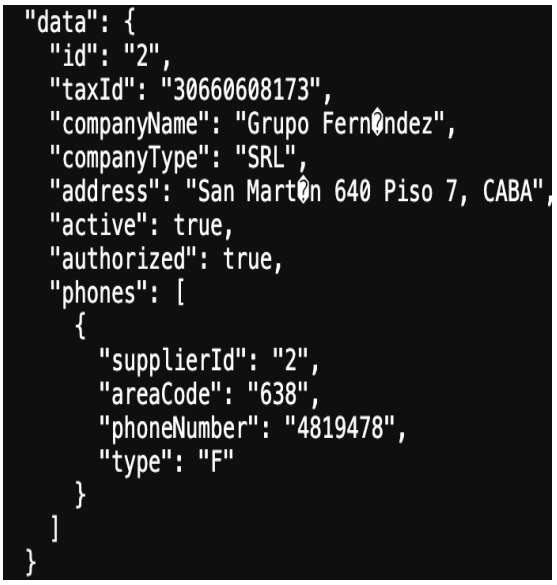

- **[Ejercicio 13]** Devuelve el CRUD de suppliers
  - `GET /suppliers`
    Devuelve la lista completa de proveedores registrados.
  - `GET /suppliers/{id}`
    Devuelve los datos de un proveedor específico, identificado por su ID.

`GET /suppliers`



```
"data": [
  {
    "id": "1",
    "taxId": "30660608172",
    "companyName": "Arcos Plateados",
    "companyType": "SA",
    "address": "Pelliza 4234, Olivos, BA",
    "active": false,
    "authorized": false,
    "phones": [
      {
        "supplierId": "1",
        "areaCode": "513",
        "phoneNumber": "4998612",
        "type": "M"
      },
      {
        "supplierId": "1",
        "areaCode": "992",
        "phoneNumber": "4241515",
        "type": "F"
      }
    ]
  },
```


`GET /suppliers/{id}`



```
"data": {
  "id": "2",
  "taxId": "30660608173",
  "companyName": "Grupo Fern@ndez",
  "companyType": "SRL",
  "address": "San Mart@n 640 Piso 7, CABA",
  "active": true,
  "authorized": true,
  "phones": [
    {
      "supplierId": "2",
      "areaCode": "638",
      "phoneNumber": "4819478",
      "type": "F"
    }
  ]
}
```


- **[Ejercicio 7]** `GET /orders/by-supplier-tax-id`
  Devuelve todas las órdenes registradas para el proveedor cuyo CUIT es `30660608175`

  En este caso se devuelve una lista vacía ya que, según el CSV de datos, ese proveedor no está activo ni habilitado por ende, no se le permitió insertarse.

- **[Ejercicio 9]** `GET /orders/with-coto-products`
  Devuelve todas las órdenes que contienen al menos un producto de la marca "COTO".

```
{
  "id": "4",
  "supplierId": "18",
  "date": "8/2/2024",
  "totalWithoutTax": 296841.08999999997,
  "tax": 21,
  "products": [
    {
      "id": "64",
      "description": "Ri\ufffdon",
      "brand": "COTO",
      "category": "meat",
      "price": 773.89,
      "currentStock": 36,
      "futureStock": 818
    }
  ]
},
{
  "id": "7",
  "supplierId": "23",
  "date": "7/12/2023",
  "totalWithoutTax": 304463.52,
  "tax": 21,
  "products": [
    {
      "id": "23",
      "description": "Manteca",
      "brand": "COTO",
      "category": "groceries",
      "price": 460,
      "currentStock": 8,
      "futureStock": 1002
    }
  ]
},
```


- **[Ejercicio 10]** `GET /orders/detailed-summary`
  Devuelve todas las órdenes registradas, ordenadas por fecha, incluyendo la razón social del proveedor y los montos totales (con y sin IVA).


```
{
  "orderId": "86",
  "date": "3/3/2023",
  "companyName": "Estes",
  "totalWithoutTax": 579121.25,
  "totalWithTax": 457505.78750000003
},
{
  "orderId": "140",
  "date": "4/3/2023",
  "companyName": "Parrish",
  "totalWithoutTax": 0,
  "totalWithTax": 0
},
```


- **[Ejercicio 8]** `GET /products/with-orders`
  Devuelve todos los productos que han sido pedidos al menos una vez.

- **[Ejercicio 11]** `GET /products/without-orders`
  Devuelve todos los productos que no han sido incluidos en ninguna orden hasta el momento.

Ejercicio 8

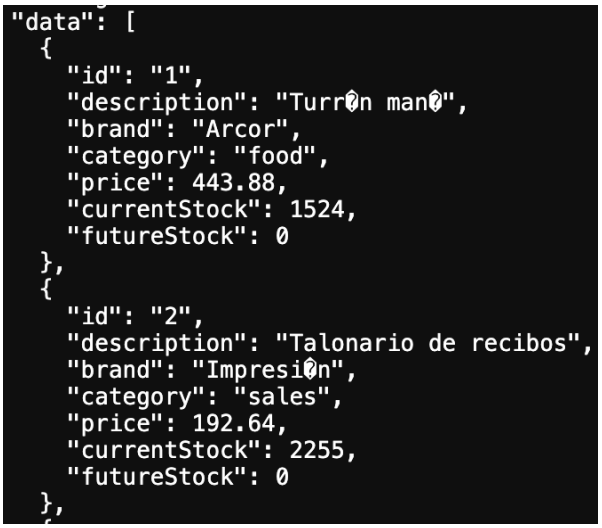
```
"data": [
  {
    "id": "1",
    "description": "Turr□n man□",
    "brand": "Arcor",
    "category": "food",
    "price": 443.88,
    "currentStock": 1524,
    "futureStock": 0
  },
  {
    "id": "2",
    "description": "Talonario de recibos",
    "brand": "Impresi□n",
    "category": "sales",
    "price": 192.64,
    "currentStock": 2255,
    "futureStock": 0
  },
```


Ejercicio 11

```
"data": [
  {
    "id": "90",
    "description": "Turrón maní",
    "brand": "Arcor",
    "category": "food",
    "price": 443.88,
    "currentStock": 404,
    "futureStock": 0
  }
]
```

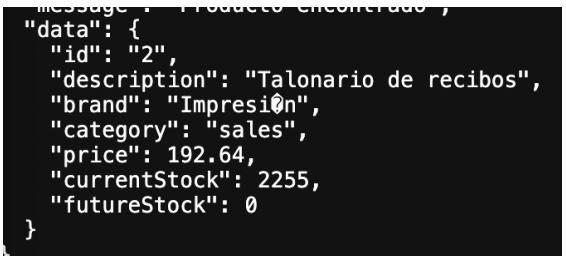

- **[Ejercicio 14]**
- `GET /products`
  Devuelve la lista completa de productos registrados.
- `GET /products/{id}`
  Devuelve los datos de un producto específico, identificado por su ID.

`GET /products`



```
"data": [
  {
    "id": "1",
    "description": "Turr□n man□",
    "brand": "Arcor",
    "category": "food",
    "price": 443.88,
    "currentStock": 1524,
    "futureStock": 0
  },
  {
    "id": "2",
    "description": "Talonario de recibos",
    "brand": "Impresi□n",
    "category": "sales",
    "price": 192.64,
    "currentStock": 2255,
    "futureStock": 0
  },
```


`GET /products/{id}`



```
"data": {
  "id": "2",
  "description": "Talonario de recibos",
  "brand": "Impresi□n",
  "category": "sales",
  "price": 192.64,
  "currentStock": 2255,
  "futureStock": 0
}
```