Solución Prueba Merqueo

Por: Lina María Mendez Herrera

Contenido carpeta:

BD: contiene el objeto dumb de la base de datos
Rest : contiene la aplicación en general.

Contenido Documento

1.Base de datos

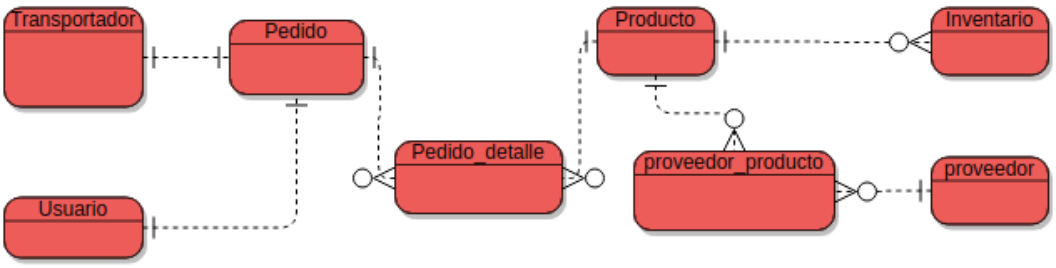
Motor utilizado: MYSQL

La base de datos esta conformada por 8 tablas, las cuales son:

- **Transportador:** tabla que almacena los datos maestros de los mensajeros / transportadores que son los encargados de alistar los pedidos y llevarlos a su destino.
- **Proveedor**: tabla que almacena los datos maestros de los proveedores, los cuales se encargan de abastecer los productos que no se encuentren en el inventario.
- **usuario**: tabla que almacena los datos maestros de los clientes / usuarios, los cuales son los encargados de realizar los pedidos.
- **Producto**: tabla que almacena los diferentes productos con sus datos bases.
- **Pedido**: tabla que almacena los pedidos junto con su dirección, el usuario que lo solicito y la fecha, y si prioridad.
- **Pedido_detalle**: tabla que almacena el detalle según el id de un pedido, en esta tabla esta la lista de productos que fueron solicitados en el pedido, la cantidad y su precio.

- **Proveedor_producto**: esta tabla almacena los productos que pueden ser abastecidos por ciertos proveedores.
- **Inventario**: esta tabla almacena el inventario de los productos y su cantidad disponible para alistar en cierta fecha dada.

A continuación, un diagrama sencillo de entidad relación de la base de datos:



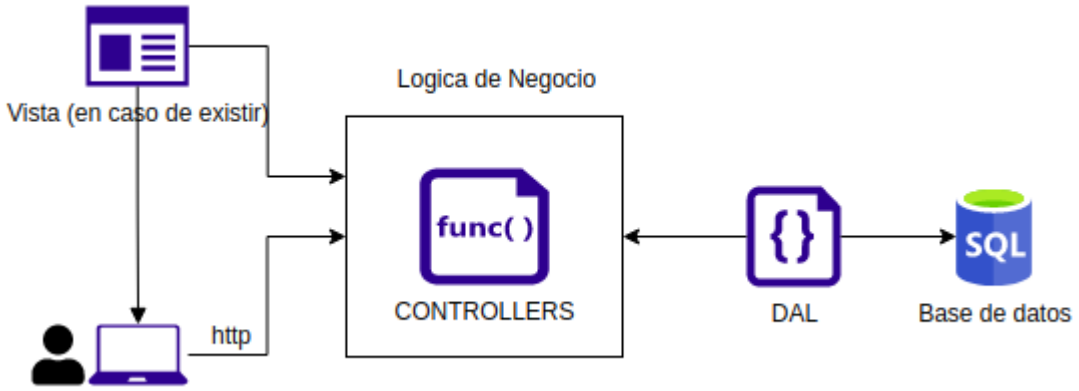
2.Funcionamiento del proyecto:

Lenguaje desarrollado: PHP

El proyecto esta divido en dos capas básicas por el patrón MVC, no se utilizó ningún framework.

