



MANUAL TÉCNICO

consultas para presentar la información requerida de una base de datos relacional

“El manual técnico es el que proporciona todas las pautas de configuración y la lógica con la que se ha desarrollado en el programa, con el objetivo de que cualquier programador entienda la idea propia y su estructura; por lo que se considera necesario ser documentada.”

Elaborado por:

Alvaro Emmanuel Socop Pérez
UNIVERSIDAD DE SAN CARLOS - Facultad de Ingeniería

202000194
18 DE AGOSTO DE 2022



ÍNDICE

Introducción.....	2.
Objetivos y alcances del sistema.....	3.
Especificaciones Técnicas.....	4.
Requisitos de Hardware.....	4.
Requisitos de software.....	4.
Sistema operativo.....	4.
Lenguaje de Programación e IDE.....	5.
Tecnologías utilizadas (Lógica del programa).....	5
Funciones utilizadas	7
Flujo del programa.....	9
Descripción del flujo.....	9
Autómata	10
Método del Arbol	11

INTRODUCCIÓN

El presente manual técnico tiene como finalidad describir la estructura y diseño del programa consultas para presentar la información requerida de una base de datos relacional, así como dar explicación de los métodos, clases y procesos de cada apartado del programa y la modificación que se le podría dar para cualquier finalidad como el de mejorar el sistema o cambiar algunos atributos propios del analizador. El sistema cuenta con implementación de varias librerías propias de python como MySQLdb, entre otros, como parte del conocimiento adquirido en los laboratorios de Bases de datos, en base a ello trataré de explicar en que métodos se fueron utilizando y como hacen funcionar el programa en los distintos sistemas operativos.

La implementación de librerías externas a Python también fueron una opción para poder realizar la estructura de los endpoints y para elegir los archivos necesarios, también se explicará paso a paso cual es el camino para entender perfectamente la arquitectura del programa. Resulta ser bastante fácil de implementar los ciclos y condiciones, también los métodos para consumir la API puesto que es una función que viene integrada con la paquetería por defecto de Flask de python y así solo instanciarlos importando las librerías y utilizarlas.

OBJETIVOS Y ALCANCES DEL PROGRAMA

Generales

- ✓ Aplicar conocimientos de programación básica en el lenguaje de programación SQL.
- ✓ Utilizar consultas para presentar la información requerida de una base de datos relacional

Específicos

- ✓ Desarrollar la habilidad de generar consultas avanzadas en el lenguaje SQL.
- ✓ Mostrar información específica de una base de datos.
- ✓ Aprender a realizar cargas masivas desde archivos de Excel a una base de datos.

Dirigido hacia los auxiliares y catedráticos encargados del Laboratorio de Bases de datos 1 y a usuarios de cualquier edad como forma de auxiliatura en el aprendizaje de Bases de datos con SQL para saber lo que podemos hacer con la programación.

ESPECIFICACIÓN TÉCNICA

Requisitos de Hardware

Memoria mínima	512 MB
Memoria recomendada	1 GB
Espacio en disco mínimo	250 MB de espacio libre
Espacio en disco recomendado	500 MB de espacio libre
MVP	Python 3.6

Requisitos de software

- **Sistema operativo**

Windows

- Windows 10 (8u51 y superiors)
- Tener instalado el programa de SQL y Python
- RAM: 128 MB
- Espacio en disco: 124 MB
- Procesador: Mínimo Pentium 2 a 266 MHz
- Algún explorador de internet

Mac OS X

- Tener instalado el programa SQL y Python
- Explorador de 64 bits
- Se requiere un explorador de 64 bits (Safari, Firefox, por ejemplo) para ejecutar Oracle Java en Mac OS X.

Linux

- Oracle Linux 5.5+1
- Oracle Linux 6.x (32 bits), 6.x (64 bits)2
- Exploradores: Firefox

- **Lenguaje de Programación e IDE**

Para el desarrollo del programa se utilizó el lenguaje de Programación PYTHON y SQL, el IDE Visual Studio Code.

- **Tecnologías utilizadas**

Python es la plataforma estándar que se utilizó para el desarrollo del programa el cual recoge todas las funcionalidades básicas del lenguaje.

Dentro de estas funcionalidades básicas de python encontramos: el uso de colecciones, acceso a ficheros con IO, con la característica de que es un lenguaje de programación multiparadigma, librerías para el desarrollo de aplicaciones de bases de datos como MySQLdb, capacidades para realizar conexiones de un motor de bases de datos, manejo de contenido XSLX...



LÓGICA DEL PROGRAMA

ESQUEMA CONCEPTUAL:

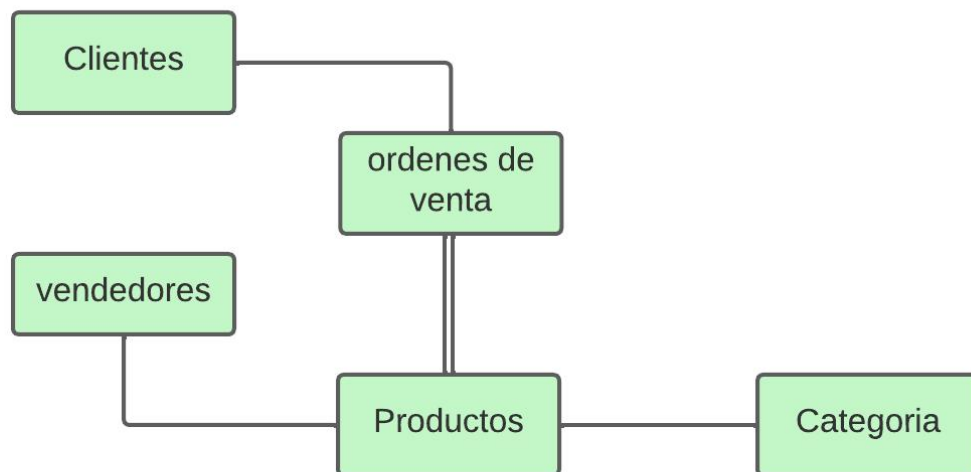
Descripción del proyecto

El buen samaritano lo ha contratado a usted para hacer el diseño y la implementación de una base de datos que permita apoyar con la gestión de un sistema de ventas. La empresa necesita llevar un control de los productos, clientes, vendedores, ordenes de ventas.

