

**Desarrollo y Programación de una Base de Datos de Comercio Electrónico para la  
Empresa TechMarket**

Andrés Camilo Romero Camargo

Juan Camilo Serrano Correa

Jose Oswaldo Sierra Tipazoca

21 de Abril de 2023

Universidad Pedagógica y Tecnológica de Colombia

Programación en Bases de Datos

PROYECTO FINAL

## Contenido

- Construcción modelo entidad
- Web Scraping
- Limpieza de datos
- Consultas a la base de datos
  - Consultas multi-tablas
  - Subconsultas
  - Consultas con expresiones regulares
  - Consultas de análisis de texto
- Reportes gráficos

## Resumen

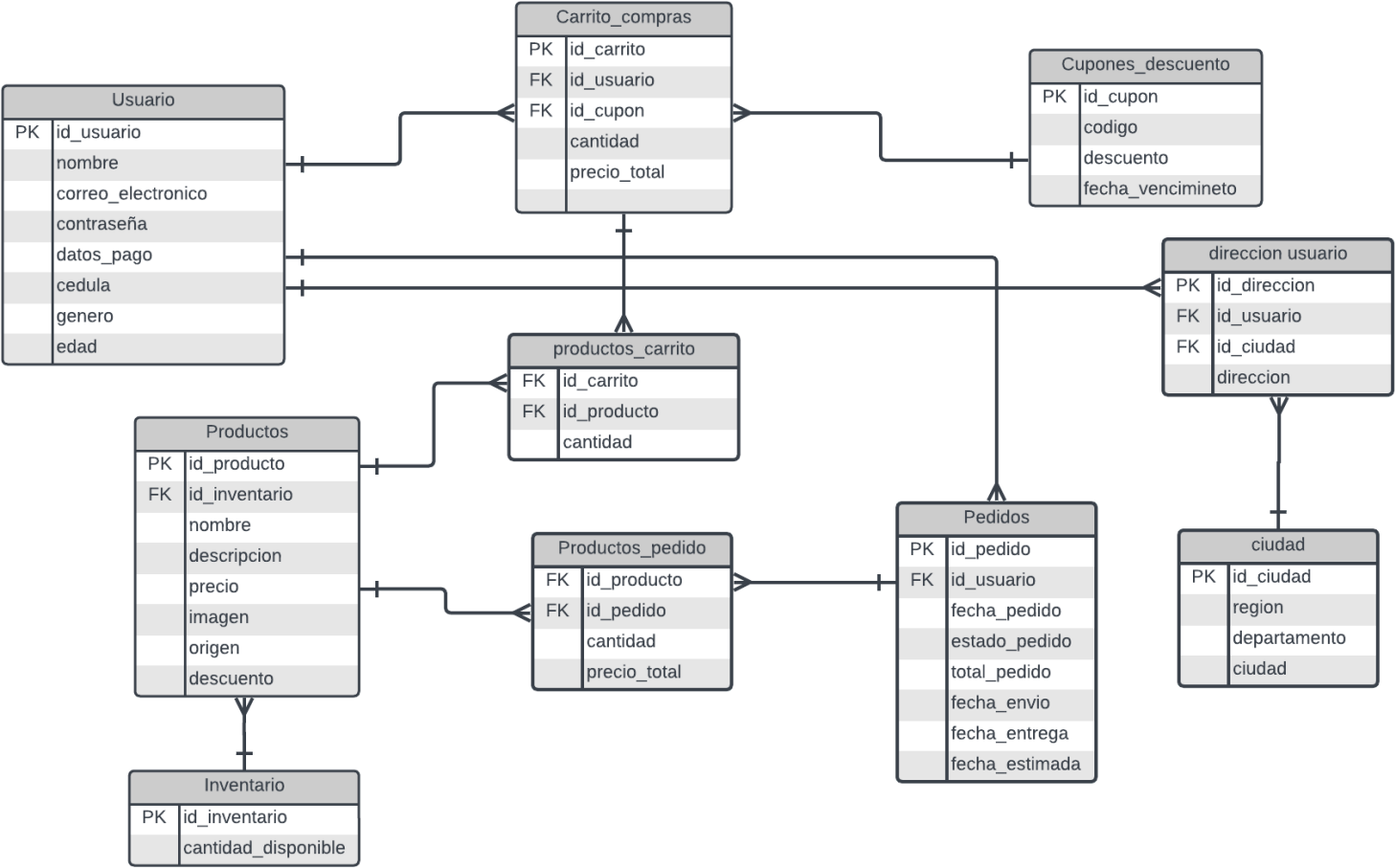
El presente trabajo aborda el desarrollo y programación de una base de datos de comercio electrónico para la empresa techmarket que vende productos tecnológicos a nivel nacional. La empresa actúa como intermediaria para la venta de algunos productos que se obtienen de MercadoLibre y otros propios. Los requisitos del sistema incluyen el registro y autenticación de usuarios, un catálogo de productos con información detallada, un carrito de compras con capacidad para aplicar cupones de descuento, la gestión de pedidos y el control de inventario en tiempo real.

