

Bases de datos 1

Manual Técnico

202111718

Walter Javier Santizo Mazariegos

Tecnologías Utilizadas

NodeJS

Node.js es un entorno de ejecución de JavaScript del lado del servidor que permite a los desarrolladores construir aplicaciones web y aplicaciones de red de manera eficiente. A diferencia de ejecutar JavaScript en el navegador, Node.js permite ejecutar código JavaScript en el servidor, lo que facilita la creación de aplicaciones de extremo a extremo con un solo lenguaje de programación.



Mysql

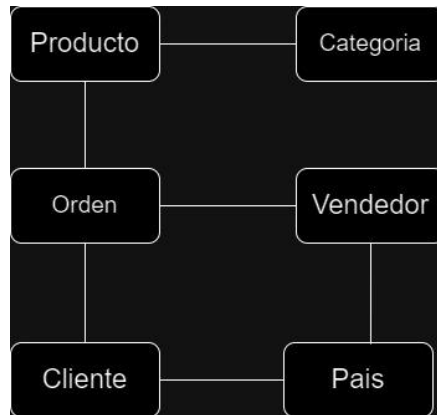
MySQL es un sistema de gestión de bases de datos relacional (RDBMS, por sus siglas en inglés) de código abierto. Fue desarrollado por MySQL AB, que luego fue adquirida por Oracle. MySQL es uno de los sistemas de gestión de bases de datos más populares en el mundo y es ampliamente utilizado para almacenar y gestionar datos en aplicaciones web y otros tipos de software.



Modelos de Negocio

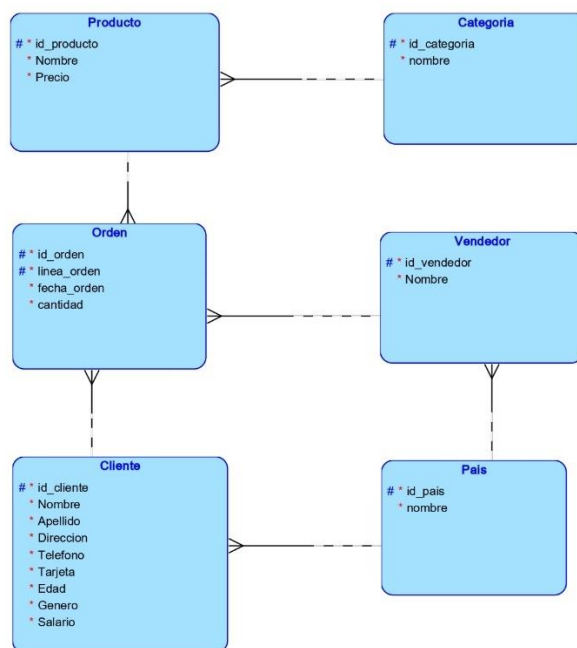
Una de las partes mas importantes del diseño de bases de datos es el análisis del modelo de negocio (reglas de negocio, restricciones, inferencias, etc..) una vez analizados todos los requerimientos necesarios se procedió a realizar el diseño EN 3 partes el modelo conceptual, lógico , físico

