

Universidad Mariano Gálvez, Sede Naranjo

Ingeniería en Sistemas

Jornada Sabatina

Sección "A"



Manual Técnico

Dostin Josue Morataya Alvarez 9490-18-1188

Julio Cesar Moreno Carranza 9490-17-675

Guatemala, septiembre 2023

Introducción

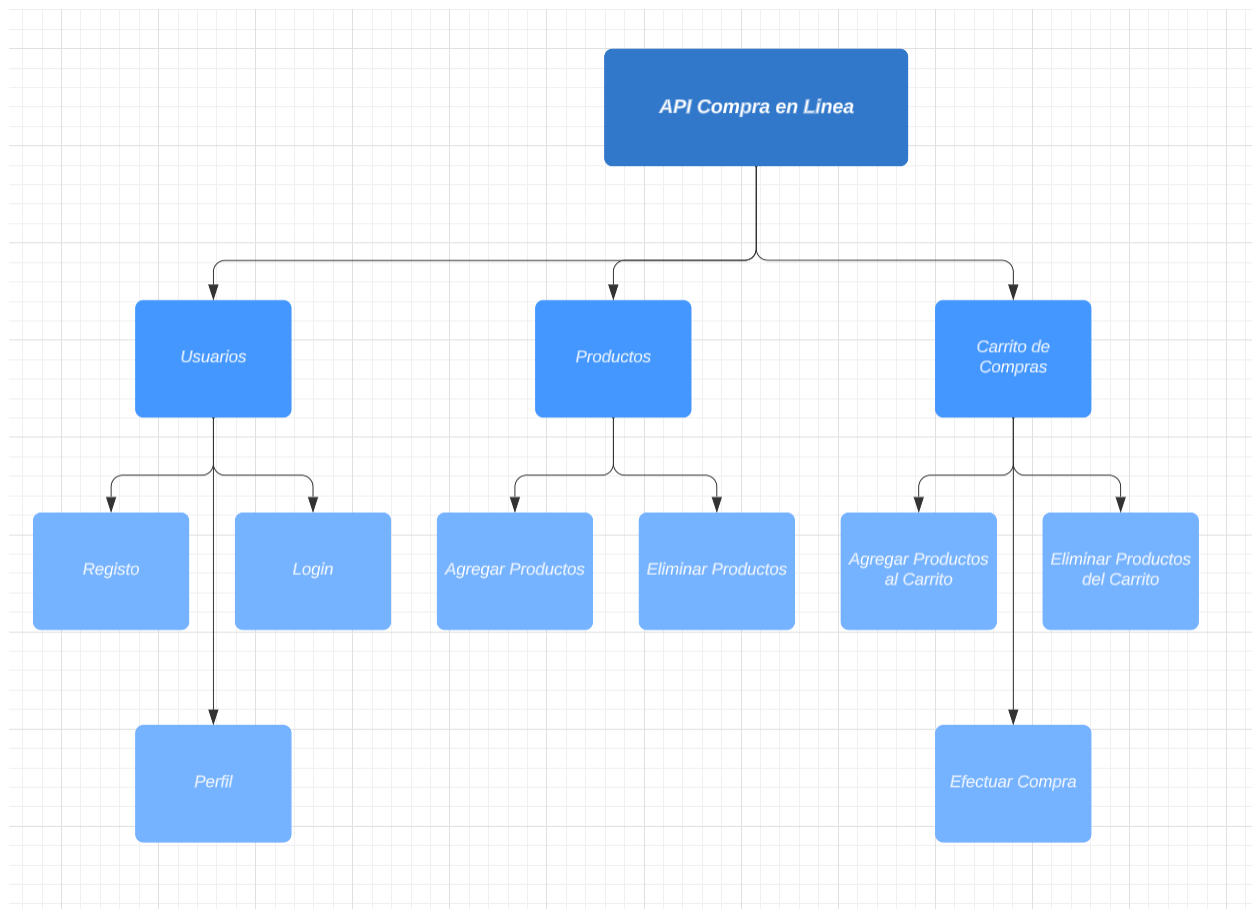
Este manual describe las herramientas utilizadas, así como la definición, diseño, organización y estructura de la API.