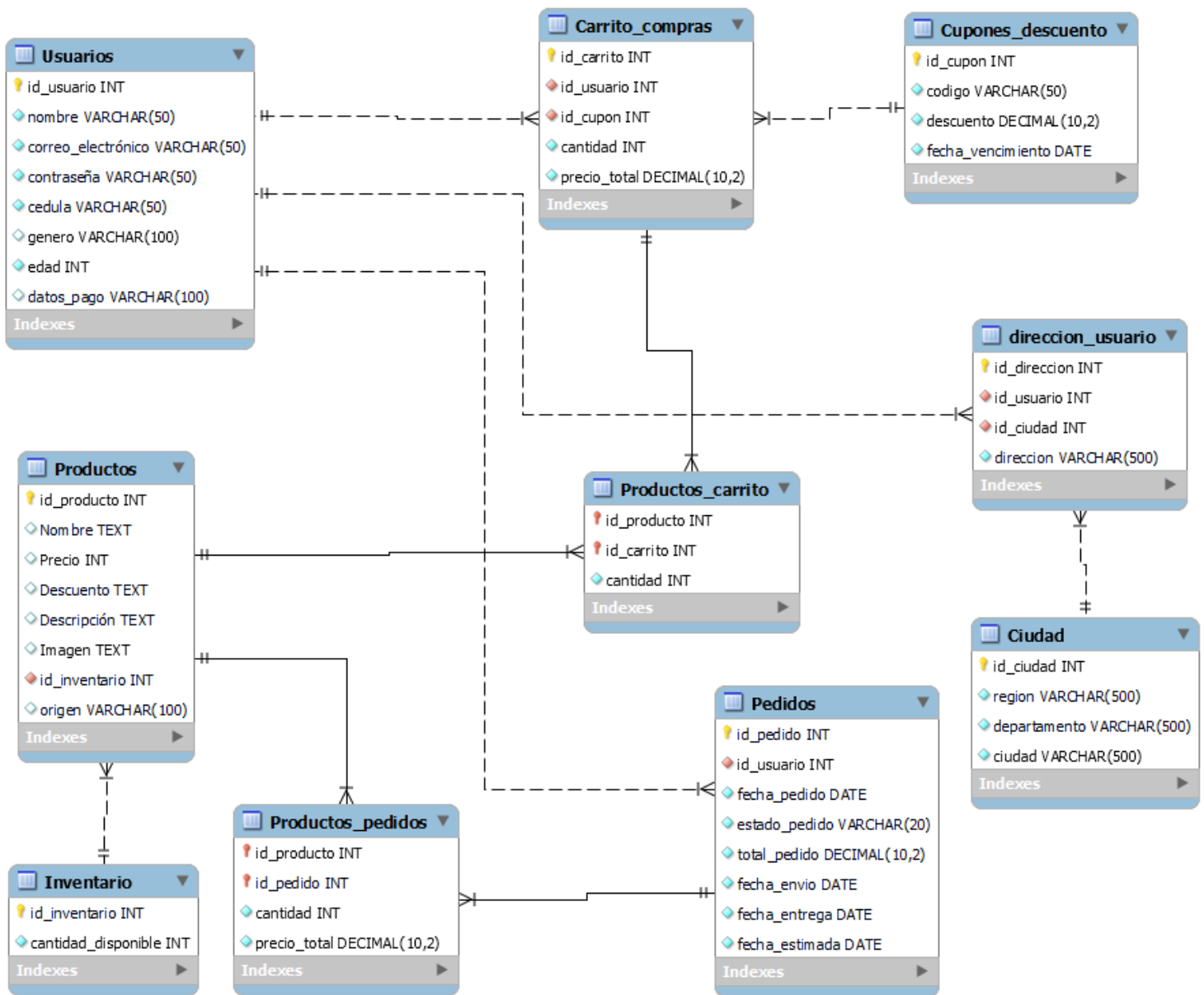
El trabajo se divide en varias etapas. En primer lugar, se propone un modelo entidad-relación que integra los requisitos del proyecto, el cual incluye un script de creación de la estructura de la base de datos y la inserción de registros de ejemplo. A continuación, se lleva a cabo la extracción de datos mediante técnicas de web scraping para obtener información desde MercadoLibre, y se realiza la limpieza y almacenamiento de estos datos en la base de datos.

Posteriormente, se aborda el tema de las consultas a la base de datos, incluyendo consultas multi-tablas, subconsultas, consultas con expresiones regulares y consultas de análisis de texto. Además, se propone la visualización de reportes gráficos utilizando el lenguaje de programación Python en conexión con la base de datos seleccionada.

En conclusión, el trabajo abarca la construcción del modelo entidad-relación, la extracción de datos mediante web scraping, las consultas a la base de datos, incluyendo consultas avanzadas y el análisis de texto, así como la generación de reportes gráficos para visualizar los resultados obtenidos. El informe escrito sigue las normas APA para una presentación adecuada del proceso desarrollado.

**Construcción modelo entidad.**





Entidades:

- Usuarios: Representa a los usuarios del sistema.
- Carrito\_compras: Representa los carritos de compras de los usuarios.
- Cupones\_descuento: Representa los cupones de descuento que los usuarios pueden aplicar a su carrito.

- Pedidos: Representa los pedidos realizados por los usuarios.
- Direccion\_usuario: Representa las direcciones de los usuarios.
- Ciudad: Representa las ciudades donde se encuentran las direcciones de los usuarios.
- Productos: Representa los productos disponibles en el sistema.
- Productos\_carrito: Representa los productos agregados a los carritos de compras.
- Productos\_pedido: Representa los productos y sus cantidades por pedidos.
- Inventario: Representa el inventario de los productos que están en el catálogo.