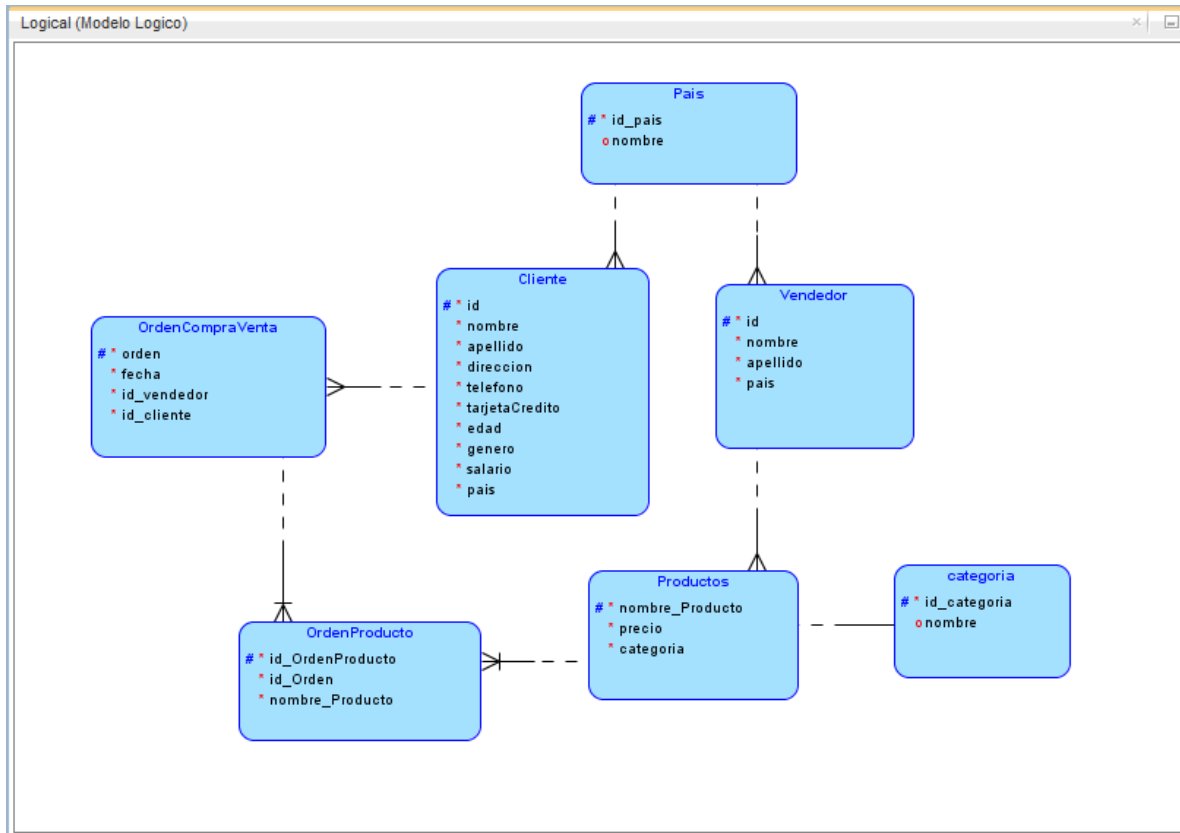
Se sabe que el buen samaritano cuenta con una amplia variedad de productos de los cuales se necesita almacenar nombre del producto, precio y la categoría del producto.

Un vendedor tiene un identificador único, nombre, apellido y país. Un Cliente tiene un identificador, nombre, apellido, dirección, teléfono, tarjeta de crédito, edad, genero, salario y país.

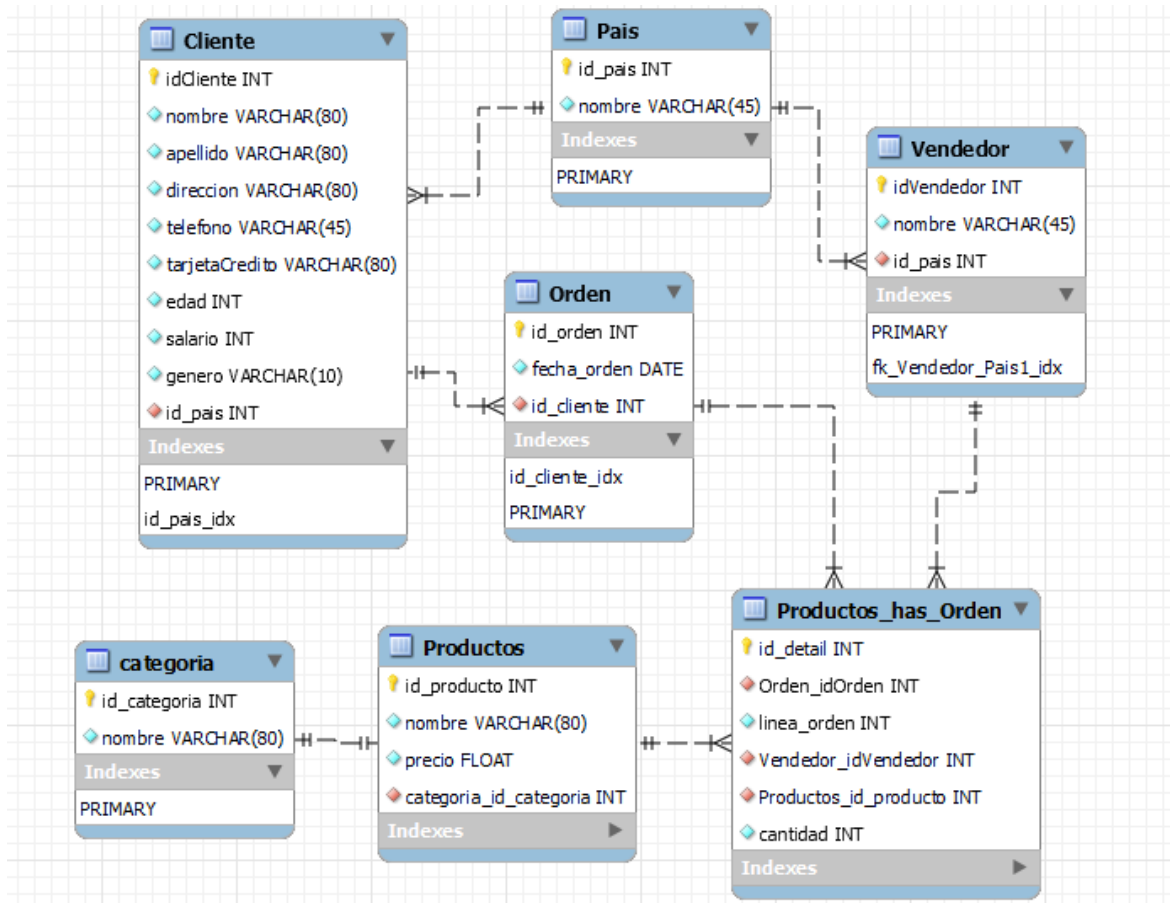
Por razones de contabilidad, se debe de registrar la información de cada orden de compra con un numero de orden, la fecha de la orden, el id del vendedor y el id del cliente, así como también cada uno de los productos correspondientes a la compra.