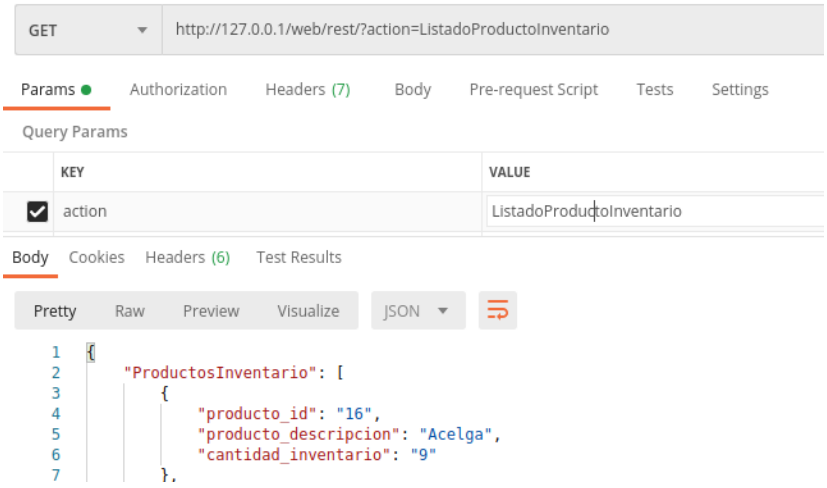
- **Capa DAL:** la capa de accedo de datos, esta capa contiene todo lo necesario para que la comunicación con la base de datos sea exitosa, los archivos que contiene son:
  - **Conexión**: es el archivo de conexión a la base de datos (mysql).
  - **InterfaceAccesoDatos**: Es la interface que expone los métodos abstractos.
  - **AccesoDatos**: Implementa la interface con todos los métodos y maneja realiza las diferentes consultas a la base de datos.
- **Capa CONTROLLERS:** la capa que maneja la lógica de negocio, esta capa sirve de enlace entre las vistas y los DAL, en ella no se debe implementar ningún tipo de consulta directa a lavase de datos, los archivos que la conforman son:
  - **Controlador**: Es el controlador general del sistema, en el cual se deben realizar las validaciones necesarias antes de ser enviadas a la capa DAL y así evitar errores.
  - **API**: es el servicio general que recibe las peticiones del usuario, maneja todo por el método GET y se base en la variable “action” para realizar las acciones correspondientes.
- **Extras**:
  - **Index**: Es el archivo raíz que instancia la clase del api.

A continuación, un diagrama simple con los componentes del proyecto:



3.Solución a los puntos

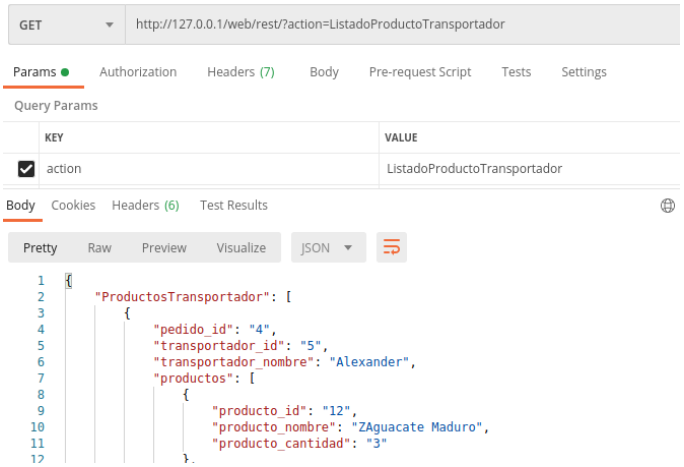
1. Consultar qué productos y qué cantidad puede ser alistada desde el inventario.
   - **Url consumo**: http://IP_SERVER/rest/?action=ListadoProductoInventario
   - **Método**: GET
   - **Parámetros**: ninguno
   - **Descripción**: Se realizo una consulta la base de datos la cual retorna una json con los siguientes datos:
     - **ProductosInventarios**:
       - **Producto_id**: id del producto disponible en el inventario.
       - **Producto_descripcion**: nombre del producto.
       - **Cantidad_inventario**: cantidad disponible del producto en el inventario.
   - **Imagen de consumo en POST-MAN:**