#### Relaciones:

- Relación uno a muchos entre Usuarios y Carrito\_compras: Un usuario puede tener varios carritos de compras, pero cada carrito de compras pertenece a un único usuario.
- Relación uno a muchos entre Usuarios y Pedidos: Un usuario puede tener varios pedidos, pero cada pedido pertenece a un único usuario.
- Relación uno a muchos entre Usuarios y Direccion\_usuario: Un usuario puede tener varias direcciones, pero cada dirección pertenece a un único usuario.
- Relación uno a muchos entre Ciudad y Direccion\_usuario: Una ciudad puede tener varias direcciones de usuarios, pero cada dirección pertenece a una única ciudad.
- Relación uno a muchos entre Productos y Productos\_carrito: Un producto puede estar en varios carritos de compras, pero cada producto en el carrito pertenece a un único producto.
- Relación uno a muchos entre Carrito\_compras y Productos\_carrito: Un carrito de compras puede tener varios productos, pero cada producto en el carrito pertenece a un único carrito de compras.

## Web Scraping

Para extraer el catalogo de productos de Mercado Libre se utilizo web scraping utilizando Selenium. Selenium es una herramienta que permite automatizar la interacción con un navegador web para extraer información de páginas web.

En este caso, el código está utilizando Selenium para extraer información sobre drones en el sitio web de MercadoLibre Colombia. El código comienza importando las librerías necesarias y definiendo la URL y el número máximo de páginas a procesar.

Luego, se crea una instancia del navegador Chrome y se accede a la URL especificada. Se crea un archivo CSV y se escribe la fila de encabezado con los campos 'Nombre', 'Precio', 'Descuento', 'Descripción' e 'Imagen'.

El código utiliza un bucle while para procesar múltiples páginas. Dentro del bucle, se espera a que los elementos de los productos sean visibles y luego se obtienen los elementos de los productos utilizando el selector CSS '.ui-search-layout\_\_item'.

Para cada producto, se extrae el nombre, el precio y el descuento (si lo hay). También se obtiene la descripción del producto abriendo una nueva pestaña y accediendo a la página del producto.

Finalmente, se escriben los datos del producto en el archivo CSV y se cierra la pestaña abierta para obtener la descripción del producto.

## Limpieza de datos

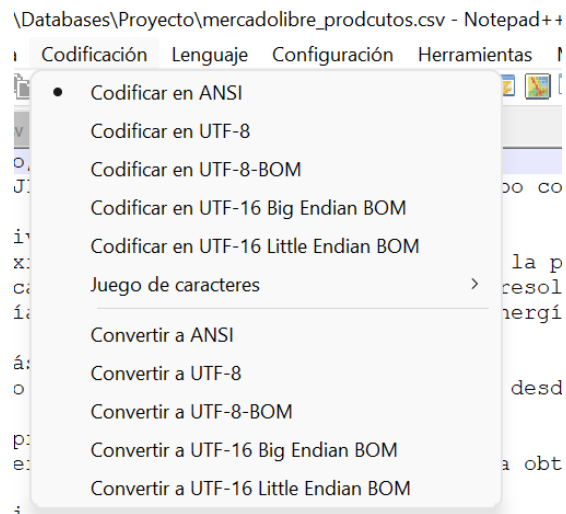
Para la creación del archivo.csv se comienza especificando la codificación `encoding='utf-8'` para asegurarse de que los caracteres especiales se escriban correctamente en el archivo CSV.

Se utiliza el parámetro `newline=""` para evitar la creación de nuevas líneas adicionales.

Cómo delimitador se utiliza `(|)`, no se recurre al delimitador `(,)` o `(;)` que son los más usados debido a que estos caracteres aparecen con frecuencia en los datos que se están scrapeando produciendo que se interprete como más campos.

```
# Escribir los datos en el archivo CSV
with open('productos6.csv', mode='a', newline='', encoding='utf-8') as file:
    writer = csv.writer(file, delimiter='|')#
    writer.writerow([name, price, discount, description_text, image])
```

El conjunto de caracteres del archivo CSV no coincide con el conjunto de caracteres esperado por MySQL Workbench. Para solucionar este problema se cambia el conjunto de caracteres a 'latin1' de la siguiente manera.





Una vez listo el archivo.csv con los datos obtenidos del scraping, lo primero que hacemos es agregar un id a cada uno de los registros, como normalmente el id es la primera columna de una tabla; tambien nos aseguramos de esto como se muestra acontinuacion.

```
#Agrega id, y que este sea la primera columna
ALTER TABLE productos_mercadolibre
ADD id INT AUTO_INCREMENT PRIMARY KEY FIRST;
```

Mysql Workbench no recibe los numeros separados por coma, y usamos la funcion find and replace de notepad++, para eliminar todos los puntos del archivo, no solo estaríamos eliminando los puntos de los precios de los productos, si no tambien del link de las imagenes, por lo que esta limpieza de datos se realizo en Mysql con la siguiente query

```
UPDATE productos_mercadolibre
SET Precio = REPLACE(precio, ',', '')
where id < 500;
```

Finalmente en el scrapeo quedaron algunos registros en donde no se obtuvieron el precio del producto almacenando (menos de 10) en su lugar un string, para que esto no cause problemas, en la columna precio igualamos a cero todas las variables que contengan letras

