

Universidad De San Carlos De Guatemala

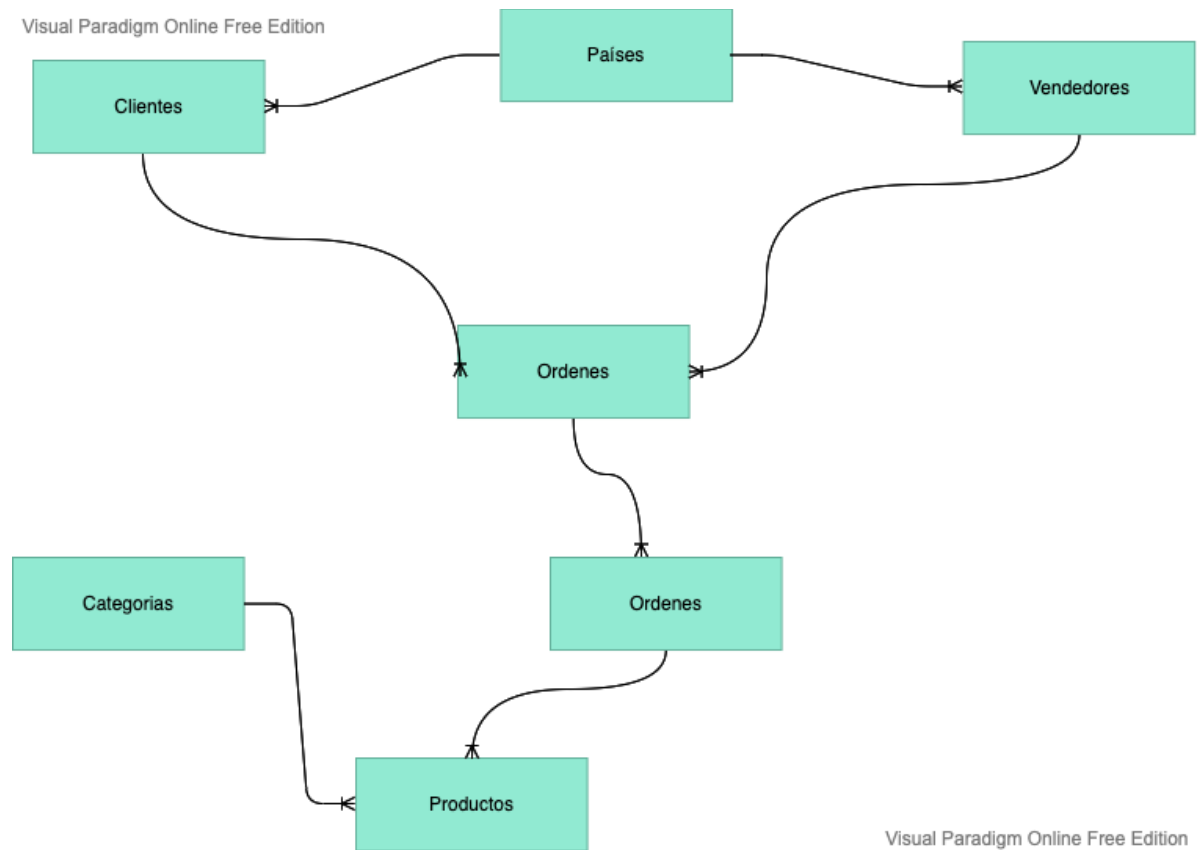
Facultad De Ingeniería, Bases De Datos 1

