

Instituto Politécnico Nacional

Unidad Interdisciplinaria En
Ingeniería Y Tecnologías
Avanzadas

Proyecto Tienda Marvel

Integrantes: Araujo Becerra Ricardo

Barquera Ortiz Paulina

Profesor: DE LA CRUZ SOSA CARLOS

15 / 01 / 24

Modelo DB

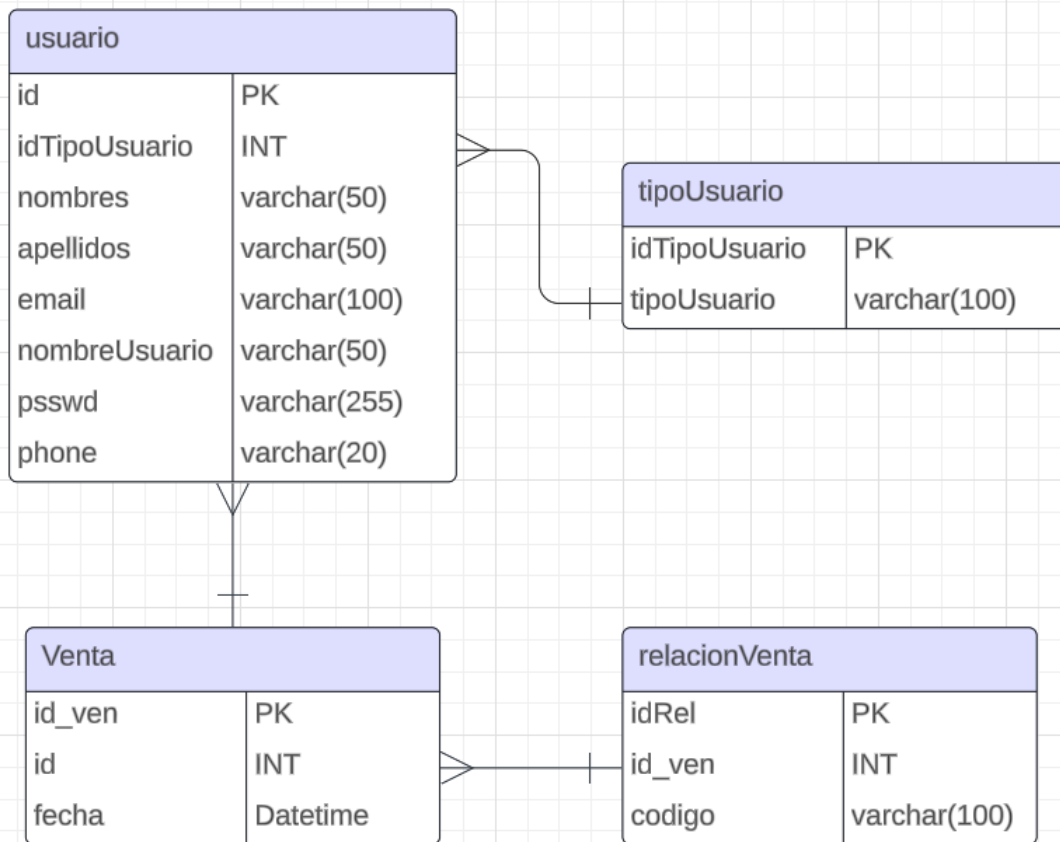
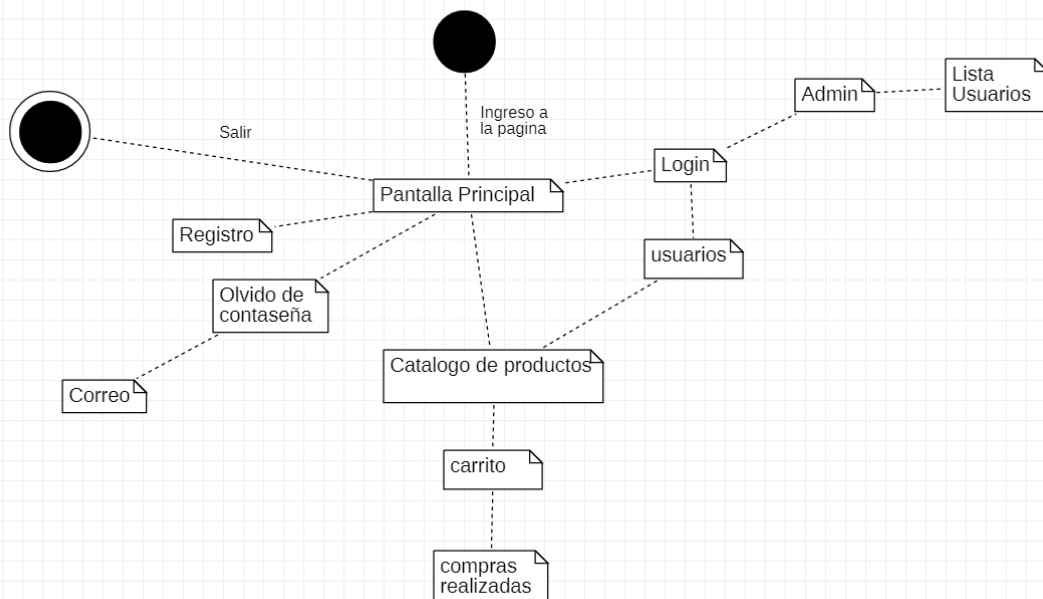


Diagrama de navegación



Estructura

Capeta controllers

Tenemos todas las instrucciones para ejecutar comandos con la base de datos aparte de los métodos para todo esto.