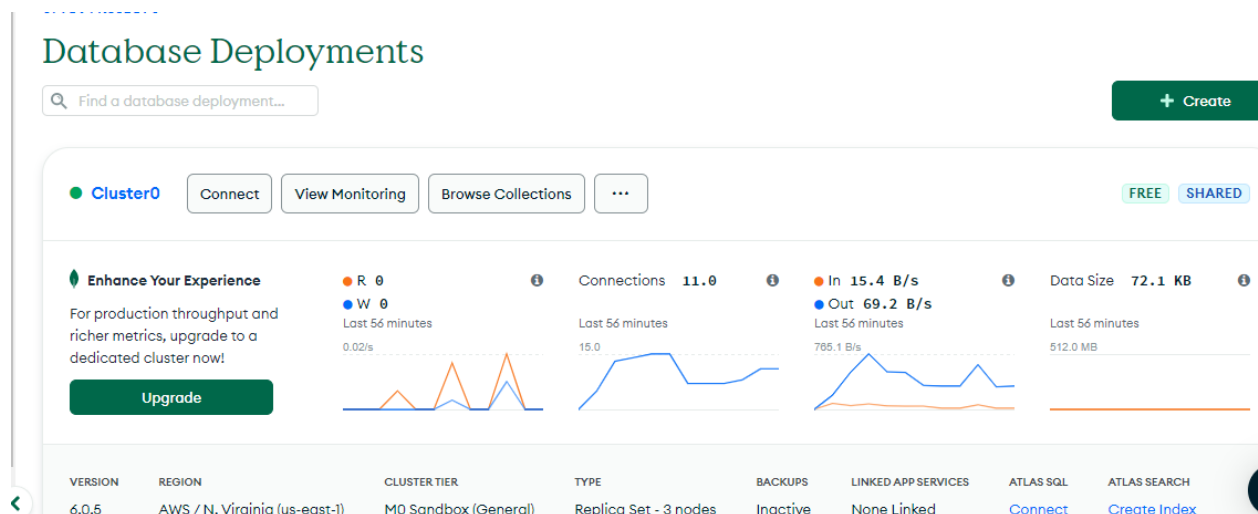
```
#la columna precio tiene datos sin precio y en lugar tienen un string, si es string entonces = 0
UPDATE productos_mercadolibre
SET precio = IF(precio REGEXP '^[0-9]+(\\.[0-9]+)?$', precio, 0)
where id < 500;
```

## Base de datos no relacional

Se decide elegir el trabajo con bases de datos no relacionales usando MongoDB ya que esta es ampliamente usada para el desarrollo de todo tipo de aplicaciones, este trabajo se realizó solo para la tabla de usuarios, es decir para el proceso de autenticación y registro.

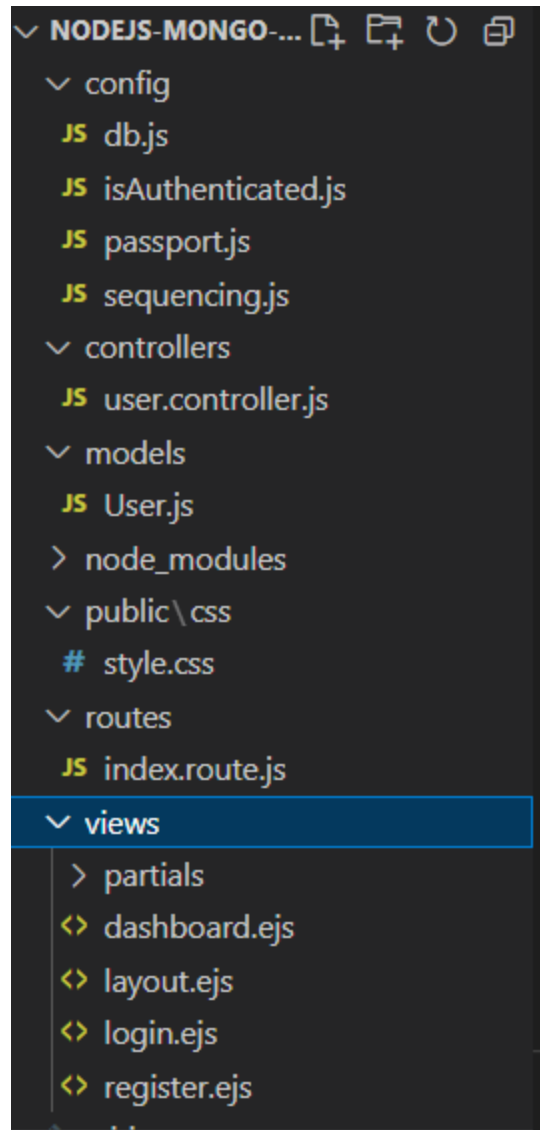
### usando mongoDB:

inicialmente se decide usar la plataforma online llamada mongoAtlas para almacenar la base de datos que contendrá nuestra tabla usuarios, para esto es necesario registrarse e iniciar el proceso de creación de un espacio gratuito en la plataforma:



En la anterior imagen se que ya está creado un cluster que almacena la base de datos, allí es donde irá toda la información de los usuarios.

ahora es necesario escoger la manera de poblar esta bases de datos, teniendo en cuenta que es necesario encriptar la contraseña para su posterior uso, por esta razón se decide trabajar con el entorno de NODE JS el cual a través de sus múltiples módulos y de código java nos permitirá realizar todo eso, además podremos establecer un servidor local que nos permita simular acción de registro y autenticación:



en la anterior imagen tenemos la estructura de nuestro proyecto en node que nos permitirá trabajar con mongoDB mediante el uso del módulo MONGOOSE.

