

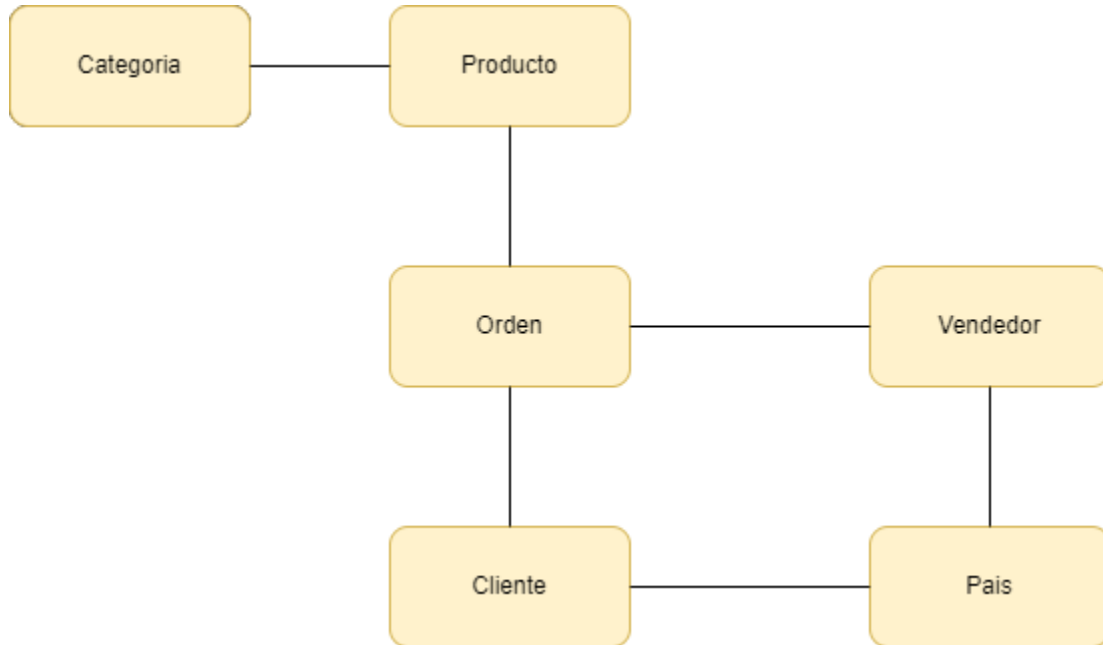
Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Laboratorio de Sistemas de Bases de Datos 1
Aux: Luis Daniel Castellanos Galindo

Manual Técnico – Proyecto 1

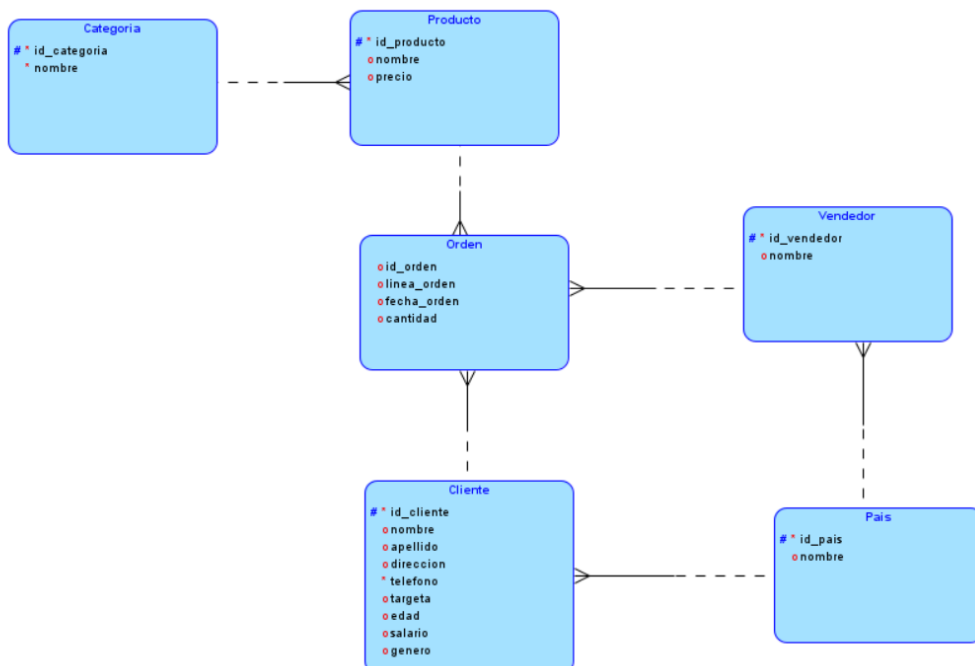
Nombre: Federico David Zet Pajoc
Carné: 201901073