Esta API está diseñada para que un usuario normal pueda efectuar compras en línea, teniendo una cuenta y viendo y seleccionando los productos que desea comprar. Los usuarios administradores por otro lado se les permite hacer la gestión del sistema, como la inclusión de nuevos productos, etc.

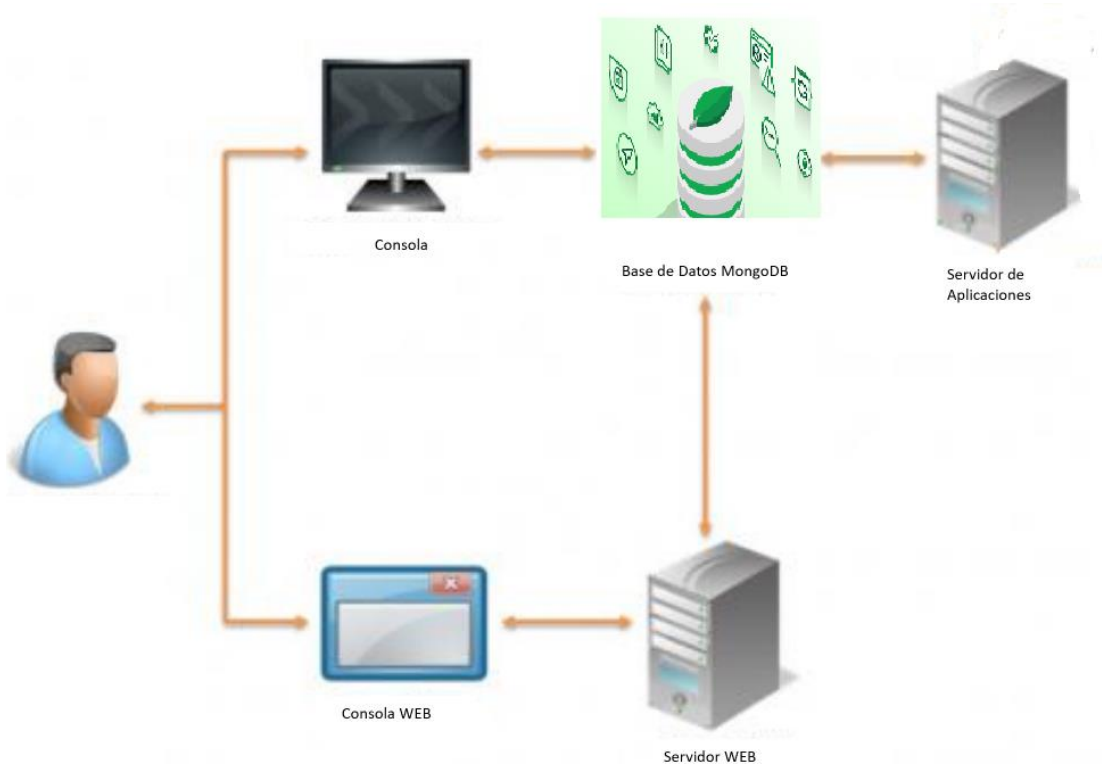
Tabla de contenido

Esquema Conceptual	1
Esquema Lógico y Físico	2
Herramientas.....	2
MongoDB.....	2
Visual Studio Code	2
Node.js	3
Descripción de las tablas	3
Bitácoras.....	3
Carritos	4
Productos	4
Usuarios	5
Descripción de la API	5
Descripción de los Endpoints utilizados	6

Esquema Conceptual



Esquema Lógico y Físico



Herramientas

MongoDB

Es un sistema de base de datos NoSQL, orientado a documentos y de código abierto.

En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

Visual Studio Code

Es un editor de código fuente creado por Microsoft con Electron Framework , para Windows , Linux y macOS.

Las características incluyen soporte para depuración, resaltado de sintaxis, finalización inteligente de código, fragmentos, refactorización de código y Git integrado. Los usuarios pueden cambiar el tema, los atajos de teclado, las preferencias e instalar extensiones que añaden funcionalidad.