Carlos Javier Martinez Polanco

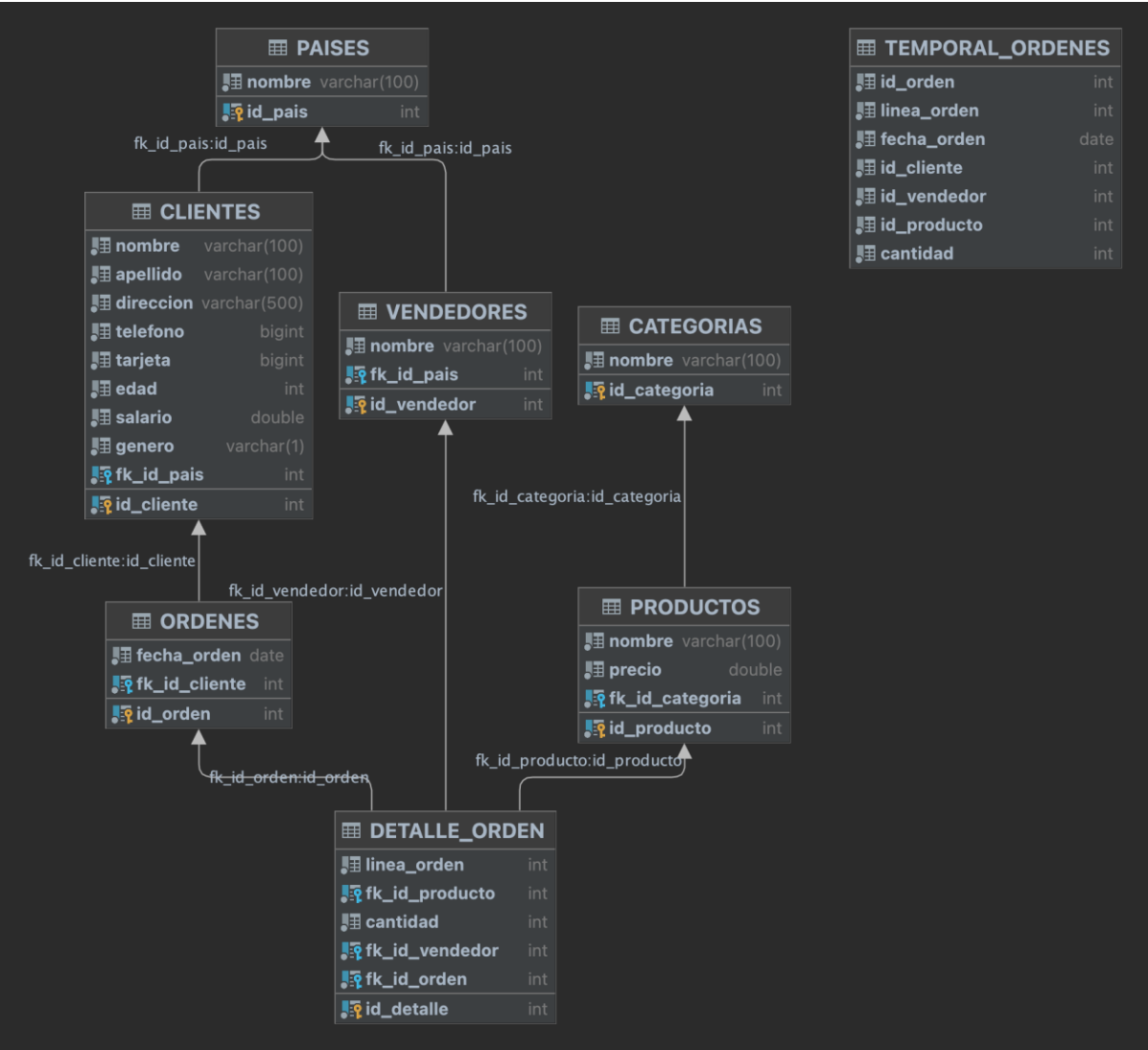
201709282

PROYECTO 1

MODELO CONCEPTUAL



MODELO LÓGICO



MODELO FÍSICO

TABLA CATEGORÍAS

Esta tabla almacena las categorías a las que pertenecen los productos de la base de datos.

```
CREATE TABLE CATEGORIAS (  
  id_categoria INT NOT NULL PRIMARY KEY ,  
  nombre VARCHAR(100) NOT NULL  
);
```