Modelo Conceptual



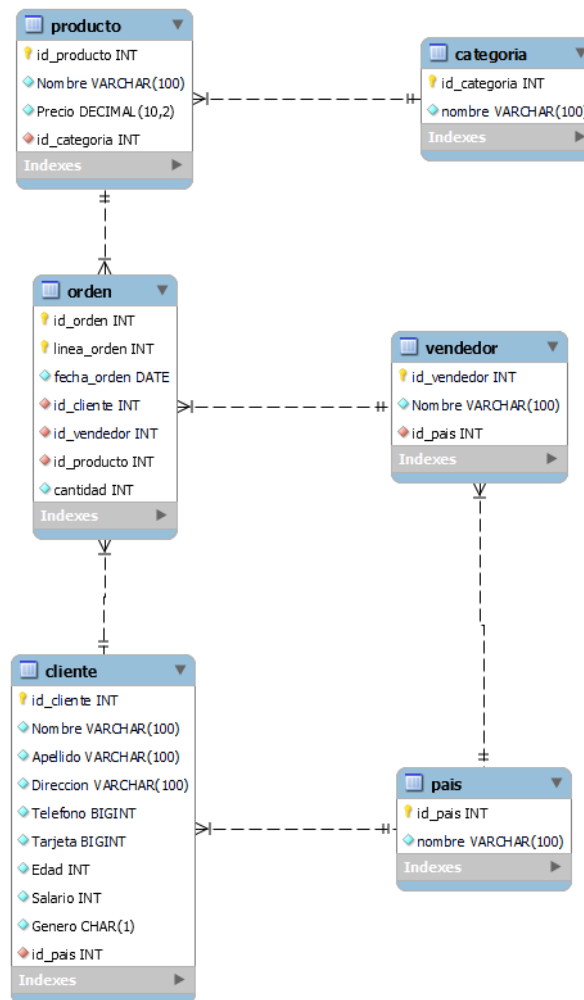
Mediante el análisis de el enunciado se llego a la conclusión que el negocio tiene 6 entidades las cuales tienen relación entre ellas se procede a marcar con una línea la relación entre entidades

Modelo Lógico



Este modelo se encargo de profundizar en los requerimientos y restricciones del negocio por ejemplo aquí se definieron los campos obligatorios por medio de un asterisco las llaves primarias por medio del carácter “#” además de las relaciones que se especifica el tipo de relación debe o puede, de uno a muchos

Modelo Relacional




Este modelo es directamente un producto del diseño del modelo físico ya que por medio del ide de workbench se realizó una ingeniería inversa a partir del script que se encarga de crear el modelo por ello es posible visualizar el tipo de dato de cada campo y si es una llave foránea color morado o primaria color amarillo o icono de una llave

Modelo Físico

Este modelo consta del script que se encarga de crear las tablas donde se almacenarán todas las entidades con sus respectivas relaciones debido a que se uso Mysql la sintaxis es la siguiente

Create, Drop y Use




```
Drop database Proyecto1;  
Create Database Proyecto1;  
Use Proyecto1;
```


Sentencias encargadas de crear y borrar la base de datos tomar en cuenta que el drop se debe utilizar únicamente cuando se este seguro de borrar todo el modelo mientras que use sirve para indicar al script que se esta trabajando sobre esa base de datos es decir todos los scripts que se ejecuten después de este estarán dentro de la base Proyecto 1

Create Table

Esta sentencia se encarga de crear las tablas dentro de nuestra base de datos dentro de los parámetros que recibe son el nombre de las columnas seguido del tipo de dato además de que aquí se indica si un campo es llave primaria o foránea o si el campo es obligatorio o no



```
Create Table Categoria(  
  id_categoria integer primary key,  
  nombre varchar(100) not null  
);
```



```
Create Table Pais(  
  id_pais integer primary key,  
  nombre varchar(100) not null  
);
```

```
Create Table Producto(  
  id_producto integer primary key,  
  Nombre varchar(100) not null,  
  Precio decimal(10,2) not null,  
  id_categoria integer not null,  
  FOREIGN KEY (id_categoria) REFERENCES Categoria(id_categoria)  
);
```

```
Create Table Vendedor(  
  id_vendedor integer primary key,  
  Nombre varchar(100) not null,  
  id_pais integer not null,  
  FOREIGN KEY (id_pais) REFERENCES Pais(id_pais)  
);
```

Drop y Truncate

Estas sentencias se encargan de eliminar sobre la tabla en cuestión la diferencia reside en que el truncate simplemente elimina los registros de la tabla pero el modelo se conserva mientras que el drop elimina el modelo entero

```
drop table Orden;  
truncate table orden;
```