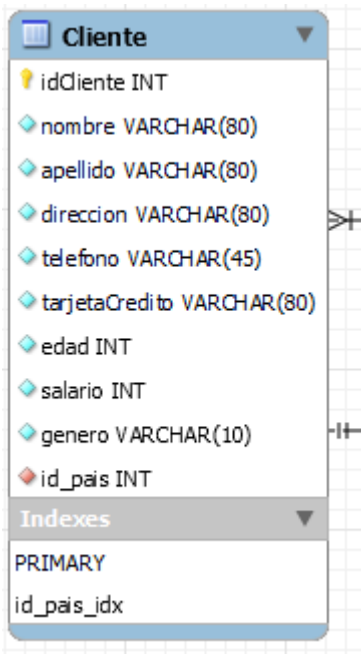
ESQUEMA LOGICO:



ESQUEMA FISICO:



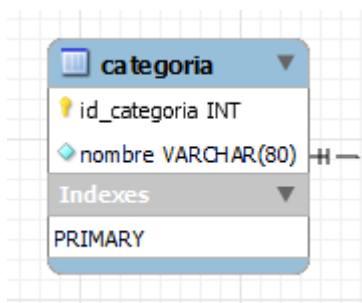
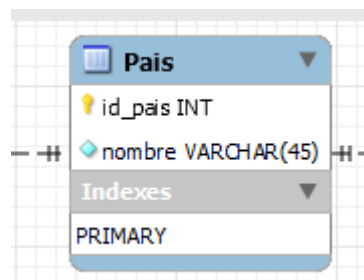
DESCRIPCION DE TABLAS:



Almacena los datos del cliente y se relaciona de uno a muchos con la tabla Orden,

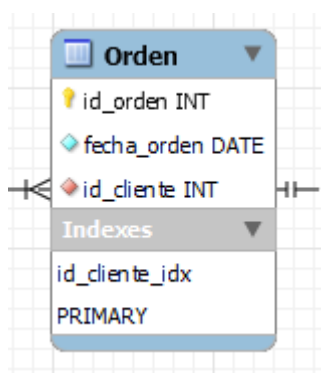
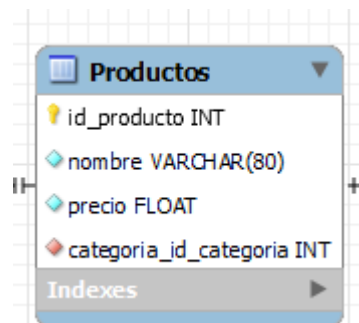
Se tiene como llave foránea a id_pais para hacer referencia a los países que pertenece cada cliente.

El país relaciona a muchos con los clientes y los vendedores



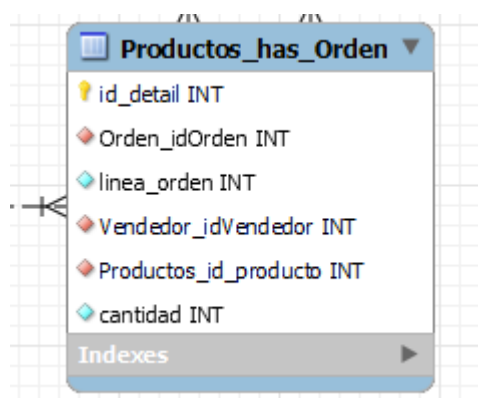
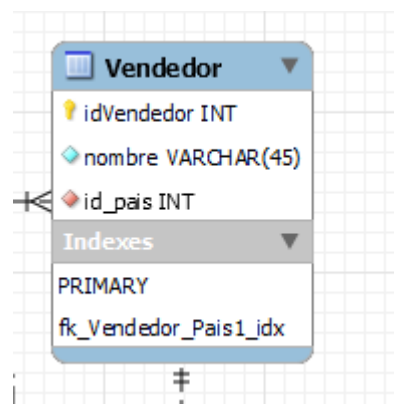
En la categoría solo tenemos relación de uno a uno ya que solo se tiene a los productos divididos en categorías.

Para los productos se tiene como llave foránea un `id_categoria` que referencia a otra llave de la tabla `Productos_has_Orden` la cual relaciona a el Orden y los vendedores .