2. Consultar los productos que deben ser alistados por transportadores, y a qué transportador le corresponde cada pedido.
   - **Url consumo**: http://IP_SERVER/rest/?action=ListadoProductoTransportador
   - **Método**: GET
   - **Parámetros**: ninguno

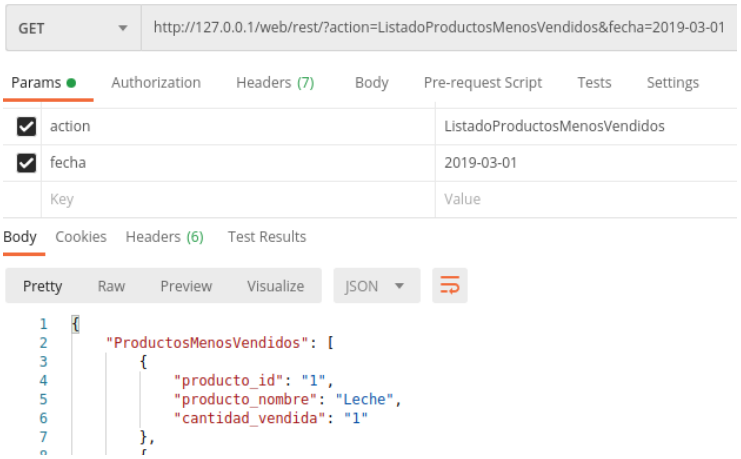
- **Descripción**: Se realizo una lista por pedido con sus correspondientes productos y se muestra el trasportador que está encargado de alistar cada pedido, un trasportador puede tener varios pedidos:
    - **ProductosTransportador**:
        - **pedido_id**: id del pedido correspondiente a un trasportador.
        - **transportador_id** : id del trasportador encargado del pedido.
        - **transportador_nombre**: nombre del trasportador encargado de llevar el pedido.
        - **productos**: arreglo con los datos de los productos correspondientes a cada pedido
            - **Producto_id**: id del producto disponible en el inventario.
            - **producto_nombre**: nombre del producto.
            - **producto_cantidad**: cantidad que debe ser alistada por el transportador traida según disponibilidad en el inventario.

- **Imagen de consumo en POST-MAN:**

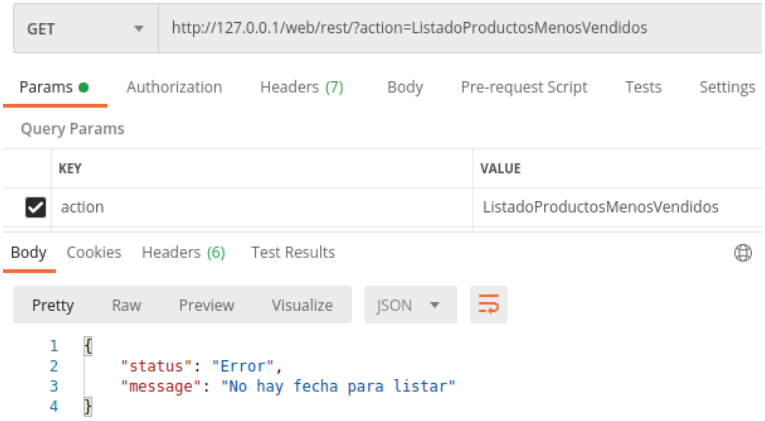


3. Productos menos vendidos el día 1 de marzo.

- **Url consumo**: http://IP_SERVER/rest/?action=ListadoProductosMenosVendidos&fecha=2019-03-01
- **Método**: GET
- **Parámetros**:
    - **fecha**: dato de fecha en formato YYYY-MM-DD.
- **Descripción**: Basados en una fecha se listan los 15 productos que menos se vendieron ese día, se tiene en cuenta la cantidad solicitada de ese producto.
    - **ProductosMenosVendidos**:
        - **Producto_id**: id del producto disponible en el inventario.
        - **producto_nombre**: nombre del producto.
        - **cantidad_vendida**: cantidad que se vendió en la fecha especificada de ese producto.
- **Imagen de consumo en POST-MAN:**