Deploy del Servidor con Node js

1. Clonar el repositorio de github

```
git clone https://github.com/walterjav19/ProyectosBD1_202111718.git
```

2. Navegar hasta la carpeta raíz del servidor

```
cd ./server
```

3. Instalar todos los paquetes de Node

```
npm i
```

4. Asegurarse que todos los scripts estén en archivo package.json

```
"scripts": {  
  "test": "echo \\\"Error: no test specified\\\" && exit 1",  
  "start": "node index.js",  
  "dev": "nodemon index.js"  
},
```

5. Utilizar el comando npm run

```
npm run dev
```

6. Asegurarse que tanto la conexión a la base de datos como el servidor estén levantados

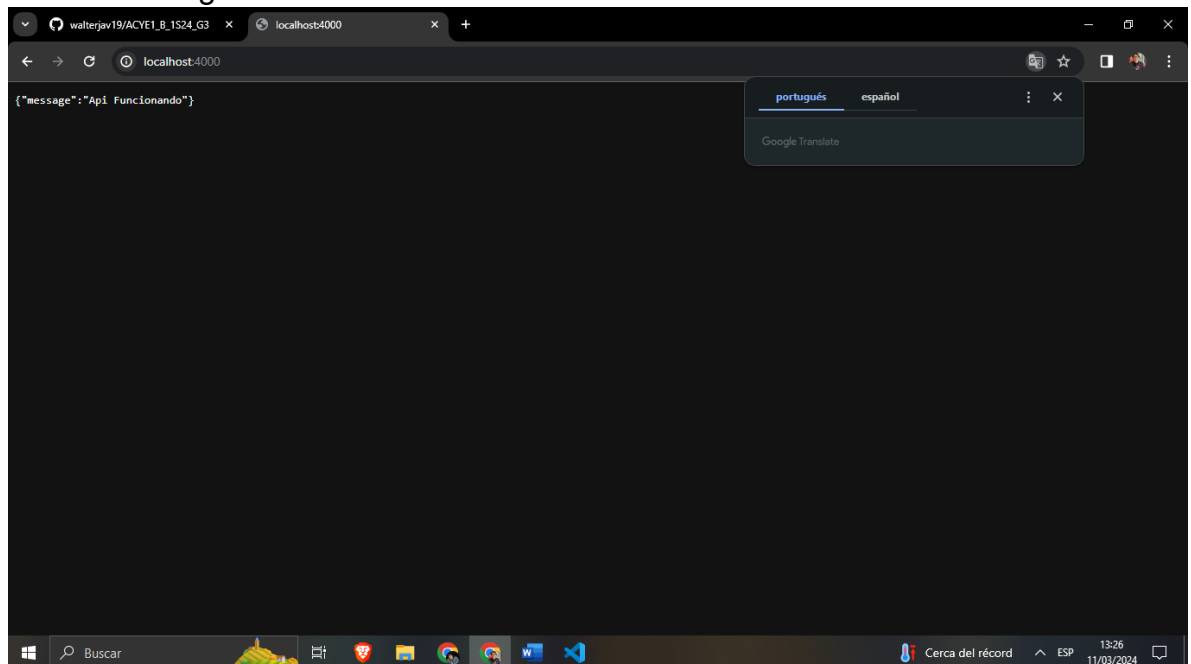
```
> npm run dev

> server@1.0.0 dev
> nodemon index.js

[nodemon] 3.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Server running on localhost:4000
Conexion exitosa a la base de datos
```

Consultas

Para realizar las consultas utilizaremos el método GET y en el ejemplo se utilizara el navegador de Chrome una vez levantado el servidor el mensaje debe ser el siguiente



Ahora bien para una consulta en especifico se debe de realizar de la siguiente forma ruta-api/consulta{n}

```
{\"message\": \"Consulta 1\", \"result\": [{\"id_cliente\": 19887, \"Nombre\": \"Evelyn\", \"apellido\": \"Olson\", \"Pais\": \"Inglaterra\", \"monto_total\": 1894.13}]}
```


La respuesta consta de un message indicando el numero de consulta realizada y el resultado de la consulta