Varias ordenes están en un cliente así que se relaciona de uno a muchos desde clientes y luego se relaciona una orden a muchos en `Productos_has_Orden` ya que allí se tienen llaves foráneas para acceder a estos datos.

El vendedor relaciona a país porque varios vendedores pertenecen a cierto país, también se relaciona con `Productos_has_Orden` porque necesitamos que se tenga una relación en conjunto con las demás tablas como Orden y Productos.



Esta tabla es la que relaciona tanto a las Ordenes Vendedores y Productos y rompe con toda relación muchos a muchos con el id de los tres y creando un `id_detail` para identificarla.

DESCRIPCION DE LA API:

Prácticamente se cargaron los datos usando Pandas de Python y agregándolas a varias listas.

```
from flask import Flask, jsonify, request...
list = [];
categoria = []
cliente = []
orden = []
pais = []
producto = []
vendedor = []

#! ██████████ WELCOME TO MY PROJECT ██████████

def takeEXCEL():...

#* ██████████ CONNECT WITH DATABASE: ██████████

connection = pymysql.connect(host='localhost',user='root',password='2412',db='mydb')
# charset='utf8mb4',
# cursorclass=pymysql.cursors.DictCursor

#* ██████████ INSERT ALL DATA IN MYSQL MOTOR: ██████████

def BulkInsert():...

'''---COMENTARIO TOQUE---'''
app = Flask(__name__)
CORS(app)

#! ██████████ ENDPOINTS ██████████
#? ██████████ 1 ██████████
# 1. Mostrar el cliente que más ha comprado. Se debe de mostrar el id del cliente,
# nombre, apellido, país y monto total.
@app.route('/1', methods=['GET'])
def uno():...
```

Luego se fueron cargando con pymysql seteando consultas de datos

```
# nombre, apellido, país y monto total.
@app.route('/1', methods=['GET'])
> def uno():...
    # finally:
    # connection.close()

#? 2
# 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.
@app.route('/2', methods=['GET'])
> def dos():...
    # finally:
    # connection.close()
# 3. Mostrar a la persona que más ha vendido. Se debe mostrar el id del
# vendedor, nombre del vendedor, monto total vendido.
@app.route('/3', methods=['GET'])
> def tres():...
    # finally:
    # connection.close()
# 4. Mostrar el país que más y menos ha vendido. Debe mostrar el nombre del
# país y el monto. (Una sola consulta).
@app.route('/4', methods=['GET'])
> def cuatro():...
    # finally:
    # connection.close()
# 5. Top 5 de países que más han comprado en orden ascendente. Se le solicita
# mostrar el id del país, nombre y monto total.
@app.route('/5', methods=['GET'])
> def cinco():...
    # finally:
    # connection.close()
# 6. Mostrar la categoría que más y menos se ha comprado. Debe de mostrar el
```

Por ultimo usando `cursor.execute` se realizaron las consultas y se reciben los datos para generar JSONS de salida.

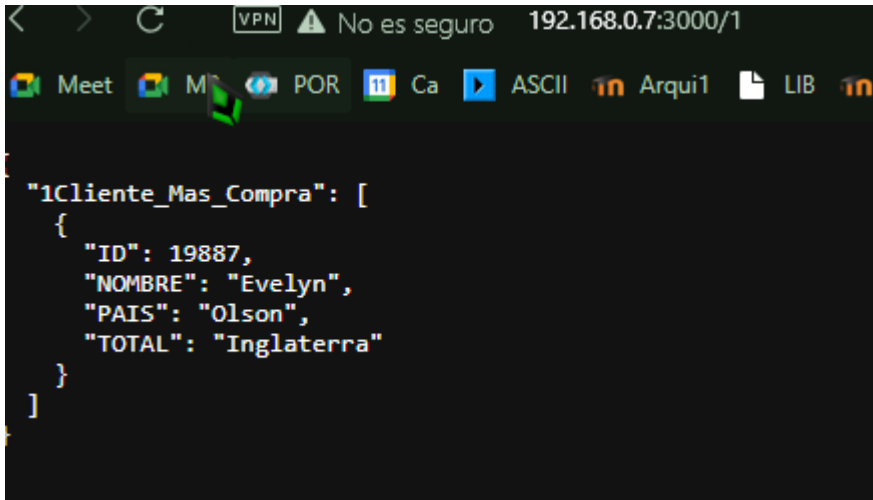
```

# 5. Top 5 de países que más han comprado en orden ascendente. Se debe mostrar el
# mostrar el id del país, nombre y monto total.
@app.route('/5', methods=['GET'])
> def cinco():...
# finally:
# connection.close()
# 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).
@app.route('/6', methods=['GET'])
> def seis():...
# finally:
# connection.close()
# 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.
@app.route('/7', methods=['GET'])
> def siete():...
# finally:
# connection.close()
# 8. Mostrar las ventas por mes de Inglaterra. Debe de mostrar el número del mes
# y el monto.
@app.route('/8', methods=['GET'])
> def ocho(): You, 7 days ago • ADDING QUERIES ...
# finally:
# connection.close()
# 9. Mostrar el mes con más y menos ventas. Se debe de mostrar el número de
# mes y monto. (Una sola consulta).
@app.route('/9', methods=['GET'])
> def nueve():...
# finally:
# connection.close()
# 10. Mostrar las ventas de cada producto de la categoría deportes. Se debe de
# mostrar el id del producto, nombre y monto.
@app.route('/10', methods=['GET'])

