

PRUEBA TECNICA KONECTA

Por: **FABIAN MONTEALEGRE YATE**

Contenido carpeta:

BD: contiene el objeto dumb de la base de datos

Rest : contiene la aplicación en general.

COMO DESPLEGAR LA APP

1. Para este APP es necesario utilizar XAMPP u otro servidor compatible con PHP. Para este caso se utilizó XAMPP.
2. Una vez instalado XAMPP nos dirigimos a la carpeta C:/xampp/htdocs/. Una vez estemos dentro de htdocs, copiamos la carpeta 'apiKonecta'.
3. Iniciamos la ejecución de apache y mysql.
4. Una vez iniciado mysql, desde nuestro navegador nos dirigimos a localhost/phpmyadmin y crearemos una base de datos llamada Konecta. Dentro de esta base de datos vamos a importar el archivo konecta.sql que se encuentra en la carpeta BD de nuestro proyecto.
5. Desde un cliente como postman(utilizado en este proyecto) utilizaremos los parámetros correspondiente como se explicara más adelante.

BASE DE DATOS

Motor utilizado: MYSQL

La base de datos está conformada por 3 tablas, las cuales son:

- producto: tabla que almacena todos los productos del inventario.
- ventaProducto: tabla que almacena las ventas del día.
- ventas: tabla intermedia que almacena el precio total de una venta y está directamente relacionada con la tabla ventaProducto.

FUNCIONAMIENTO DE LA APP

Lenguaje desarrollado: PHP

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

- Capa DAL: la capa de acceso 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: o Index: Es el archivo raíz que instancia la clase del api.

METODOS DE LA API

LISTAR INVENTARIO

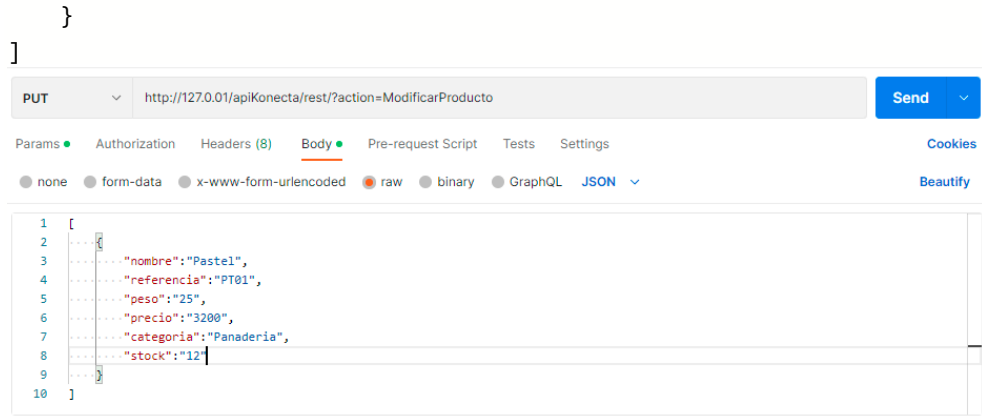
- Url consumo: <http://127.0.0.1/apiKonecta/rest/?action=ListadoProductoInventario>
- Método: GET
- Parámetros: ninguno
- Descripción: Se realizo una consulta la base de datos la cual retorna una json con los siguientes datos:

```
"ProductosInventario": [
  {
    "nombre": "Brownie",
    "referencia": "BR01",
    "peso": "5",
    "precio": "2500",
    "categoria": "Panaderia",
    "stock": "27"
  }
]
```

MODIFICACION DE UNO O VARIOS PRODUCTO

- Url consumo: <http://127.0.0.1/apiKonecta/rest/?action=ModificarProducto>
- Método: PUT
- Parámetros: Se debe agregar un body-raw tipo Json de la siguiente forma:

```
[
  {
    "nombre": "Pastel",
    "referencia": "PT01",
    "peso": "25",
    "precio": "3200",
    "categoria": "Panaderia",
    "stock": "12"
  }
]
```



- Descripción: Se realizó una petición a la base de datos donde se haga un update teniendo como llave el parámetro 'referencia', esto nos devuelve la siguiente respuesta:

```

{
  "status": "OK",
  "message": "Actualizacion realizada con exito"
}