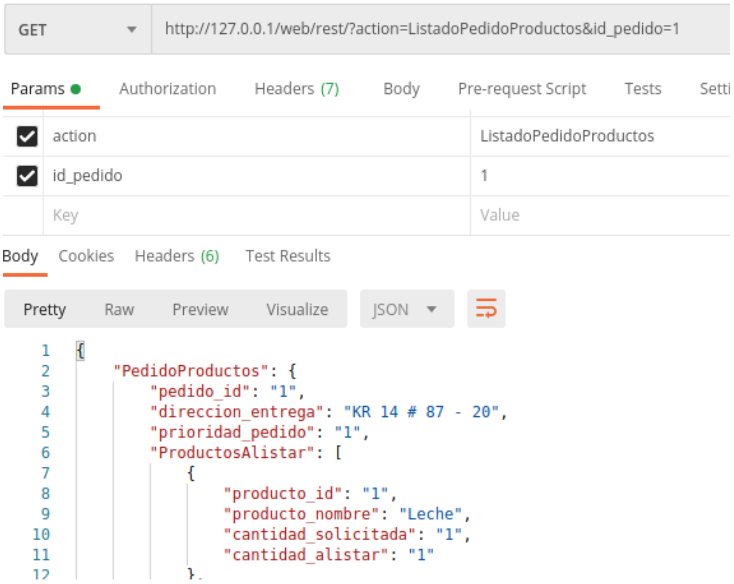
Cundo no se especifica parámetro o su formato es incorrecto:



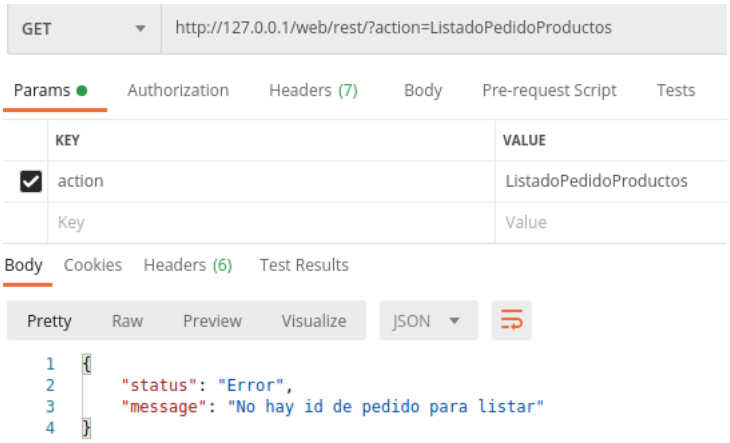
4. Dado el Id de un pedido, saber qué productos y qué cantidad pueden ser alistados según sistema de inventario y cuáles deben ser abastecidos por los proveedores.
   - **Url consumo**: http://IP_SERVER/rest/?action=ListadoPedidoProductos&id_pedido=1
   - **Método**: GET
   - **Parámetros**:
     - **Id_pedido**: dato numérico correspondiente al id del pedido.
   - **Descripción**: Se realizo unca consulta la cual retorna según el id de un pedido que productos y que cantidad de ellos puede ser alistados y que productos y que cantidad de ellos deben ser abastecidos por los proveedores:
     - **PedidoProductos**:
       - **pedido_id**: id del pedido correspondiente a un trasportador.
       - **direccion_entrega**: Dirección entrega del pedido.
       - **prioridad_pedido**: prioridad del pedido.
       - **ProductosAlistar**: arreglo de datos que contiene la información de los productos que pueden ser alistados con su cantidad:
         - **producto_id** : id del producto a alistar
         - **producto_nombre**: nombre del producto a alistar.
         - **cantidad_solicitada** : cantidad que fue solicitada en el pedido original.
         - **cantidad_alistar:** cantidad disponible en el inventario que puede ser alistada

- **ProductosAbastecer**: arreglo de datos que contiene la información de los productos y la cantidad que debe ser abastecida.
  - **producto_id** : id del producto a alistar
  - **producto_nombre**: nombre del producto a alistar.
  - **cantidad_abastecer**: cantidad que debe ser abastecida del producto para poder completar el pedido.

