



proyecto

---

# Manual para elaborar una galeria de fotos en node.js

*Desarrollo de  
Aplicaciones Web*



## TABLA DE CONTENIDO

---

Introducción .....	3
1. Conociendo los materiales de trabajo....	4
Instalación inicial.....	4
2. creacion de la galeria.....	8
Creación de la base de datos y la conexión de nuestra aplicación.....	8
Creacion de el archivo.ejs.....	10
Conclusión .....	12

# INTRODUCCIÓN

---

En el presente documento tiene como objetivo mostrar un poco el como usar el lenguaje de programación node.js y su framework express esto con la finalidad de que puedas llevar a cabo una serie de proyectos los cuales servirán para tener una idea de cómo usar esta tecnología.

# 1. CONOCIENDO LOS MATERIALES DE TRABAJO

---

Antes de empezar a explicar paso a paso como se llevó a cabo este proyecto En este manual se asume que el usuario ya tiene un conocimiento base sobre programación y node.js, en el manual se explicara de manera detallada como ir realizando el proyecto poco a poco no se hará una explicación del lenguaje sin embargo, para consultar información sobre node.js visite el siguiente enlace: <https://nodejs.org/es/docs/>

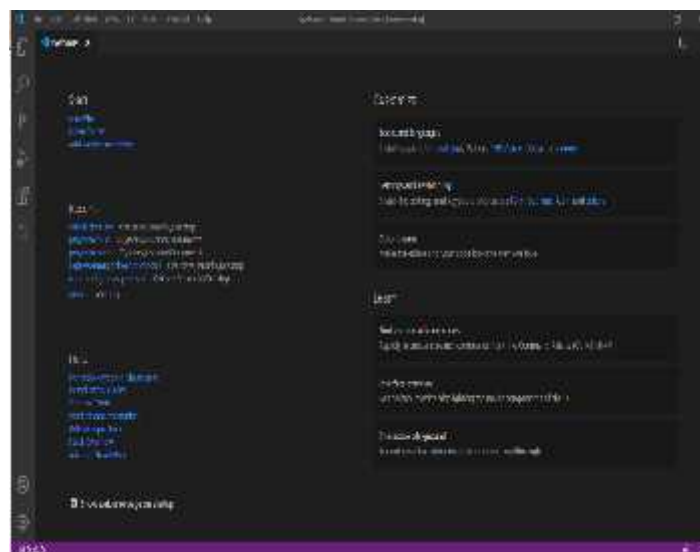
Requisitos necesarios para realizar el proyecto

- ) Tener instalado el node.js (adjunto en enlace de descarga: <https://nodejs.org/en/> )
- ) Descargar el framework express (adjunto el enlace de descarga: <http://expressjs.com/> )
- ) Tener instalado un gestor de BD (Mysql,MongoDB,MariaDB,Apache,etc)

## Instalación inicial

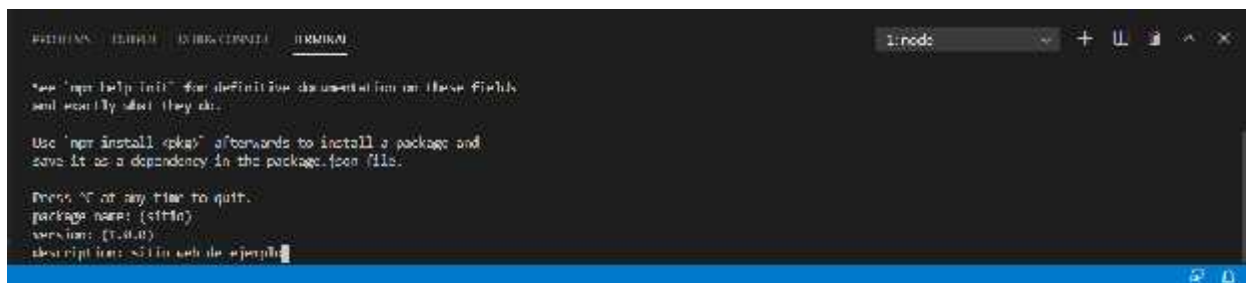
Creación de un archivo

Una vez instalado todos los programas requeridos lo que haremos después es crear una carpeta en documentos la cual una vez creada pues lo abrimos desde visual estudio code (file/open folder/sitio)



**Paso 1:** Ahora para trabajar desde la carpeta es necesario abrir terminal que es la que nos ayudara a trabajar con la línea de comandos con npm; el npm es el que va a permitir traer todos los paquetes y dependencias necesarios para el proyecto