```

DESCRIPCIÓN DE LOS ENDPOINTS UTILIZADOS

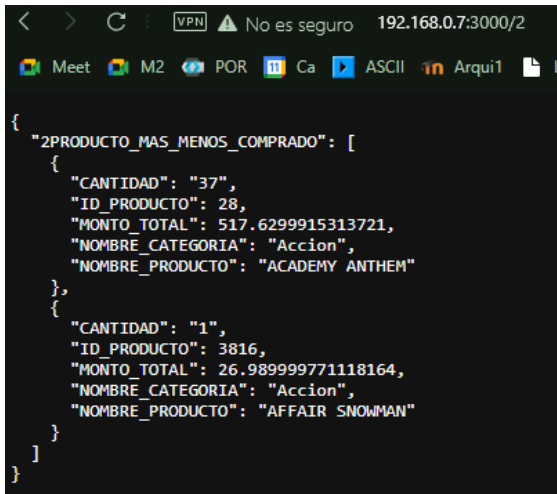
1. Se muestra en este ENDPOINT (Consulta) el cliente que más ha comprado. Se debe de mostrar el id del cliente, nombre, apellido, país y monto total.



```
{
  "1Cliente_Mas_Compra": [
    {
      "ID": 19887,
      "NOMBRE": "Evelyn",
      "PAIS": "Olson",
      "TOTAL": "Inglaterra"
    }
  ]
}
```

2. Se muestra en este ENDPOINT (Consulta) el producto más y menos comprado. Se debe mostrar el id del producto, nombre del producto, categoría, cantidad de unidades y monto

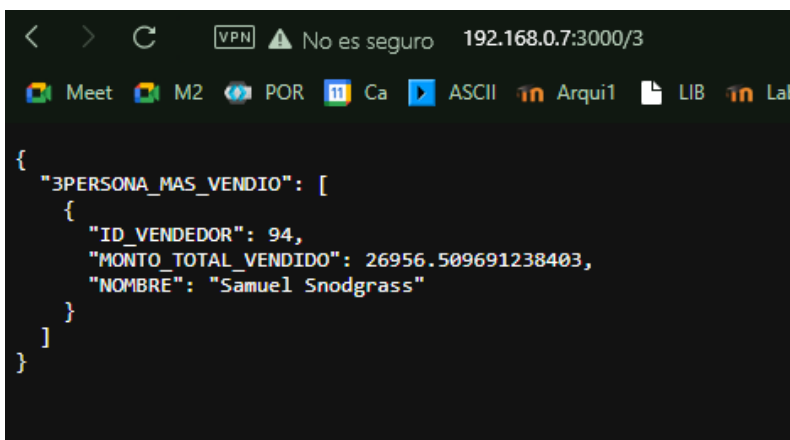
vendido.



```
{
  "2PRODUCTO_MAS_MENOS_COMPRADO": [
    {
      "CANTIDAD": "37",
      "ID_PRODUCTO": 28,
      "MONTO_TOTAL": 517.6299915313721,
      "NOMBRE_CATEGORIA": "Accion",
      "NOMBRE_PRODUCTO": "ACADEMY ANTHEM"
    },
    {
      "CANTIDAD": "1",
      "ID_PRODUCTO": 3816,
      "MONTO_TOTAL": 26.989999771118164,
      "NOMBRE_CATEGORIA": "Accion",
      "NOMBRE_PRODUCTO": "AFFAIR SNOWMAN"
    }
  ]
}
```

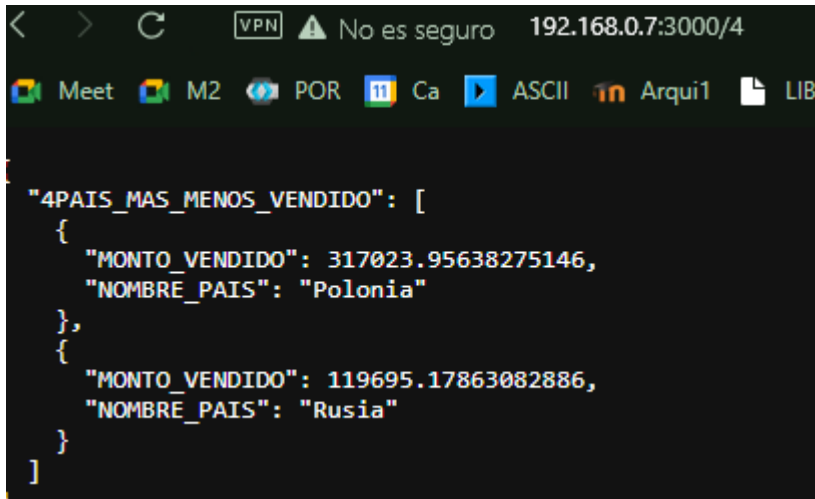
3. Se muestra en este ENDPOINT (Consulta) a la persona que más ha vendido. Se debe mostrar el id del

vendedor, nombre del vendedor, monto total vendido.



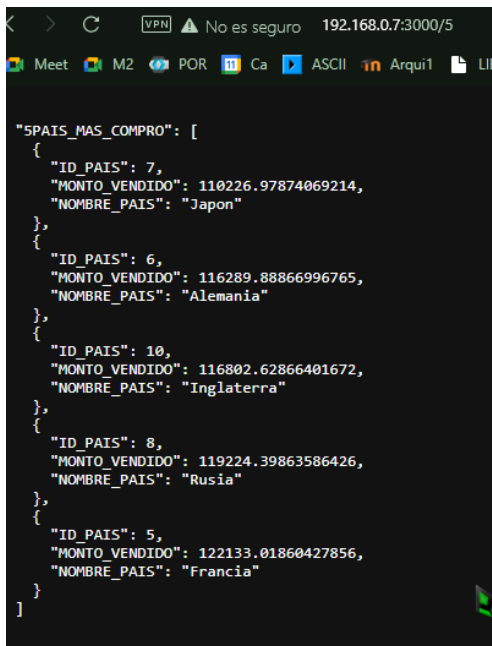
```
{
  "3PERSONA_MAS_VENDIO": [
    {
      "ID_VENDEDOR": 94,
      "MONTO_TOTAL_VENDIDO": 26956.509691238403,
      "NOMBRE": "Samuel Snodgrass"
    }
  ]
}
```


4. Se muestra en este ENDPOINT (Consulta) el país que más y menos ha vendido. Debe mostrar el nombre del país y el monto. (Una sola consulta).