Carpeta database

Tenemos un archivo llamado db.js que se utiliza para la conexión a la base de datos especificando puertos y el nombre de la DB.

Carpeta node_modules

En ella se almacenan todas las dependencias del proyecto.

Carpeta views

En ella tenemos nuestras respectivas vistas junto con los css que se utilizan en cada una de ellas

Archivo app.js

En este archivo la configuración básica para iniciar un servidor web con Node.js y Express

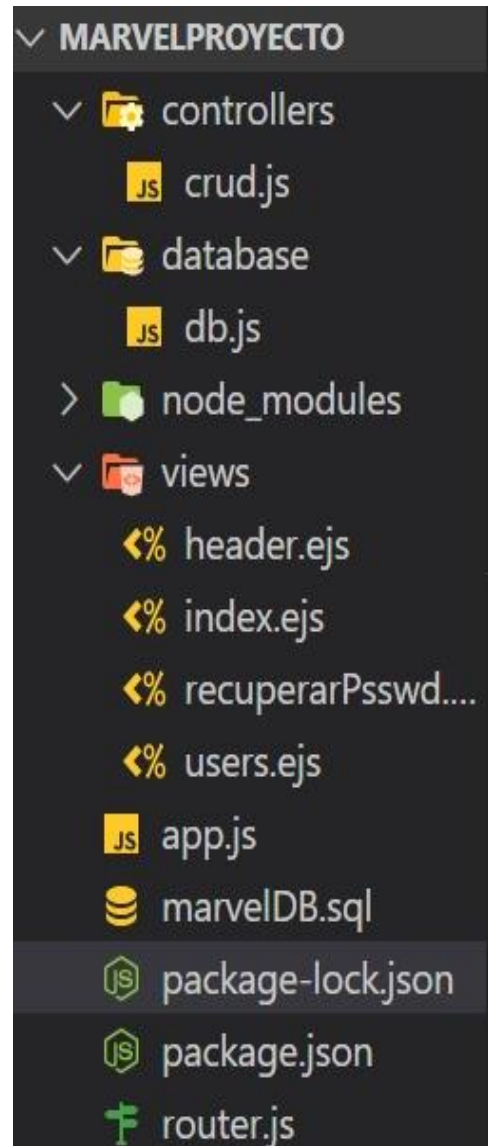
Package-lock.json y

package.json

En ellos tenemos nuestras dependencias para la app además de las instancias de ella

Router.js

En ella están especificadas todas las rutas de nuestro proyecto maneja la configuración de sesiones y realiza algunas consultas a la base de datos.



Dependencias de la aplicación

```
"dependencies": {
  "ejs": "^3.1.9",
  "express": "^4.18.2",
  "express-session": "^1.17.3",
  "mysql": "^2.18.1",
  "nodemailer": "^6.9.8",
  "nodemon": "^3.0.2"
}
```

EJS

Es un motor de plantillas para JavaScript que nos permite generar HTML de una manera dinámica. Se utiliza para incrustar código JavaScript en archivos HTML y facilitar la generación dinámica de contenido en la aplicación.

Express

Lo utilizamos para construir la aplicación con Node.js. Se utiliza para crear aplicaciones APIs de manera sencilla y eficiente. Algunas de las características clave de Express incluyen enrutamiento robusto, manejo de solicitudes y respuestas, middleware extensible, y soporte para la construcción rápida de APIs RESTful.

Express-session

Es utilizado en la app como gestor de sesiones para Express.js en Node.js. Permite a la aplicación web almacenar y gestionar la información de sesión del usuario a lo largo de las solicitudes HTTP. Las sesiones son útiles para realizar un seguimiento del estado del usuario entre las solicitudes, como la autenticación y la autorización.

Mysql

Es usado para conectarte a la base de datos MySQL y ejecutar consultas desde la aplicación.

Nodemailer

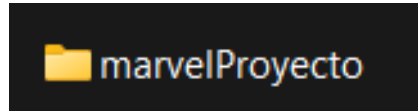
Utilizado como una biblioteca de Node.js que facilita el envío de correos electrónicos desde aplicaciones Node.js. Nos permite enviar correos electrónicos utilizando diferentes servicios de transporte, como SMTP, sendmail o Amazon SES, etc.

Nodemon

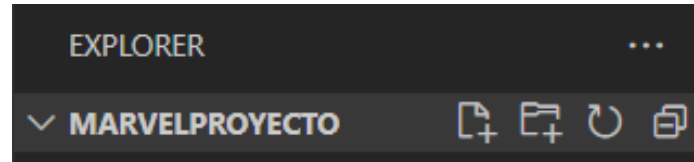
Útil como una herramienta que ayuda en el desarrollo reiniciando automáticamente la aplicación cuando detecta cambios en los archivos. Esto evita que tengamos que reiniciar manualmente el servidor cada vez que realizamos modificaciones en el código, además de poder ignorar ciertos directorios o archivos y especificar extensiones de archivo que deben observarse.

