**El script de la base de datos se encuentra ubicado en la carpeta DB en el proyecto, este documento se encuentra ubicado dentro la misma carpeta**

1. **Cargue de información correspondiente a productos.**

En este punto lo que realize fue una función que tome los datos del json correspondientes a los productos de una orden y los inserta en la base de datos.

Para este cargue de información se usa el controlador de tipo recurso Product/ProductController en la función Store.

El json debería ser algo similar a esto:

{ "products": [

{

"id": 1,

"name": "Leche",

"quantity": 1

}

]

}

La ruta correspondiente es: /product - POST

1. **Cargue de información correspondiente a los Proveedores**

En este punto se realiza el cargue de información correspondiente para los proveedores, para esto es necesario enviar el json enviado en la prueba **providers-merqueo.**

Para este cargue de información se usa el controlador de tipo recurso Provider/ProviderController en la función Store.

La ruta correspondiente es: /provider - POST

1. **Cargue de información correspondiente a inventarios**

En este punto se realiza el cargue de información a los inventario, para eso solo es necesario enviar el json enviado en la prueba **inventory-merqueo.**

Para este cargue de información se usa el controlador de tipo recurso Inventary/InventaryController en la función Store.

La ruta correspondiente es: /inventory - POST

1. **Cargue de información correspondiente para órdenes**

En este punto se realiza el cargue de información correspondiente para las órdenes,

para esto es necesario cargar el json adjuntado en la prueba **orders-merqueo.**

Para este cargue de información se usa el controlador de tipo recurso Order/OrderController en la función Store.

La ruta correspondiente es: /order - POST

1. **Consulta de productos y cantidad disponible para ser alistada**

Esta consulta es realizada por el controlador de los inventarios, ubicada en Inventary/InventaryController en la función index ya que es un controlador de recursos, además es recepcionada por la ruta definida correspondiente.

La ruta correspondiente es: /inventary - GET

1. **Productos que deben ser alistados por transportadores**

Esta consulta es realizada por el controlador de los Ordenes, ubicada en Order/OrderController en la función productsEnlisted, además es recepcionada por la ruta definida correspondiente.

La ruta correspondiente es: /productsEnlisted - GET

1. **Productos menos vendidos**

Esta consulta es realizada por el controlador de los Productos, ubicada en Product/ProductController en la función productWorst, además es recepcionada por la ruta definida correspondiente.

La ruta correspondiente es: /productWorst - GET

1. **Dado el ID del pedido saber que productos y cantidad deben ser alistados y los que deben ser abastecidos por los provedores**

Esta consulta es realizada por el controlador de recursos correspondiente a los Productos, ubicada en Product/ProductController en la función show, además es recepcionada por la ruta definida correspondiente.

La ruta correspondiente es: /product/{ codigo\_de\_orden } - GET

1. **Inventario para el dia 2 de marzo**

Esta consulta es realizada por el controlador de los Inventarios, ubicada en Inventary/InventaryController en la función productInventary, además es recepcionada por la ruta definida correspondiente.

La ruta correspondiente es: /productInventary - GET

1. **Productos más vendidos**

Esta consulta es realizada por el controlador de los Productos, ubicada en Product/ProductController en la función productBest, además es recepcionada por la ruta definida correspondiente.

La ruta correspondiente es: /productBest - GET

**Agradezco su atencion prestada,**

**Cordialmente,**

**Brahian Sánchez - Desarrollador Backend**

**Cel. 3202522965**