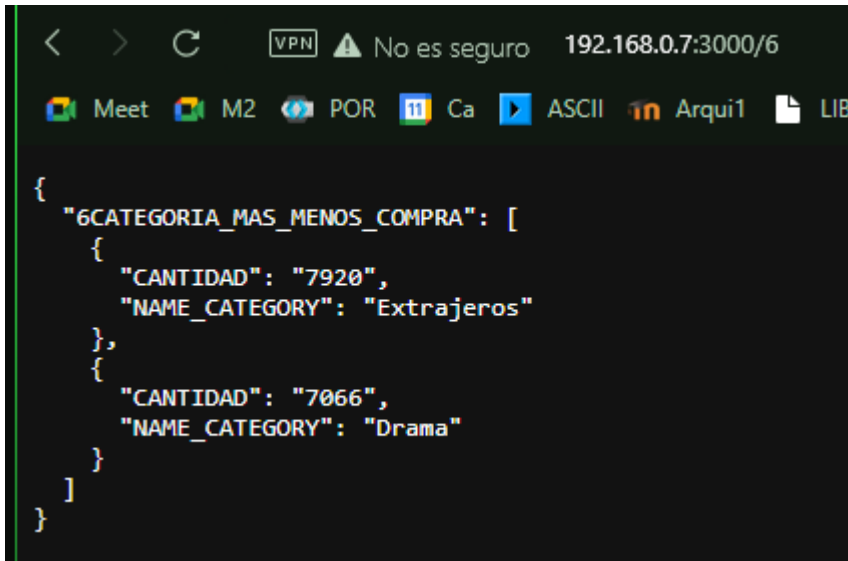
```
4PAIS_MAS_MENOS_VENDIDO: [
  {
    "MONTO_VENDIDO": 317023.95638275146,
    "NOMBRE_PAIS": "Polonia"
  },
  {
    "MONTO_VENDIDO": 119695.17863082886,
    "NOMBRE_PAIS": "Rusia"
  }
]
```

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



```
5PAIS_MAS_COMPRO: [
  {
    "ID_PAIS": 7,
    "MONTO_VENDIDO": 110226.97874069214,
    "NOMBRE_PAIS": "Japon"
  },
  {
    "ID_PAIS": 6,
    "MONTO_VENDIDO": 116289.88866996765,
    "NOMBRE_PAIS": "Alemania"
  },
  {
    "ID_PAIS": 10,
    "MONTO_VENDIDO": 116802.62866401672,
    "NOMBRE_PAIS": "Inglaterra"
  },
  {
    "ID_PAIS": 8,
    "MONTO_VENDIDO": 119224.39863586426,
    "NOMBRE_PAIS": "Rusia"
  },
  {
    "ID_PAIS": 5,
    "MONTO_VENDIDO": 122133.01860427856,
    "NOMBRE_PAIS": "Francia"
  }
]
```

6. Se muestra en este ENDPOINT (Consulta) la categoría que más y menos se ha comprado. Debe de mostrar el nombre de la categoría y cantidad de unidades. (Una sola consulta).



```
{
  "6CATEGORIA_MAS_MENOS_COMPRA": [
    {
      "CANTIDAD": "7920",
      "NAME_CATEGORY": "Extrajeros"
    },
    {
      "CANTIDAD": "7066",
      "NAME_CATEGORY": "Drama"
    }
  ]
}
```

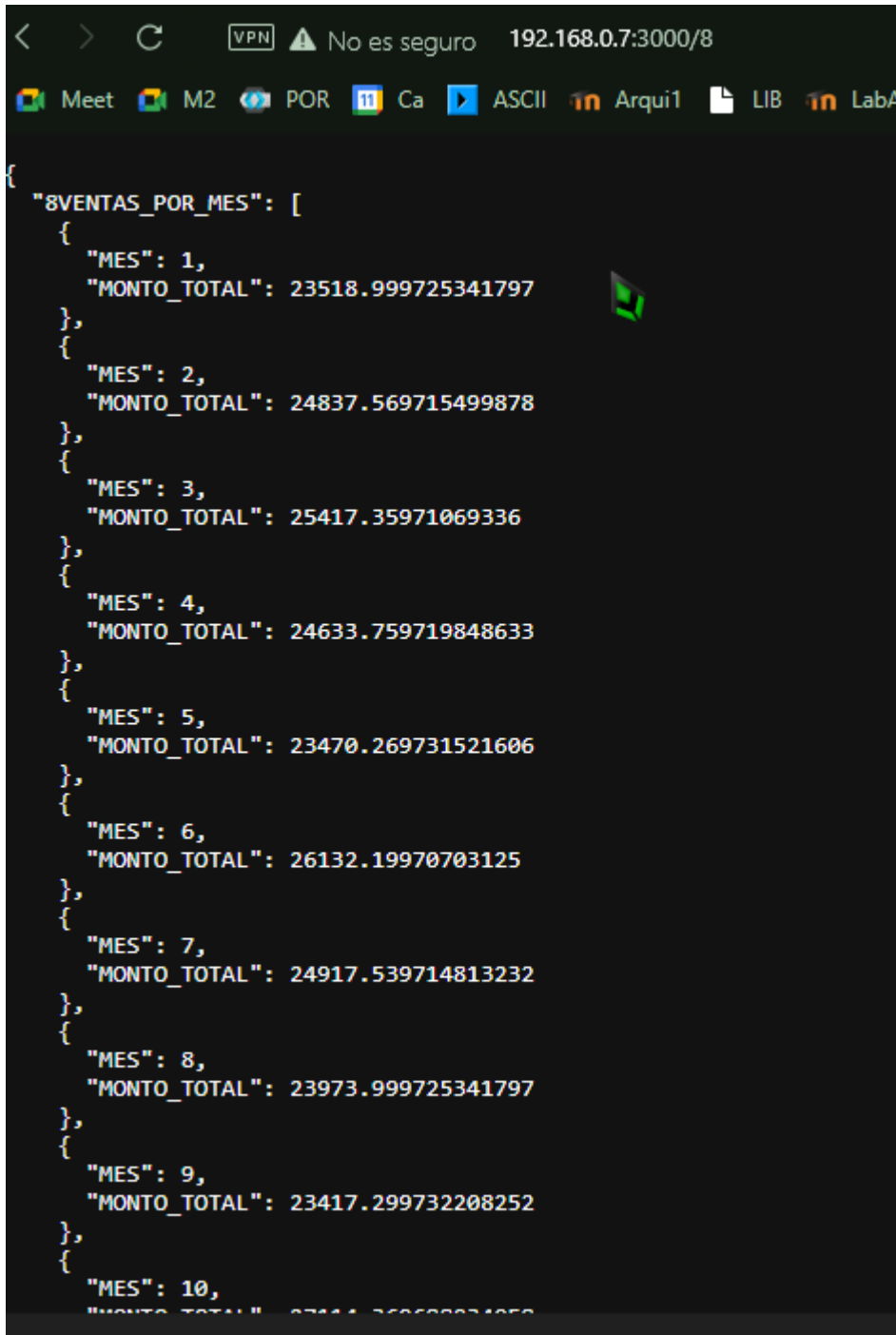
7. Se muestra en este ENDPOINT (Consulta) la categoría más comprada por cada país. Se debe de mostrar el nombre del país, nombre de la