## Register here

Name:

Email:

Password:

Confirm Password:

Edad:

cedula:

pago:

genero:

Esta será la página de registro de usuarios, es decir son los campos que se requerirá al usuario para registrarse.

para todo esto hay que tener en cuenta que mongodb recibe documentos dentro de su base de datos y hay que enviarle el requerimiento con los datos específicos y el los agrega automáticamente el cuerpo tiene la siguiente estructura:

```
JS User.js X
models > JS User.js > [X] UserSchema
4  const autoIncrement = require('mongoose-seque
5
6  const UserSchema = new mongoose.Schema({
7    _id: Number,
8    name: {
9      type: String,
10     required: true
11   },
12   email: {
13     type: String,
14     required: true
15   },
16   password: {
17     type: String,
18     required: true
19   },
20   edad: {
21     type: Number,
22     required: true
23   },
24   cedula: {
25     type: Number
```

ahora para la encriptación de la contraseña se usa el módulo bcrypt de mongoose

```

        res.render('register', {
            name,
            email,
            errors: {...errors}
        })
    } else {
        const newUser = new User({...req.body});
        bcryptjs.genSalt(10, (err, salt) => {
            bcryptjs.hash(newUser.password, salt, (err, hash) => {
                newUser.password = hash;
                newUser.save()
                    .then(user => {return res.redirect("/login")})
                    .catch(err => res.send(err));
            })
        })
    }
})
.catch(error => {
    console.log("error", error);
});

```

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

Filter



Type a query: { field: 'value' }

ADD DATA

EXPORT COLLECTION



```

_id: 1
name: "jose oswaldo sierra"
email: "jost9804@gmail.com"
password: "$2a$10$qU.I2uE0cV/CLX0RoYgovuooU8B31ijY02b6jVfoI92EUInKBY8gq"
edad: 24
cedula: 1054683066
pago: "efecty"
genero: "MAS"
__v: 0

```

El resultado del password convertido en un hash se puede ver en la imagen anterior con todos los demás datos dentro del documento. se creó un id único y auto-incremental por el hecho de futuras consultas relacionadas a la base de datos de mysql.

## Consultas a la base de datos

- Consultas multi-tablas

1. Obtener los productos del carrito de compras con ID = 1 de carrito:
2. Obtener los usuarios tienen más de una ciudad de domicilio
3. Obtener todos los productos del usuario llamado Pedro Rodriguez en su carrito de compras:
4. Obtener la lista de productos disponibles en el inventario, incluyendo su información de categoría:
5. Obtener los clientes y sus pedidos pendientes:
6. Obtener lista de los clientes que hicieron un pedido para Medellín y ya fueron enviados
7. Consulta que me muestre cuántos usuarios hay por ciudad.
8. Consultar los 10 productos con más inventarios.
9. Cantidad de productos pedidos por cada origen.
10. Consulta cuantos productos se encuentran en el carrito\_compras.
11. Consulta los usuarios que tienen los Cupones\_descuento mas grandes.
12. Consulta los usuarios que tienen los Cupones\_descuento mas proximos a vencerse.
13. Consultar los 10 productos con menos inventarios.
14. Consulta los usuarios que tienen los Cupones\_descuento mas pequeños.
15. Saber que carritos de compras están usando cupones que aún no han vencido

- Subconsultas

1. Obtener los nombres y correos electrónicos de los usuarios que han realizado pedidos.



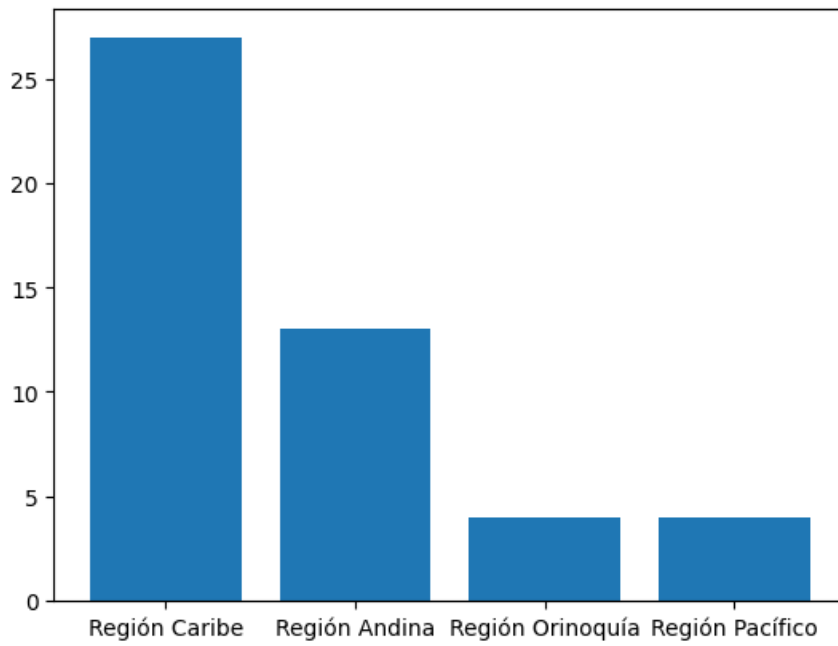
2. Obtener la cantidad de productos en el carrito de compras de un usuario específico.
3. Obtener los nombres de los usuarios que han realizado un pedido y la cantidad de pedidos que han hecho.
4. Obtener los nombres y correos electrónicos de los usuarios que han realizado un pedido y que tienen una dirección en una ciudad específica.
5. Obtener la cantidad total de productos en el inventario que han sido pedidos.
6. Obtener los nombres de los usuarios que han usado un cupón de descuento específico en su carrito de compras.
7. Obtener los productos que han sido pedidos al menos una vez.
8. Obtener los nombres y correos electrónicos de los usuarios que han realizado un pedido y que tienen más de 30 años.
9. Obtener el número total de pedidos realizados por cada usuario.
10. Obtener los nombres de los usuarios que tienen un carrito de compras con al menos 3 productos.
11. Obtener el número total de productos en el inventario que no han sido pedidos.
12. tienen una dirección en una ciudad que pertenece a una región específica.
13. Obtener los nombres y correos electrónicos de los usuarios que han utilizado un cupón de descuento que está vencido.
14. Obtener los nombres de los usuarios que han realizado un pedido y que han usado un cupón de descuento que tiene un descuento superior al 30.000.
15. Obtener los nombres de las ciudades que tienen más de 2 usuarios registrados.