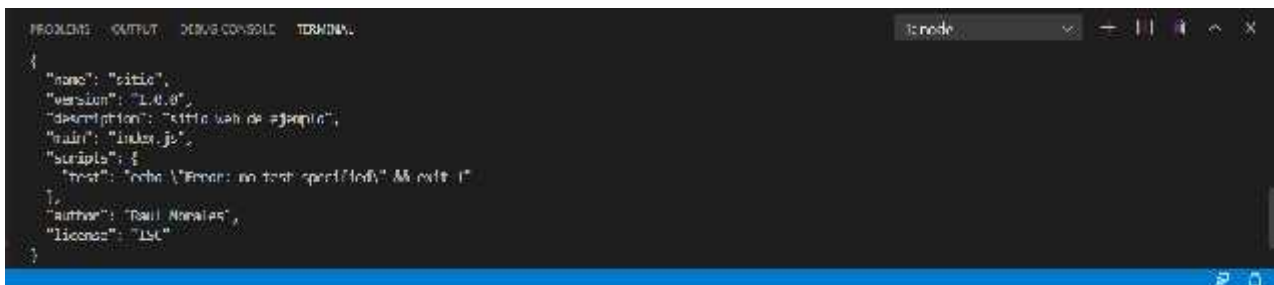
Lo primero es escribir en la línea de comandos npm init, esta pedirá después una serie de paquetes como lo es el nombre y versión el cual solo hacemos clic en enter para continuar hasta llegar a la parte de descripción donde se escribirá pues la descripción de la misma (en este caso se escribirá sitio web de ejemplo)



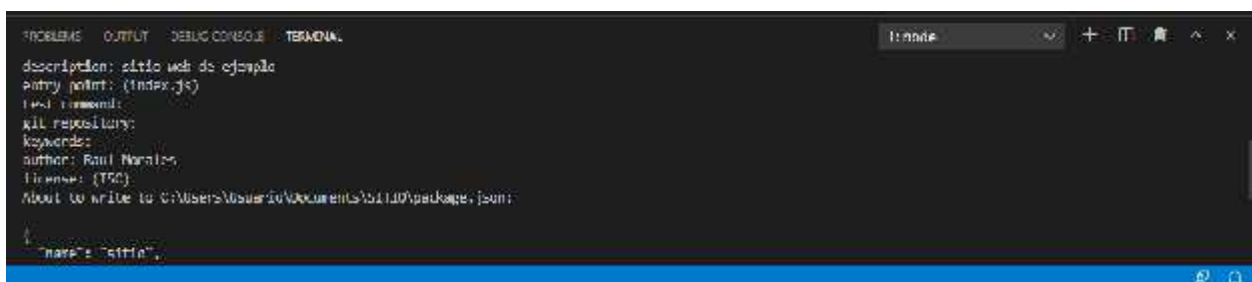
```
npm init
? package name (sitio):
? version (1.0.0):
? description: sitio web de ejemplo
```

**paso 2:** Después de escribir la descripción se te pedirán otros datos, estos simplemente es cuestión de darle continuar hasta que llegue la parte de el autor donde puedes escribir tu nombre hasta que te el programa te cree un archivo tipo package.json, el cual controlará toda la información y ya para último si toda la información es correcta solo escribes si y automáticamente el archivo package.json aparecerá en tu carpeta