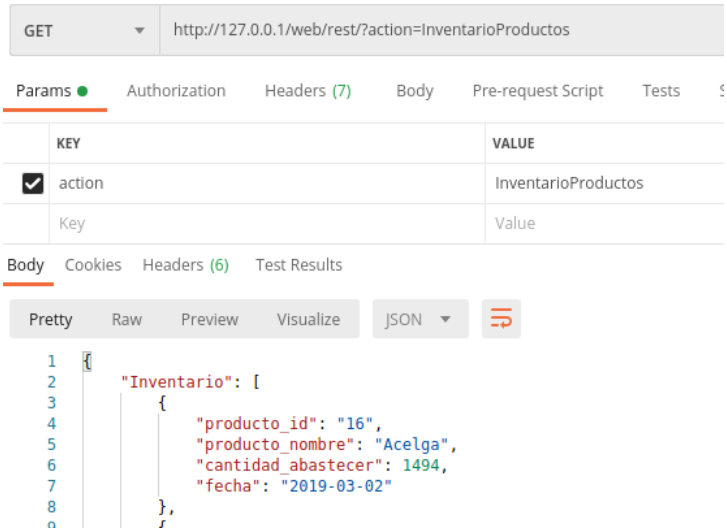
- **Imagen de consumo en POST-MAN:**



- **Cuando no se especifica parámetro o su formato es incorrecto:**



5. Calcular el inventario del día 2 de marzo, teniendo en cuenta los pedidos despachados el 1 de marzo.
   - **Url consumo**: http://127.0.0.1/web/rest/?action=InventarioProductos
   - **Método**: GET
   - **Parámetros**: ninguno
   - **Descripción**: Al no existir información de pedidos para el día 02 de marzo se tomó como base que únicamente serían entregados los pedidos que estarían completos, es decir que tenían todas las unidades disponibles de todos sus productos en el inventario, teniendo en cuenta esto, se realizó el inventario para el día 02 de marzo con los productos que necesitaban ser abastecidos para poder completar los pedidos del 01 de marzo:
     - **Inventario**:
       - **producto_id** : id del producto a alistar
       - **producto_nombre**: nombre del producto a alistar.
       - **cantidad_abastecer**: cantidad que debe ser abastecida del producto. para poder completar el pedido
       - **fecha**: fecha de disponibilidad del producto.
   - **Imagen de consumo en POST-MAN:**



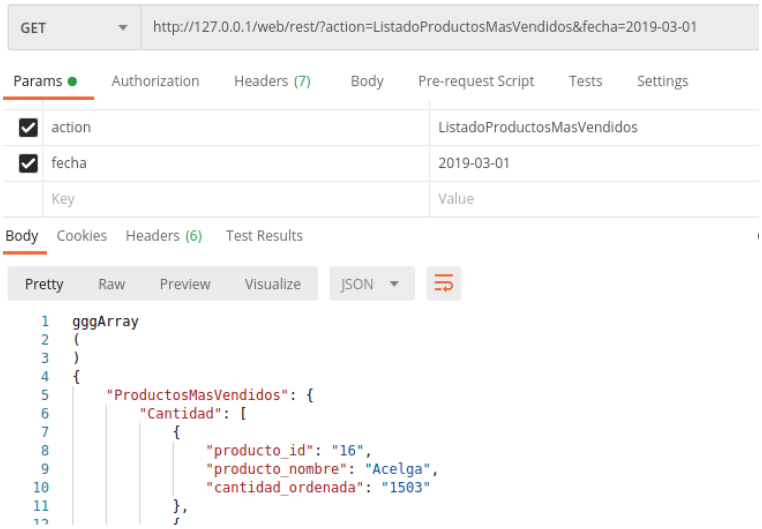
6. Productos más vendidos el día 1 de marzo.
   - **Url consumo**: http://IP_SERVER/rest/?action=ListadoProductosMasVendidos&fecha=2019-03-01
   - **Método**: GET
   - **Parámetros**:
     - **fecha**: dato de fecha en formato YYYY-MM-DD
   - **Descripción**: Basados en una fecha se listan los 15 productos más vendidos teniendo en cuenta su cantidad y su solicitud de la siguiente manera:
     - **ProductosMasVendidos**:
       - **Cantidad**: arreglo que contiene la lista de productos mas vendidos sumando su la cantidad que solicitaron los usuarios.
         - **Producto_id**: id del producto disponible en el inventario.
         - **producto_nombre**: nombre del producto.