```

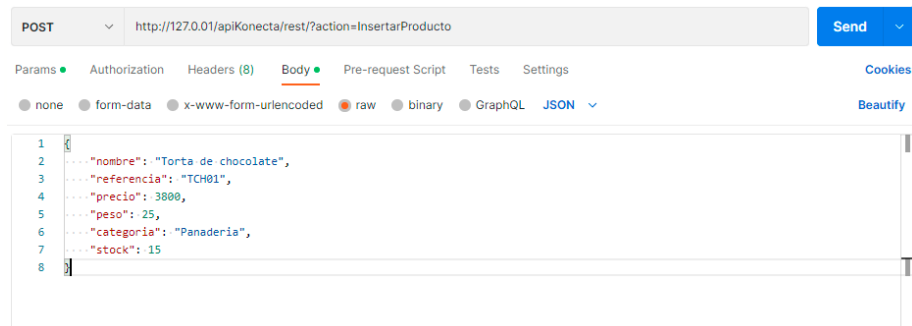
INSERTAR UN PRODUCTO

- Url consumo: <http://127.0.0.1/apiKonecta/rest/?action=InsertarProducto>
- Método: POST
- Parámetros: Se debe agregar un body-raw tipo Json de la siguiente forma:

```

{
  "nombre": "Torta de chocolate",
  "referencia": "TCH01",
  "precio": 3800,
  "peso": 25,
  "categoria": "Panaderia",
  "stock": 15
}

```



- Descripción: Se realiza un insert a la base de datos, específicamente a la tabla productos (por el momento solo se permite agregar un producto a la vez), esto nos devuelve lo siguiente:

```

{

```

```
"status": "Success",  
"message": "Insertado correctamente"  
}
```

ELIMINAR UNO PRODUCTO O VARIOS

- Url consumo: <http://127.0.0.1/apiKonecta/rest/?action=EliminarProducto>
- Método: DELETE
- Parámetros: Se debe agregar un body-raw tipo Json de la siguiente forma:

```
[  
  {  
    "referencia": "JGM01"  
  },  
  {  
    "referencia": "JGM02"  
  }  
]
```

- Descripción: Se realiza un delete a la base de datos, en donde la llave principal es la referencia, esto nos devuelve los siguientes datos:

```
{  
  "status": "OK",  
  "message": "Producto eliminado"  
}
```

HACER UNA VENTA

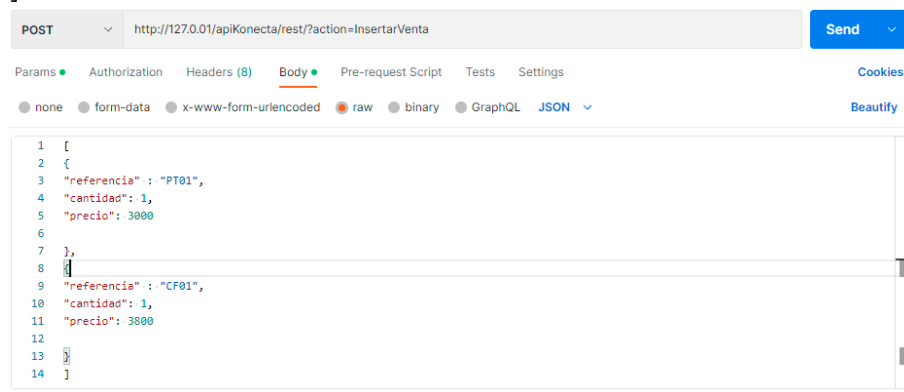
- Url consumo: <http://127.0.0.1/apiKonecta/rest/?action=InsertarVenta>
- Método: POST
- Parámetros: Se debe agregar un body-raw tipo Json de la siguiente forma:

```
[  
  {  
    "referencia": "PT01",  
    "cantidad": 1,  
    "precio": 3000  
  },  
]
```

```

{
  "referencia" : "CF01",
  "cantidad": 1,
  "precio": 3800
}
]

```



- Descripción: Para crear una venta, se pueden agregar uno o varios productos teniendo en cuenta los parámetros anteriores. Esto internamente interactúa con la tabla ventas y ventas producto, retornando lo siguiente:

```

{
  "status": "OK",
  "message": "Compra realizada con exito"
}

```