Instalación de paquetes a través de npm

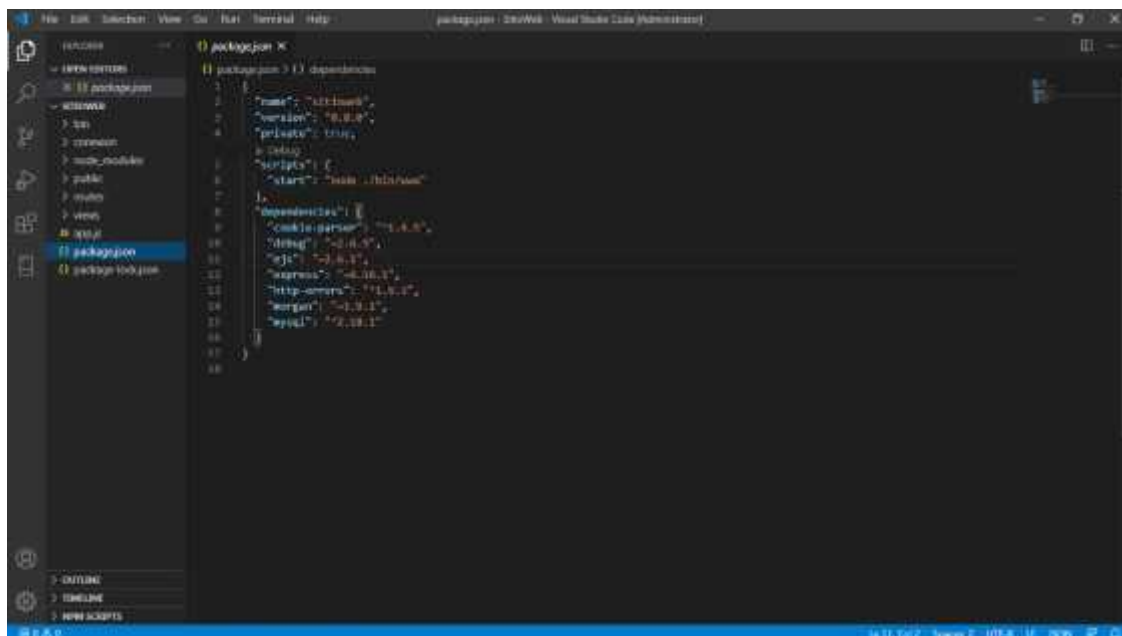


```
{
  "name": "sitio",
  "version": "1.0.0",
  "description": "sitio web de ejemplo",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Raul Morales",
  "license": "ISC"
}
```



```
npm init
description: sitio web de ejemplo
entry point: (index.js)
? package name:
? version:
? description:
? main:
? test:
? author:
? license:
? About to write to C:\Users\Usuario\Documents\sitio\package.json:
{
  "name": "sitio",
  "version": "1.0.0",
  "description": "sitio web de ejemplo",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Raul Morales",
  "license": "ISC"
}
```

Al terminar de ingresar los datos se nos genera un archivo JSON como el de la Siguiente figura:



**Paso 3:** Después instalaremos las estructuras necesarias, esto es muy importante hacerlo si no nuestro proyecto no funcionara, existen 2 tipos de estructuras, la primera que es de desarrollo se instala con el siguiente comando:

**(npm install express)**

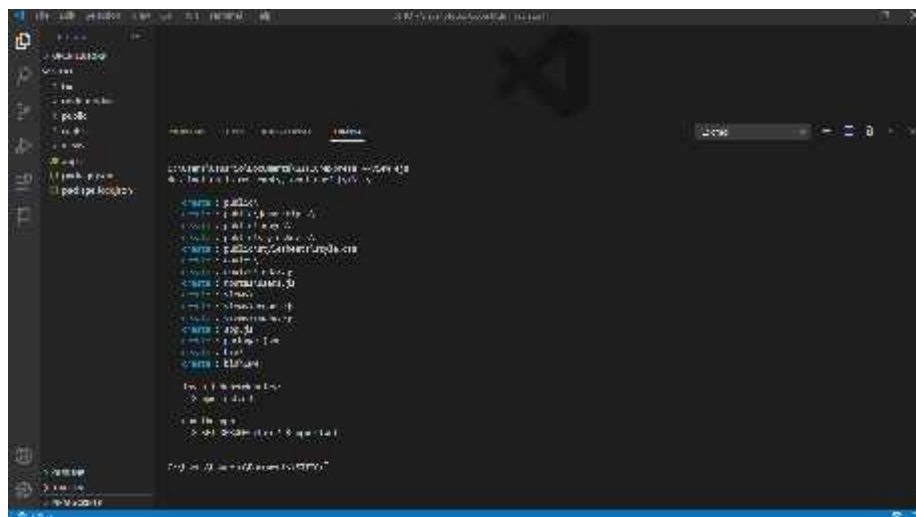
La segunda dependencia que es la de producción la cual sirve para crear una estructura base y se instala de la siguiente manera:

**(npm install express-generator -g)**

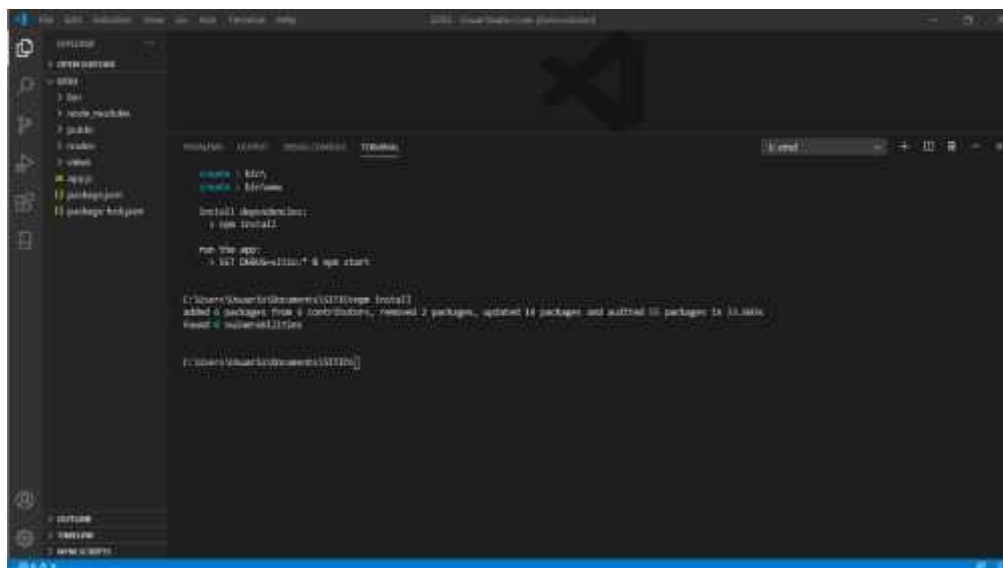
Después comenzaremos a crear nuestra estructura total la cual para realizarla utilizaremos este comando:

**(express -view=ejs)**

Seguidamente escribes si para aceptar que en ese mismo lugar crearas tú estructura como se muestra a continuación:



**Paso4:** una vez terminada la estructura procederemos a instalar las dependencias en esta paso se escribira el siguiente comando (**npm install**) como se muestra a continuacion:



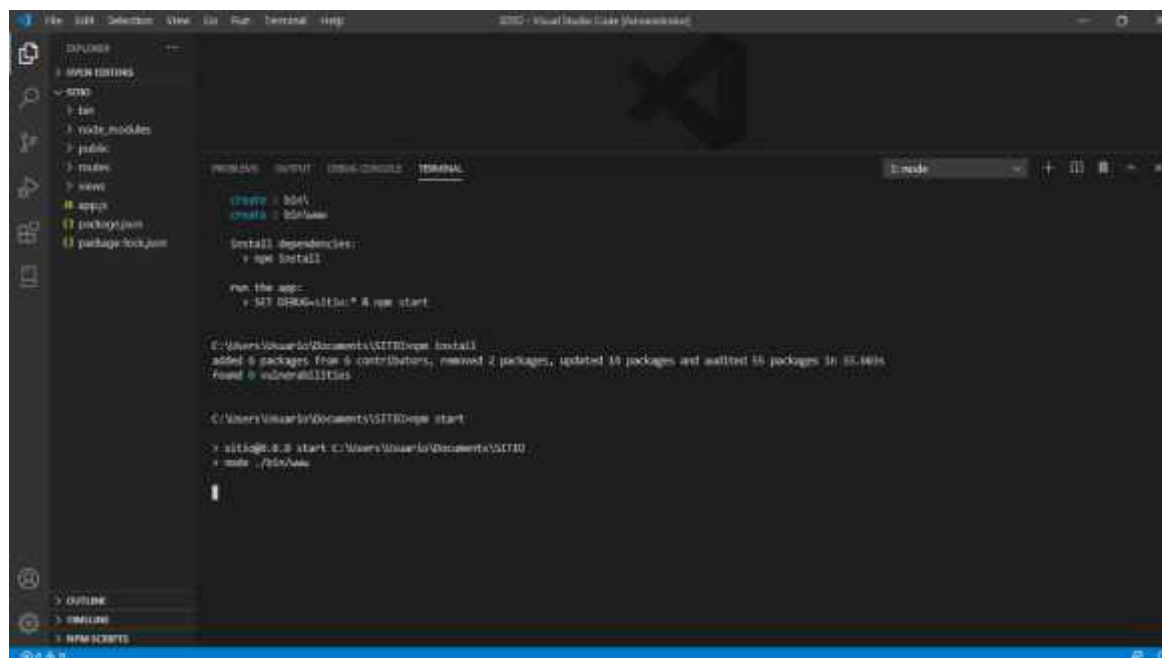
```
create > mkdir
create > cd /www
Install dependencies:
> npm install

run the app:
> set DEBUG=stio:* & npm start

C:\Users\Usuario\Documents\STIO>npm install
added 6 packages from 6 contributors, removed 2 packages, updated 14 packages and audited 15 packages in 11.041s
found 0 vulnerabilities

C:\Users\Usuario\Documents\STIO>
```