TABLA PAISES

Esta tabla almacena los diferentes países donde se encuentran los vendedores o tiendas, además de los clientes.

```
CREATE TABLE PAISES(  
  id_pais INT NOT NULL PRIMARY KEY ,  
  nombre VARCHAR(100) NOT NULL  
);
```

TABLA PRODUCTOS

Esta tabla almacena los diferentes productos que se encuentran disponibles en las tiendas.

```
CREATE TABLE PRODUCTOS(  
  id_producto INT NOT NULL PRIMARY KEY ,  
  nombre VARCHAR(100) NOT NULL ,  
  precio DOUBLE NOT NULL ,  
  fk_id_categoria INT NOT NULL ,  
  FOREIGN KEY (fk_id_categoria) REFERENCES CATEGORIAS(id_categoria)  
);
```

TABLA VENDEDORES

Esta tabla contiene la información personal de los vendedores.

```
CREATE TABLE VENDEDORES(  
  id_vendedor INT NOT NULL PRIMARY KEY ,  
  nombre VARCHAR(100) NOT NULL ,  
  fk_id_pais INT NOT NULL ,  
  FOREIGN KEY (fk_id_pais) REFERENCES PAISES(id_pais)  
);
```

TABLA CLIENTES

Esta tabla contiene la información personal de los clientes.

```
CREATE TABLE CLIENTES(  
  id_cliente INT NOT NULL PRIMARY KEY ,  
  nombre VARCHAR(100) NOT NULL ,  
  apellido VARCHAR(100) NOT NULL ,  
  direccion VARCHAR(500) NOT NULL ,  
  telefono BIGINT NOT NULL ,  
  tarjeta BIGINT NOT NULL ,  
  edad INT NOT NULL ,  
  salario DOUBLE NOT NULL ,  
  genero VARCHAR(1) NOT NULL ,  
  fk_id_pais INT NOT NULL ,  
  FOREIGN KEY (fk_id_pais) REFERENCES PAISES(id_pais)  
);
```

TABLA ORDENES

Esta tabla almacena información relevante a la orden de los productos que desea comprar el cliente.

```
CREATE TABLE ORDENES(  
  id_orden INT NOT NULL PRIMARY KEY ,  
  fecha_orden DATE NOT NULL ,  
  fk_id_cliente INT NOT NULL ,  
  FOREIGN KEY (fk_id_cliente) REFERENCES CLIENTES(id_cliente)  
);
```

TABLA DETALLE_ORDEN

Esta tabla almacena información relevante a los productos en la orden que el cliente ha realizado.

```
CREATE TABLE DETALLE_ORDEN(  
  id_detalle INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  linea_orden INT NOT NULL ,  
  fk_id_producto INT NOT NULL ,  
  cantidad INT NOT NULL ,  
  fk_id_vendedor INT NOT NULL ,  
  fk_id_orden INT NOT NULL ,  
  FOREIGN KEY (fk_id_producto) REFERENCES PRODUCTOS(id_producto),  
  FOREIGN KEY (fk_id_vendedor) REFERENCES VENDEDORES(id_vendedor),  
  FOREIGN KEY (fk_id_orden) REFERENCES ORDENES(id_orden)  
);
```

TABLA TEMPORAL_ORDENES

Esta tabla almacena temporalmente los datos que se insertan en las tablas ORDENES y DETALLE ORDEN luego de un procedimiento de filtrado de la información para proteger la integridad de la base de datos que son cargados desde el archivo csv.

```
CREATE TABLE TEMPORAL_ORDENES(  
  id_orden INT NOT NULL ,  
  linea_orden INT NOT NULL ,  
  fecha_orden VARCHAR(50) NOT NULL ,  
  id_cliente INT NOT NULL ,  
  id_vendedor INT NOT NULL ,  
  id_producto INT NOT NULL ,  
  cantidad INT NOT NULL  
);
```