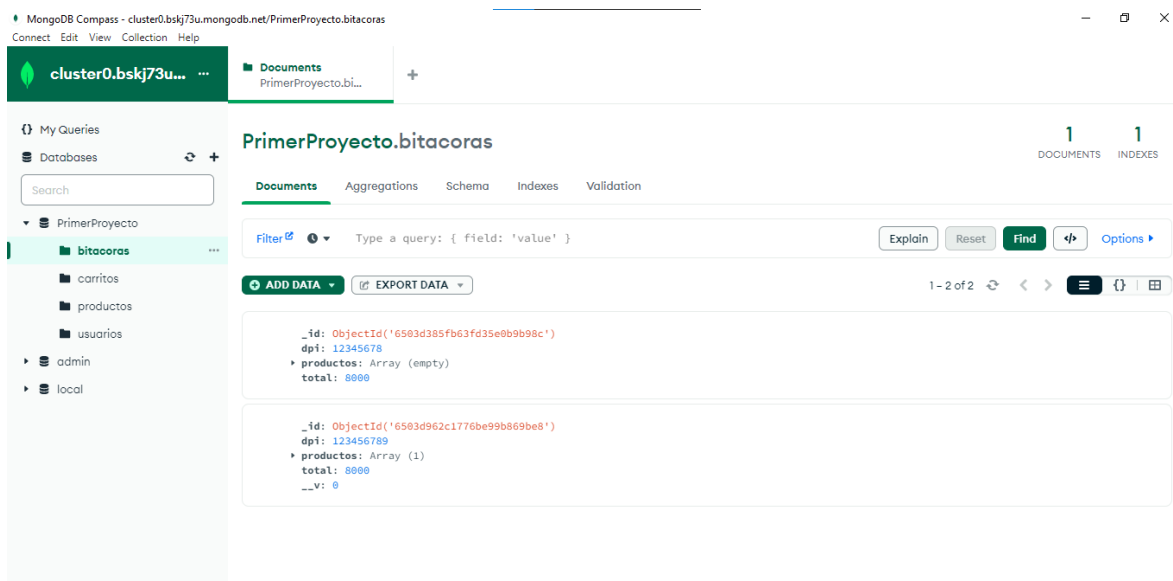
Node.js

Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web.

Descripción de las tablas

Bitácoras

En esta tabla se guardan las transacciones de compra de los usuarios.



Carritos

En esta tabla se encontrarán los diferentes carritos de los clientes que aun se encuentran pendientes de vender, aquí los usuarios aun pueden modificar los productos que se encuentran en sus respectivos carritos de compra.

MongoDB Compass - cluster0.bskj73u.mongodb.net/PrimerProyecto.carritos

Documents: PrimerProyecto.carritos

1 DOCUMENTS 1 INDEXES

PrimerProyecto.carritos

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } Explain Reset Find Options

ADD DATA EXPORT DATA 1 - 2 of 2

```
{
  "_id": ObjectId("6503d36afb63fd35e0b9b989"),
  "dpi": 12345678,
  "productos": Array (empty),
  "total": 8000
}
```

```
{
  "_id": ObjectId("6503d874c1776be99b869bdc"),
  "dpi": 123456789,
  "productos": Array (empty),
  "total": 0,
  "___v": 1
}
```

> MONGOSH

Productos

En esta tabla se encontrarán los diferentes productos que vende la empresa, se detalle el precio de venta, descuento si lo hubiera, así como, las características del mismo.

MongoDB Compass - cluster0.bskj73u.mongodb.net/PrimerProyecto.productos

Documents: PrimerProyecto.p...

2 DOCUMENTS 1 INDEXES

PrimerProyecto.productos

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } Explain Reset Find Options