categoría y cantidad de unidades.

```
< > ↻ VPN ⚠ No es seguro 192.168.0.7:3000/7
Meet M2 POR Ca ASCII Arqui1 LIB La

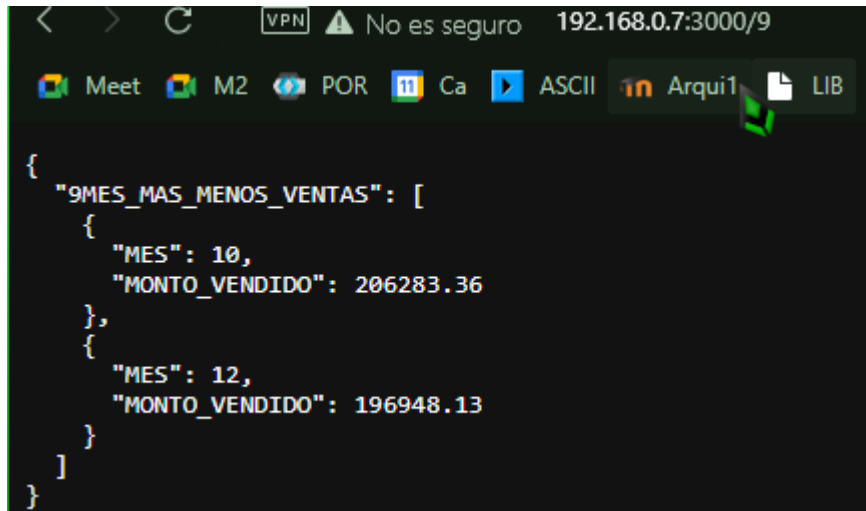
{
  "7CATETORIA_PAIS_MAS_COMPRA": [
    {
      "CANTIDAD": "59768",
      "NOMBRE_CATEGORIA": "Accion",
      "NOMBRE_PAIS": "Estados Unidos"
    },
    {
      "CANTIDAD": "6615",
      "NOMBRE_CATEGORIA": "Viajes",
      "NOMBRE_PAIS": "Australia"
    },
    {
      "CANTIDAD": "6530",
      "NOMBRE_CATEGORIA": "Viajes",
      "NOMBRE_PAIS": "Canada"
    },
    {
      "CANTIDAD": "6253",
      "NOMBRE_CATEGORIA": "Musica",
      "NOMBRE_PAIS": "Chile"
    },
    {
      "CANTIDAD": "6189",
      "NOMBRE_CATEGORIA": "Musica",
      "NOMBRE_PAIS": "Polonia"
    },
    {
      "CANTIDAD": "6156",
      "NOMBRE_CATEGORIA": "Clasicos",
      "NOMBRE_PAIS": "China"
    },
    {
      "CANTIDAD": "6098",
      "NOMBRE_CATEGORIA": "Clasicos",
      "NOMBRE_PAIS": "Francia"
    },
    {
      "CANTIDAD": "5960",
      "NOMBRE_CATEGORIA": "Documentales",
      "NOMBRE_PAIS": "Polonia"
    }
  ]
}
```

8. Se muestra en este ENDPOINT (Consulta) las ventas por mes de Inglaterra. Debe de mostrar el número del mes y el monto.