REST API

Para el desarrollo del proyecto se ha utilizado la arquitectura cliente-servidor, la cual consta de un Backend elaborado con Node JS, con la funcionalidad de REST API. Esta consta de 10 endpoints que realizan una petición GET que ejecuta las consultas a nuestra base de datos previamente cargada en el servidor. Se utilizó como gestor de bases de datos MySQL, el servidor se levanta con un contenedor de Docker por medio de un Dockerfile para la creación de la imagen y Docker-compose para configurar dicho contenedor. Para ejecutar los endpoints se utiliza la herramienta POSTMAN que nos permite hacer peticiones http a nuestro servidor por medio del localhost en el puerto 3000.

ENDPOINTS

1. # CONSULTA 1 Mostrar el cliente que más ha comprado.
Se debe de mostrar el id del cliente, nombre, apellido, país y monto total.

http://localhost:3000/p1_bases1/api/consulta1

2. # CONSULTA 2 Mostrar el producto más y menos comprado.
Se debe mostrar el id del producto, nombre del producto, categoría, cantidad de unidades y monto vendido.

http://localhost:3000/p1_bases1/api/consulta2

3. # CONSULTA 3 Mostrar a la persona que más ha vendido.
Se debe mostrar el id del vendedor, nombre del vendedor, monto total vendido.

http://localhost:3000/p1_bases1/api/consulta3

4. # CONSULTA 4 Mostrar el país que más y menos ha vendido.
Debe mostrar el nombre del país y el monto. (Una sola consulta).

http://localhost:3000/p1_bases1/api/consulta4

5. # CONSULTA 5 Top 5 de países que más han comprado en orden ascendente.
Se le solicita mostrar el id del país, nombre y monto total.

http://localhost:3000/p1_bases1/api/consulta5

6. # CONSULTA 6 Mostrar la categoría que más y menos se ha comprado.
Debe de mostrar el nombre de la categoría y cantidad de unidades. (Una sola consulta).

http://localhost:3000/p1_bases1/api/consulta6

7. # CONSULTA 7 Mostrar la categoría más comprada por cada país.
Se debe de mostrar el nombre del país, nombre de la categoría y cantidad de unidades.

http://localhost:3000/p1_bases1/api/consulta7

8. # CONSULTA 8 Mostrar las ventas por mes de Inglaterra.
Debe de mostrar el número del mes y el monto.