ADD DATA EXPORT DATA 1 - 2 of 2

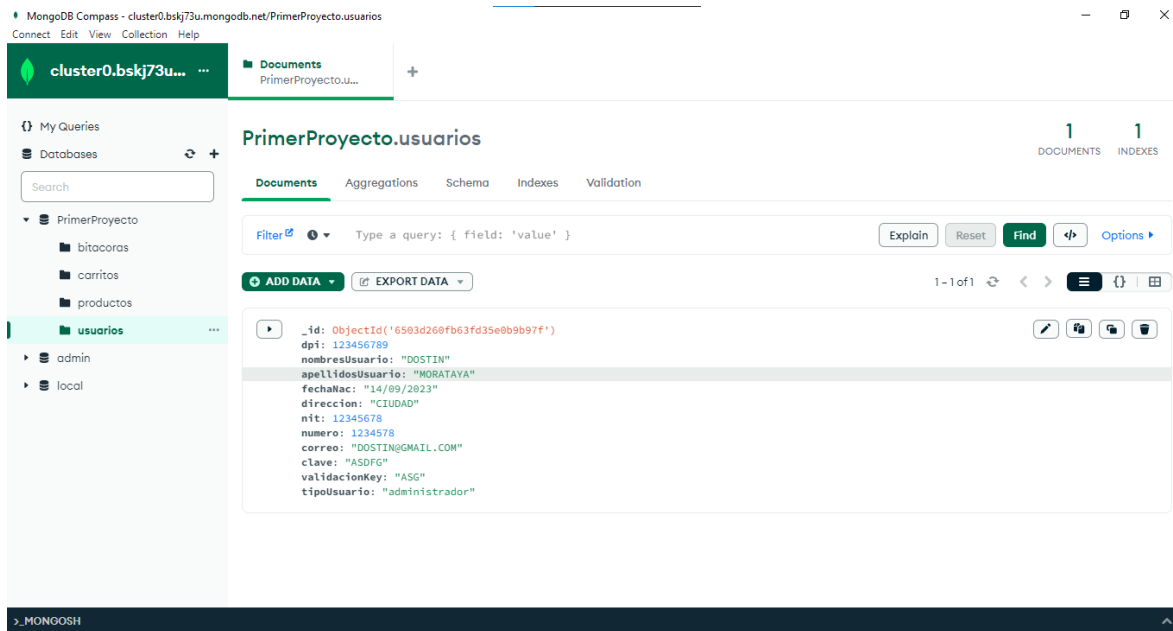
```
{
  "descuento": 1000,
  "precio": 9000,
  "precioDescuento": 8000,
  "imagen": "C:/img",
  "descripcion": "Computadora Dell",
  "categorias": Array (2),
  "habilitado": 1
}
```

```
{
  "_id": ObjectId("65079aafb63fd35e0b9b99e"),
  "idProducto": 11,
  "nombre": "Disco Duro",
  "marca": "Kingston",
  "stock": 25,
  "descuento": 500,
  "precio": 2500,
  "imagen": "ssd.png",
  "descripcion": "Capacidad 1TB ",
  "categorias": Array (2),
  "habilitado": 0
}
```

> MONGOSH

Usuarios

En esta tabla se encuentran el listado de los usuarios que se encuentran registrados en la aplicación web, se detalla su DPI, nombre, apellido, fecha de nacimiento, dirección, NIT, número, correo, también está qué tipo de usuario tiene dentro de la aplicación.



Descripción de la API

Esta API está enfocada en una solución web que permitirá a los usuarios la compra de productos en línea.

Para llevar a cabo las compras deberán registrarse previamente, a partir de ese proceso podrá seleccionar los productos que desee y llenar su carrito de compras. Ya con todos los productos que desee en su carrito podrá efectuar la compra.

Para los administradores del negocio, podrán contar con usuarios administradores que estarán gestionando los productos dentro de la plataforma para dar de alta o baja un producto, así como, para actualizar la existencia de los mismos.

Descripción de los Endpoints utilizados

Este endpoint permite a los usuarios efectuar su login.

- <http://localhost:3000/proyecto1/login>

Este endpoint en método post permite a los usuarios registrarse, en método get se puede ver la información del usuario, en método patch los usuarios pueden actualizar sus datos, en método delete permite al usuario a borrar su perfil.

- <http://localhost:3000/proyecto1/perfil/123456789>

Este endpoint en método get permite a los usuarios el ver los productos, en método post permite a los usuarios administradores insertar productos.

- <http://localhost:3000/proyecto1/producto>

Este endpoint en método get permite a los usuarios ver un producto en específico, en método patch permite a los usuarios administradores modificar el producto.

- <http://localhost:3000/proyecto1/producto/11>

Este endpoint en método get permite a los usuarios ver los productos que están en su carrito, en método post permite a los usuarios agregar un producto a su carrito, en método delete permite a los usuarios quitar un producto de su carrito.

- <http://localhost:3000/proyecto1/carrito>

Este endpoint en método get permite a los usuarios ver que es lo que han comprado, en método post permite efectuar la compra de los productos.

- <http://localhost:3000/proyecto1/compra>