- Consultas con expresiones regulares

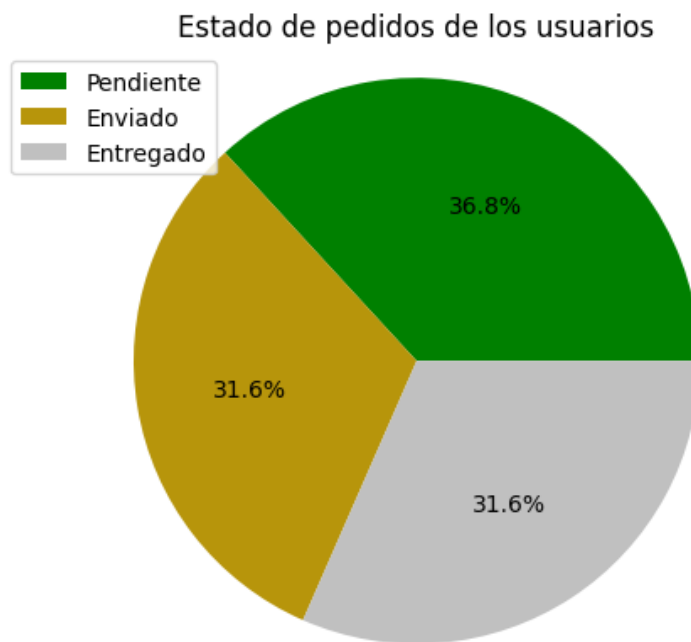
1. Seleccionar todos los productos que en su descripción contienen la palabra negro
  2. Seleccionar todos los usuarios cuyo correo electrónico termine en ".com".
  3. Seleccionar todos los productos cuyo nombre empiece con la letra "C" y termine con la letra "o"
  4. Seleccionar todos los cupones de descuento que tengan en su código la letra "A" o "B"
  5. Seleccionar todos los usuarios cuyo nombre empiece con la letra "M" o "A" y cuyo correo electrónico contenga la palabra "gmail"
- Consultas de análisis de texto
    1. Contar la cantidad de productos que contienen la palabra "nuevo" en su descripción.
    2. Listar los productos cuya descripción contiene la palabra "oferta".
    3. Encontrar los productos que tienen una descripción similar a "WiFi".
    4. utilizando la función SOUNDEX' Obtener los productos que contienen la frase "envío gratis" en su descripción.

## Reportes gráficos

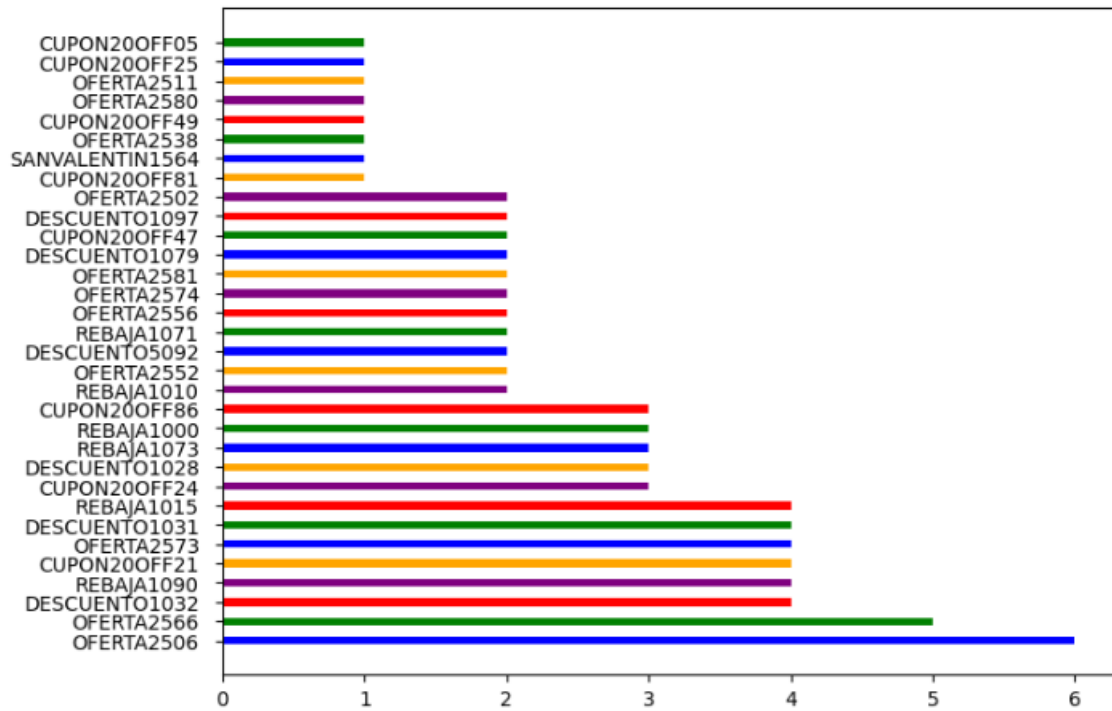
1. Cantidad de ciudades registradas como domicilio de los usuarios por cada región del país