http://localhost:3000/p1_bases1/api/consulta8

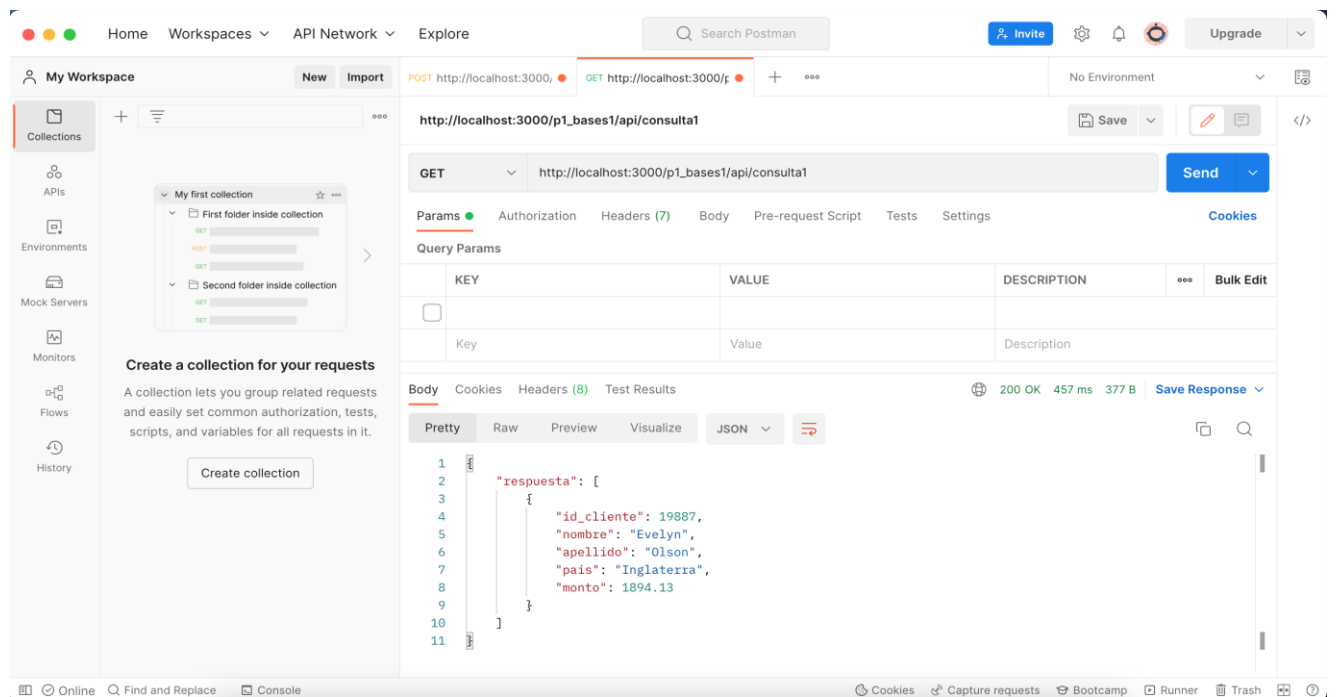
9. # CONSULTA 9 Mostrar el mes con más y menos ventas.
Se debe de mostrar el número de mes y monto. (Una sola consulta).

http://localhost:3000/p1_bases1/api/consulta9

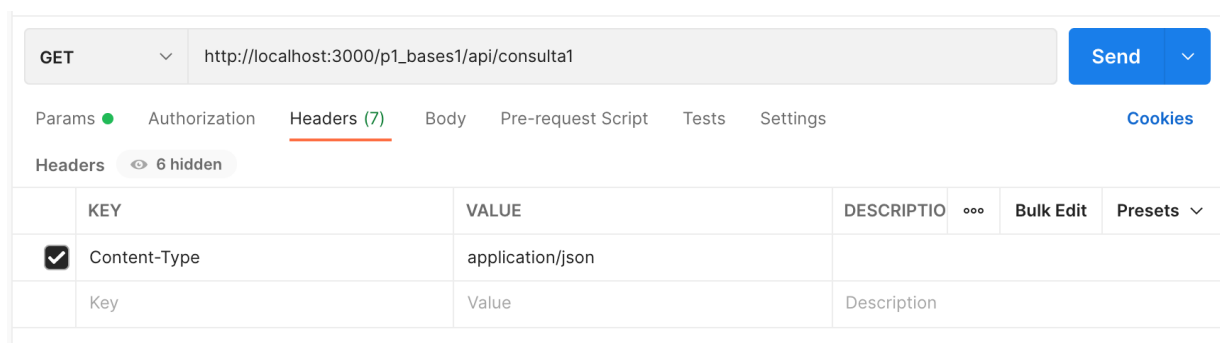
10. # CONSULTA 10 Mostrar las ventas de cada producto de la categoría deportes.
Se debe de mostrar el id del producto, nombre y monto.

http://localhost:3000/p1_bases1/api/consulta10

EJEMPLO POSTMAN



Se debe ingresar la dirección a la cual haremos la petición en POSTMAN, se debe configurar con el método GET y en la sección de Headers debe utilizarse como KEY: Content-Type y VALUE: application/json



CODIGO: https://github.com/2730538920101/P1_BASES1_201709282.git