Manual Técnico

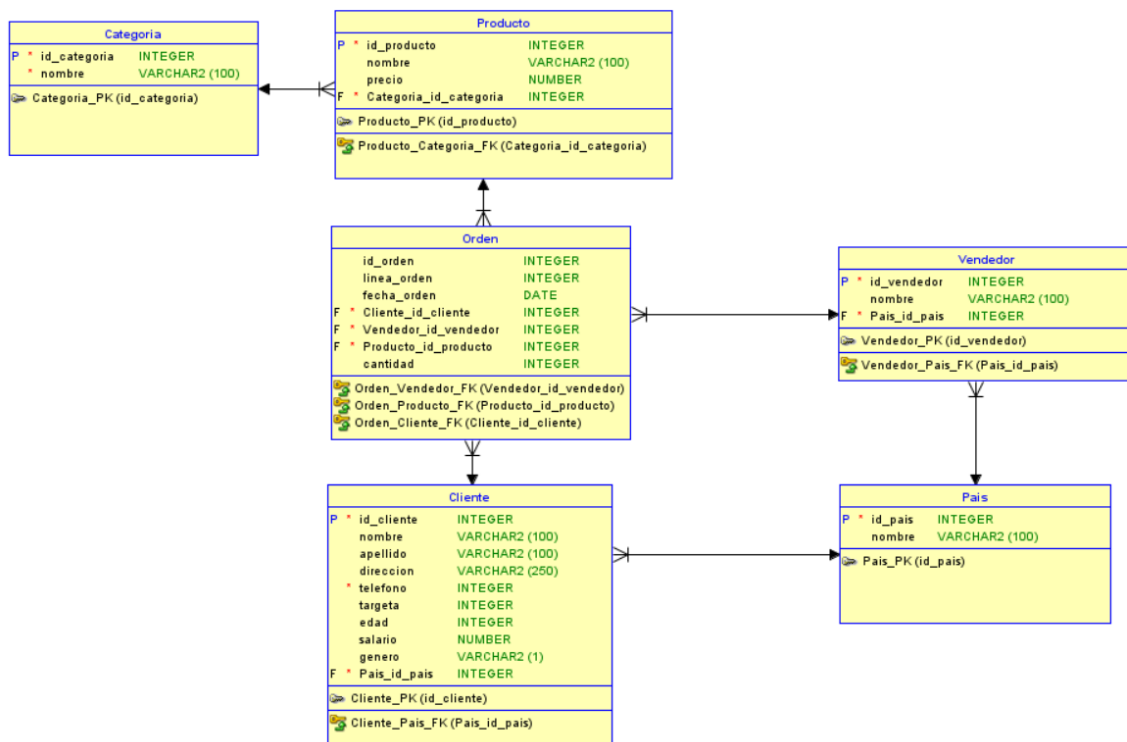
Esquema Conceptual



Esquema Lógico



Esquema Físico



Descripción de las Tablas

1. Tabla Categoría

Dicha tabla se encarga de guardar el nombre de las categorías de los productos, posee una clave primaria con relación de uno a muchos con la tabla productos.

2. Tabla Productos

Guarda los datos de todos los productos como el id, el nombre y el precio, y posee una llave foránea relacionada a la tabla categoría.

3. Tabla Cliente

Se encarga de guardar los datos de todos los clientes que realizan una compra en el sistema, posee una llave foránea de la Tabla País.

4. Tabla Vendedor

Se encarga de guardar los datos de los vendedores, siendo el id, el nombre y su país.

5. Tabla País

Guarda los países de los vendedores y clientes, teniendo un id y el nombre.

6. Tabla Orden

Se encarga de registrar las ordenes de compras o ventas, teniendo un id, la fecha de orden, línea de orden y cantidad. Posee una relación de uno a muchos con la tabla productos, clientes y vendedores.

Descripción de la Api

La Api se desarrolló con el micro-framework Flask por la facilidad de su implementación ya que el lenguaje de programación Python posee librerías útiles para realizar conexiones de bases de datos. La librería utilizada para la conexión de base de datos Oracle es cx_Oracle, también ofrece funciones para la interacción con la base de datos. Realizando la conexión de la siguiente manera. Para realizar el cierre de conexión se utiliza la función close().

```
conexion = cx_Oracle.connect(  
    user = 'userproyecto',  
    password = 'oracle123',  
    dsn = 'localhost/xes',  
    encoding = 'UTF-8'  
)
```

Descripción de los endpoints

Para los endpoints se utilizaron diez funciones con peticiones GET. La librería cx_Oracle posee una función cursor() que se encarga de realizar la conexión, y execute que recibe un parámetro SQL el cual se encarga de ejecutar la consulta y comunicarse con la base de datos. Para obtener los datos en forma de filas se utiliza la función fetchall().

```
@app.route('/consulta1', methods = ['GET'])  
def consulta1():  
    try:  
        cur = conexion.cursor()  
        consulta = '''  
        SELECT c.ID_CLIENTE, C.NOMBRE, c.APELLIDO, p.NOMBRE AS "PAIS", SUM(o.CANTIDAD * pro.PRECIO) AS monto  
        FROM CLIENTE c, ORDEN o, PAIS p, PRODUCTO pro  
        WHERE c.ID_CLIENTE = o.CLIENTE_ID_CLIENTE AND pro.ID_PRODUCTO = o.PRODUCTO_ID_PRODUCTO  
        AND c.PAIS_ID_PAIS = p.ID_PAIS  
        GROUP BY c.ID_CLIENTE, C.NOMBRE, p.NOMBRE, c.APELLIDO ORDER BY monto DESC  
        FETCH FIRST 1 ROWS ONLY  
        '''  
        cur.execute(consulta)  
        rows = cur.fetchall()
```

Para obtener los valores de las columnas se itera con dos ciclos for y se obtienen dichos valores. Los reportes se muestran en formato HTML.

```
for row in rows:
    REPORTE += '<tr>'
    for column in row:
        REPORTE += f'<td>{column}</td>'
    REPORTE += '</tr>'
REPORTE += '''
</tbody>
</table>
'''
return REPORTE
```

Referencias

https://cx-oracle.readthedocs.io/en/latest/user_guide/introduction.html