2. Estado del envío de todos los pedidos que se han realizado, con sus respectivos porcentajes.



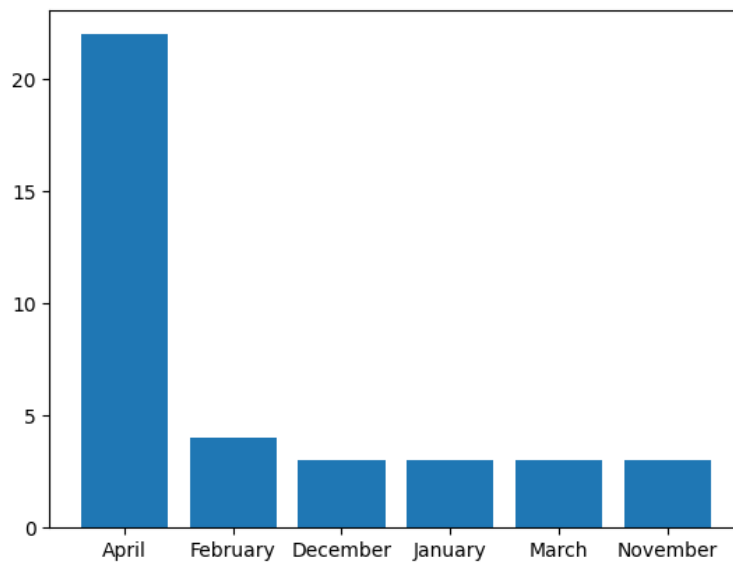
3. Se quiere saber cual es el cupón de descuento más usado por los usuarios en su carrito de compras, con su respectivo porcentaje.



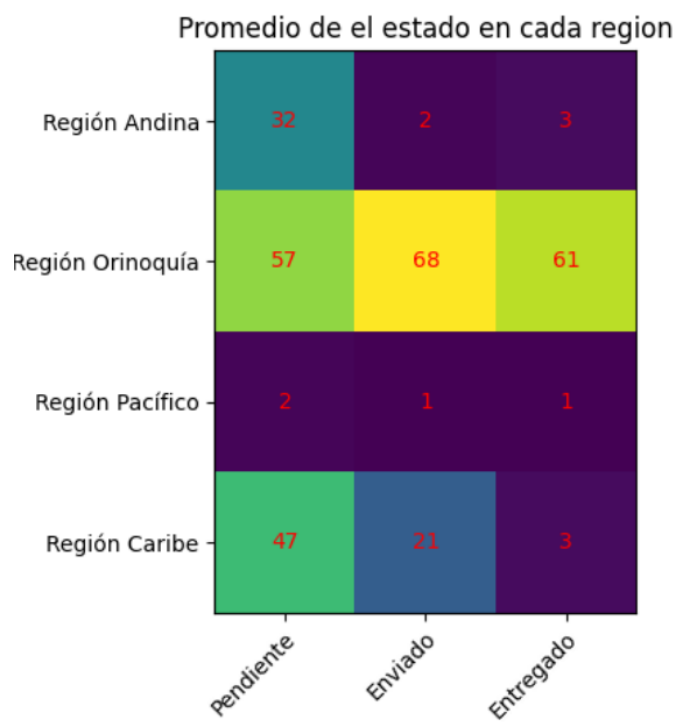
4. Se desea conocer qué día de la semana se han generado más ventas.



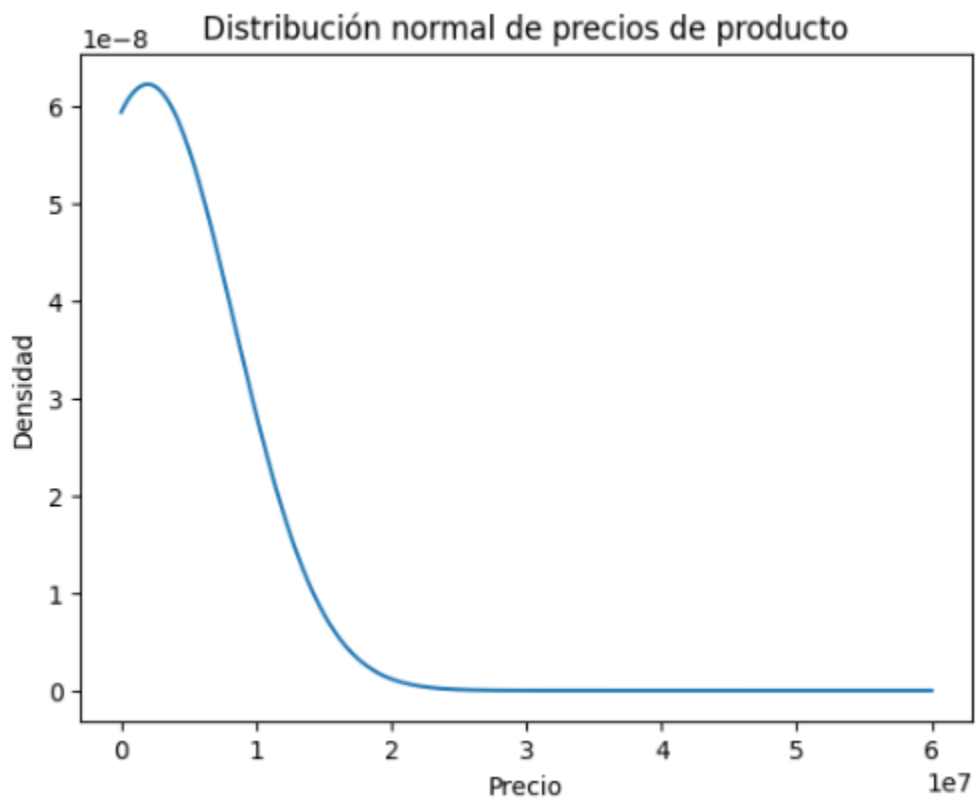
5. Se desea conocer en qué mes del año se han generado más ventas.