Información general para la configuración de la app.

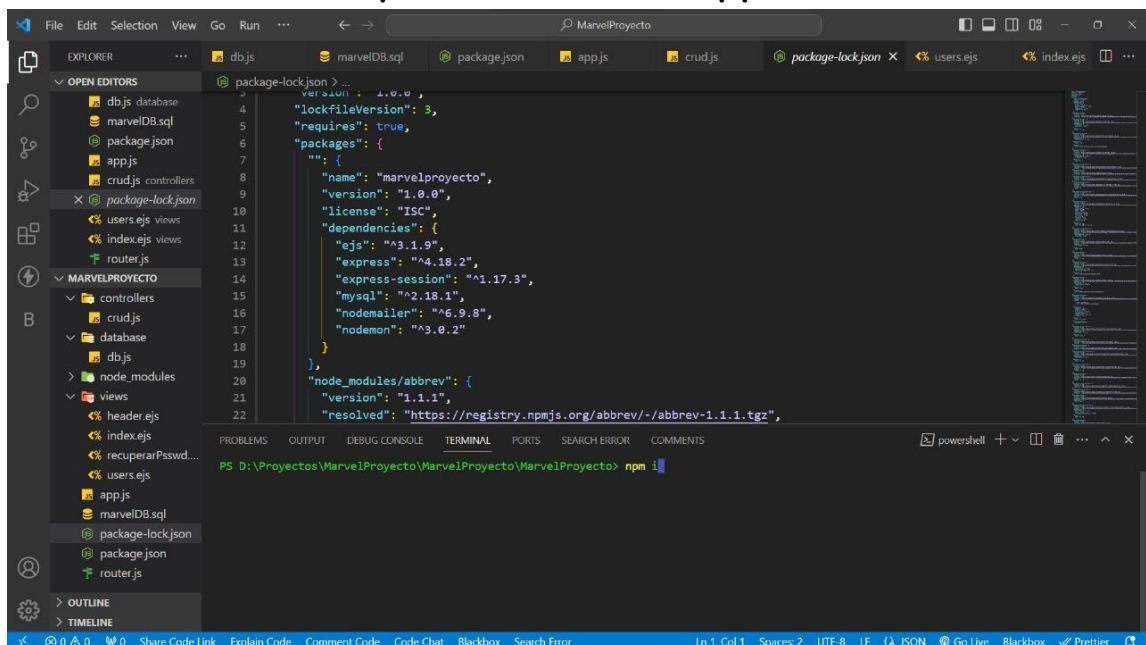
1.) Tener el archivo de la aplicación.



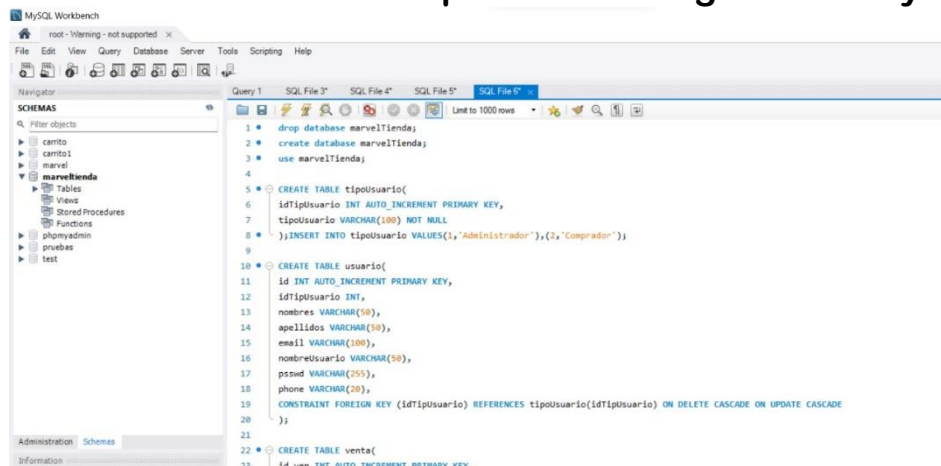
2.) Ejecutarlo desde Visual Studio Code.



3.) Abrir una terminal `ctr + ñ` y ejecutar `npm i` para instalar las dependencias de la app.



4.) Crear la base de datos con el script `MarveledB` que se encuentra en las carpetas del código desde MySQL.



5.)Configurar el usuario y contraseña en el archivo db.js

```
const conexion = mysql.createConnection({  
  host: '127.0.0.1',  
  port: 3306,  
  user: 'root',  
  password: '',  
  database: 'marvelTienda'  
});
```

6.)Ejecutar npm run start para que se inicialice la aplicación

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  COMMENTS  
PS D:\Proyectos\MarvelProyecto\MarvelProyecto\MarvelProyecto> npm run start
```

7.)Ir al navegador de nuestro agrado y ejecutar <http://localhost:3000/> para ver nuestra aplicación.