- **cantidad_ordenada**: suma de la cantidad que se vendió en la fecha especificada de ese producto.

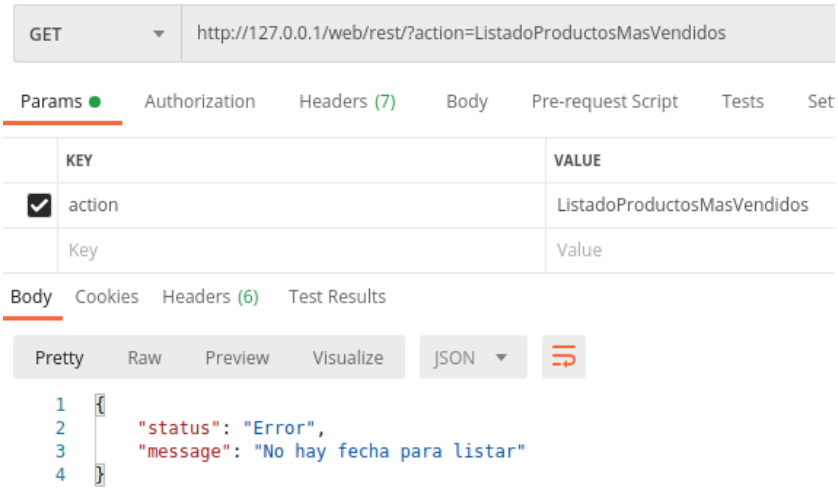
▪ Solicitud: arreglo que contiene la lista de productos más vendidos teniendo en cuenta el numero de veces que fueron solicitados.

- **Producto_id**: id del producto disponible en el inventario.
- **producto_nombre**: nombre del producto.
- **numero_ordenes**: numero de veces que fue ordenado un producto en la fecha dada.

- **Imagen de consumo en POST-MAN:**



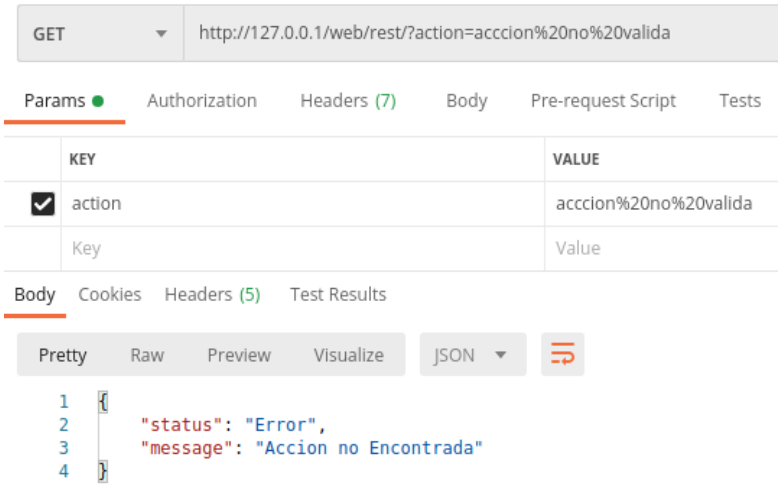
- **Cuando no se especifica parámetro o su formato es incorrecto:**



4.Otras consideraciones:

1.Cuando es una petición invalida

Cuando no se especifica una acción, o se envía una acción invalida el servicio responde de la siguiente forma:



2. códigos de respuesta:

- 500 : Internal Server Error → Se ha producido un error interno
- 422 : Unprocessable Entity → Entidad no procesable
- 400 : Bad Request → La solicitud contiene sintaxis errónea y no debería repetirse
- 204 : No Content → La petición se ha completado con éxito pero su respuesta no tiene