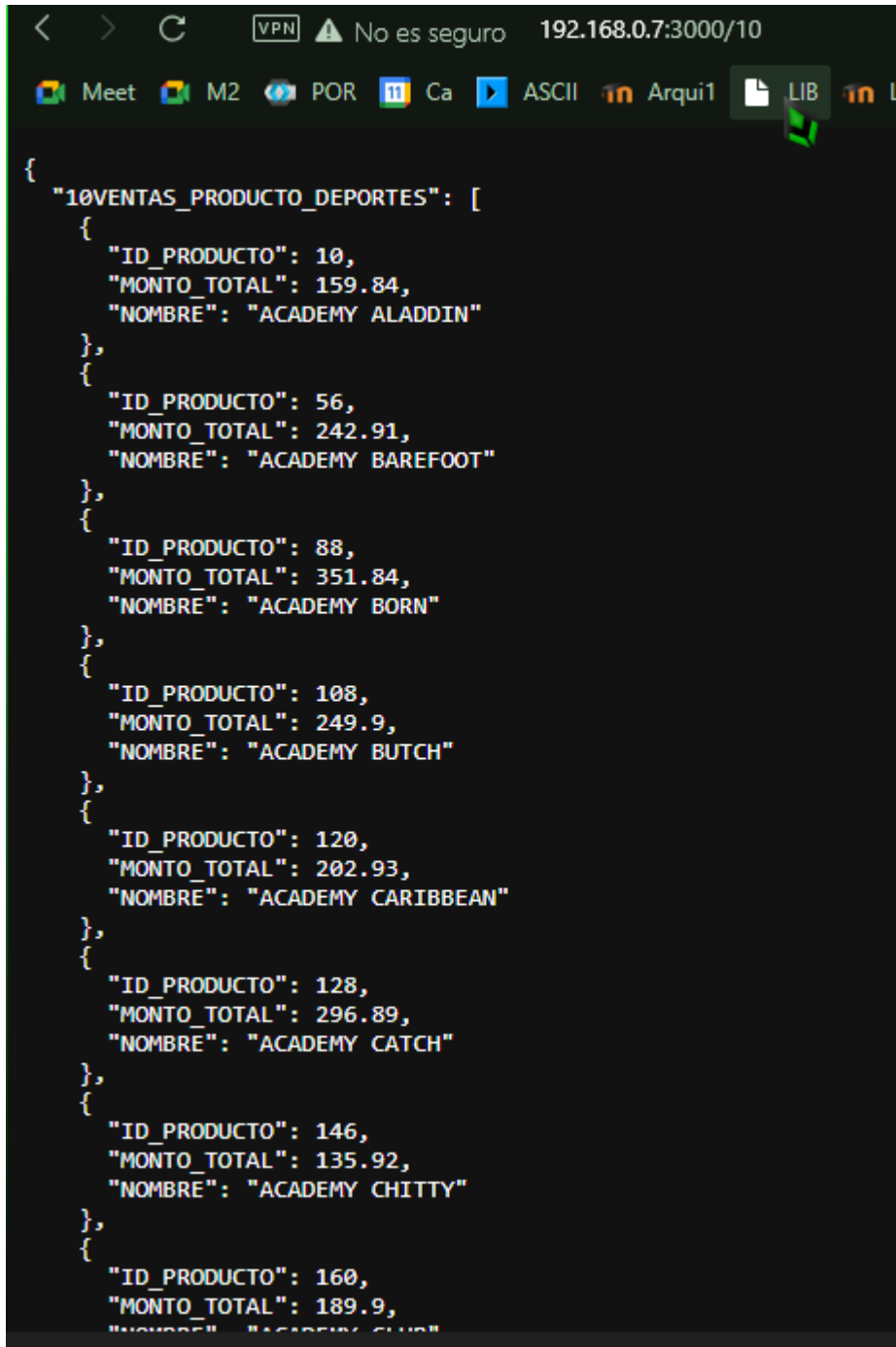
```
{
  "8VENTAS_POR_MES": [
    {
      "MES": 1,
      "MONTO_TOTAL": 23518.999725341797
    },
    {
      "MES": 2,
      "MONTO_TOTAL": 24837.569715499878
    },
    {
      "MES": 3,
      "MONTO_TOTAL": 25417.35971069336
    },
    {
      "MES": 4,
      "MONTO_TOTAL": 24633.759719848633
    },
    {
      "MES": 5,
      "MONTO_TOTAL": 23470.269731521606
    },
    {
      "MES": 6,
      "MONTO_TOTAL": 26132.19970703125
    },
    {
      "MES": 7,
      "MONTO_TOTAL": 24917.539714813232
    },
    {
      "MES": 8,
      "MONTO_TOTAL": 23973.999725341797
    },
    {
      "MES": 9,
      "MONTO_TOTAL": 23417.299732208252
    },
    {
      "MES": 10,
      "MONTO_TOTAL": 23417.299732208252
    }
  ]
}
```

9. Se muestra en este ENDPOINT (Consulta) el mes con más y menos ventas. Se debe de mostrar el número de mes y monto. (Una sola consulta).



```
{
  "9MES_MAS_MENOS_VENTAS": [
    {
      "MES": 10,
      "MONTO_VENDIDO": 206283.36
    },
    {
      "MES": 12,
      "MONTO_VENDIDO": 196948.13
    }
  ]
}
```

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



The screenshot shows a web browser window with a dark theme. The address bar displays '192.168.0.7:3000/10'. The browser's taskbar at the top includes icons for Meet, M2, POR, Ca, ASCII, Arqui1, LIB, and L. The main content area displays a JSON object representing sales data for products in the sports category. The JSON is structured as follows:

```
{
  "10VENTAS_PRODUCTO_DEPORTES": [
    {
      "ID_PRODUCTO": 10,
      "MONTO_TOTAL": 159.84,
      "NOMBRE": "ACADEMY ALADDIN"
    },
    {
      "ID_PRODUCTO": 56,
      "MONTO_TOTAL": 242.91,
      "NOMBRE": "ACADEMY BAREFOOT"
    },
    {
      "ID_PRODUCTO": 88,
      "MONTO_TOTAL": 351.84,
      "NOMBRE": "ACADEMY BORN"
    },
    {
      "ID_PRODUCTO": 108,
      "MONTO_TOTAL": 249.9,
      "NOMBRE": "ACADEMY BUTCH"
    },
    {
      "ID_PRODUCTO": 120,
      "MONTO_TOTAL": 202.93,
      "NOMBRE": "ACADEMY CARIBBEAN"
    },
    {
      "ID_PRODUCTO": 128,
      "MONTO_TOTAL": 296.89,
      "NOMBRE": "ACADEMY CATCH"
    },
    {
      "ID_PRODUCTO": 146,
      "MONTO_TOTAL": 135.92,
      "NOMBRE": "ACADEMY CHITTY"
    },
    {
      "ID_PRODUCTO": 160,
      "MONTO_TOTAL": 189.9,
      "NOMBRE": "ACADEMY CLUB"
    }
  ]
}
```