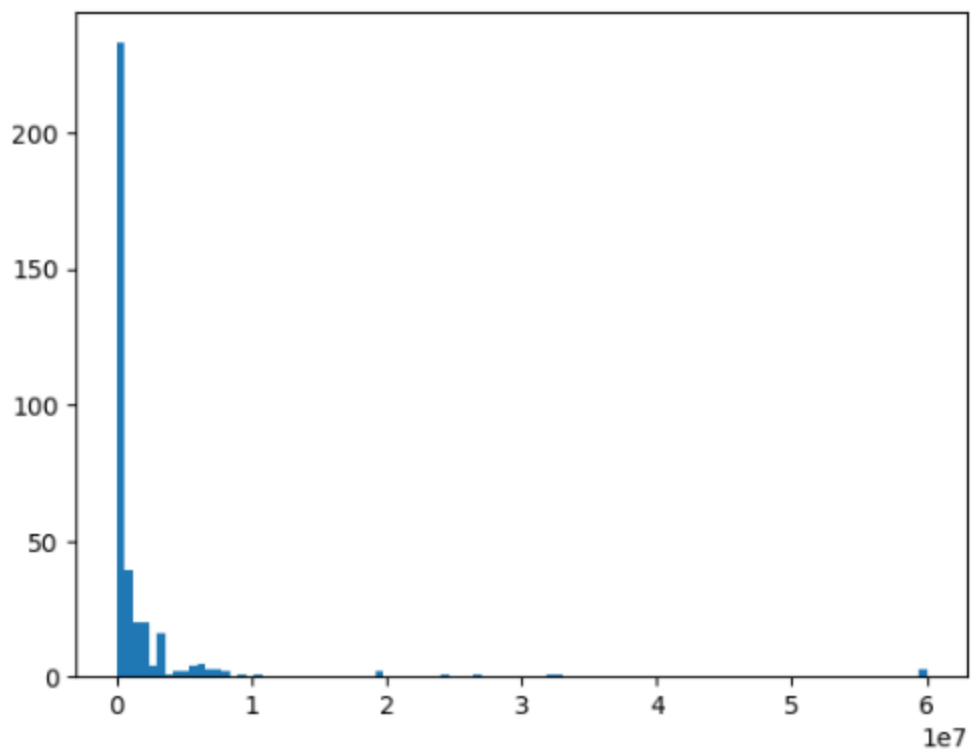


6. Con respecto a el estado de los pedidos y su ubicación por región se tiene un total promedio de productos

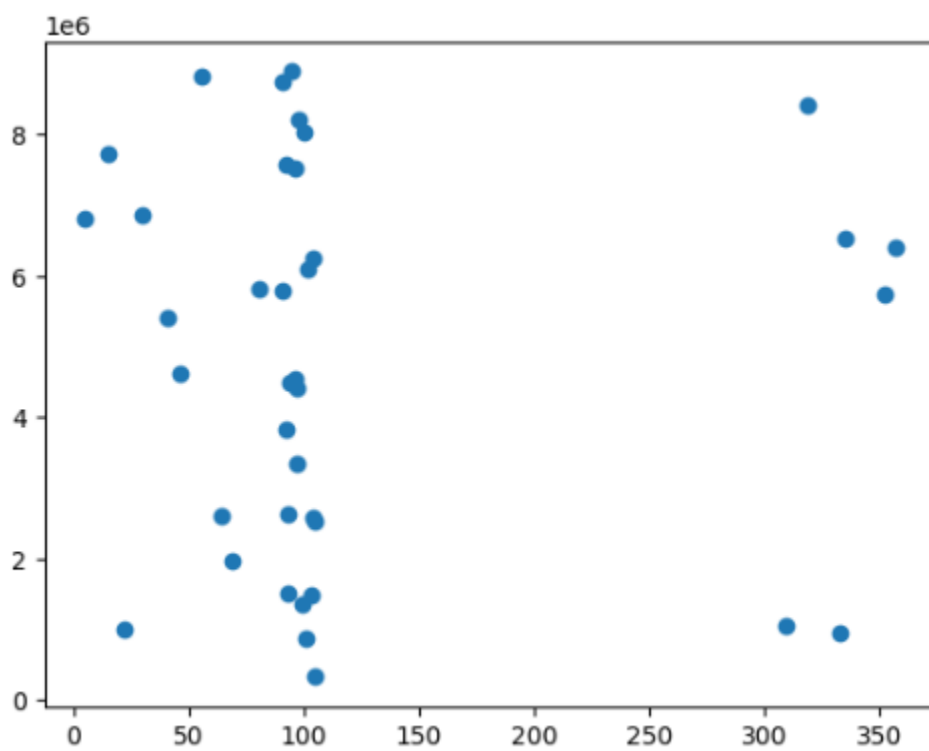


7. Distribución normal de los precios de los productos





8. Cantidad de pedidos registrados por día a través del año





9. Regresión lineal para el comportamiento predictivo de las ventas a través del los dias de año