Ya para tener listo la estructura y verificar que todo este bien se escribira el siguiente comando: (**npm start**) si todo funciona bien nos debera aparecer una enunciado el cual afirma que ya esta creado el sitio. Con esto ya tenemos la estructura donde empezaremos a trabajar el proyecto.



```
create > mkdir
create > cd /www
Install dependencies:
> npm install

run the app:
> set DEBUG=stio:* & npm start

C:\Users\Usuario\Documents\STIO>npm install
added 6 packages from 6 contributors, removed 2 packages, updated 14 packages and audited 15 packages in 11.041s
found 0 vulnerabilities

C:\Users\Usuario\Documents\STIO> npm start
> stio@0.0.0 start: C:\Users\Usuario\Documents\STIO
> node ./bin/www
```

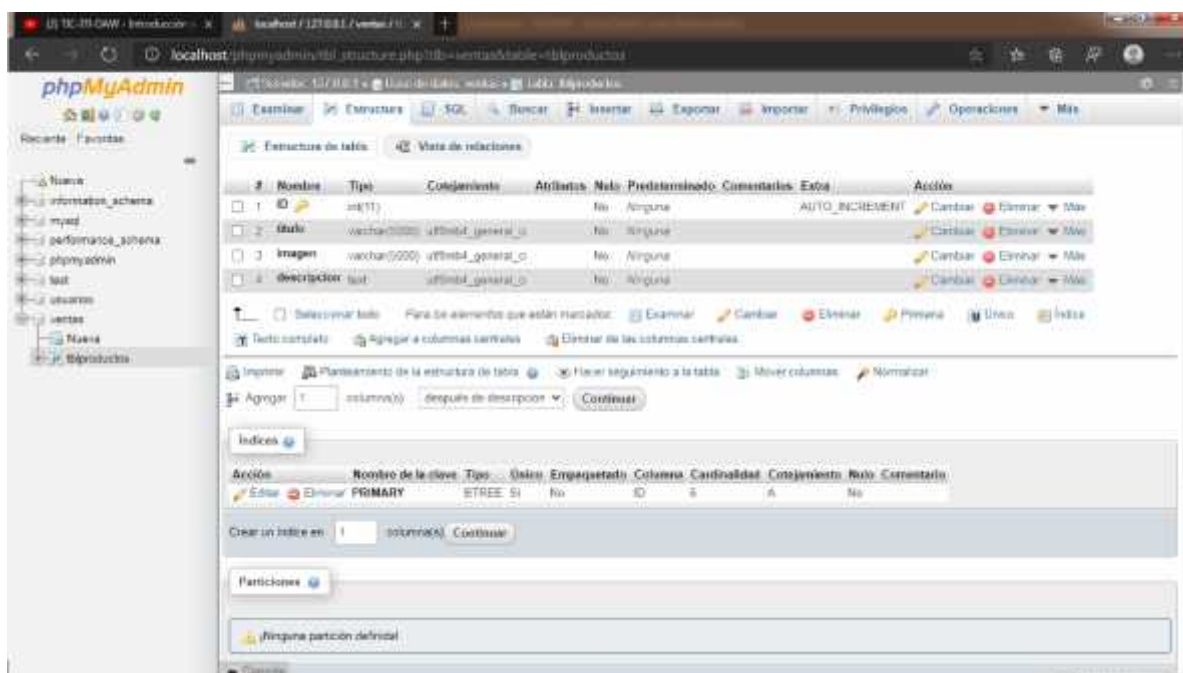
## 2. CREACION DE LA GALERIA

**Nota:** En este manual se asume que el usuario ya tiene un conocimiento base sobre programación y node.js, en el manual se explicara de manera detallada como ir realizando el proyecto poco a poco no se hará una explicación del lenguaje sin embargo, para consultar información sobre node.js visite el siguiente enlace: <https://nodejs.org/es/docs/>

### Creación de la base de datos y la conexión de nuestra aplicación

Para crear nuestra base de datos lo primero que se hará es dirigirse al localhost/phpmyadmin (en el caso de usar mysql/xampp) y lo que haremos es crear una nueva base de datos, luego se creara el archivo de conexión para conectar nuestra base de datos con la aplicación.

Paso1: creamos una nueva base de datos (en este caso nuestra base de datos se llamara ventas) y crearemos una tabla en donde agregaremos los datos correspondientes de la misma (en este caso los valores que tendrá la tabla son: id, titulo, imagen y descripción) como se muestra a continuación:



**Paso2:** en la parte de insertar es en donde agregaremos las imágenes y sus descripciones de esta manera.

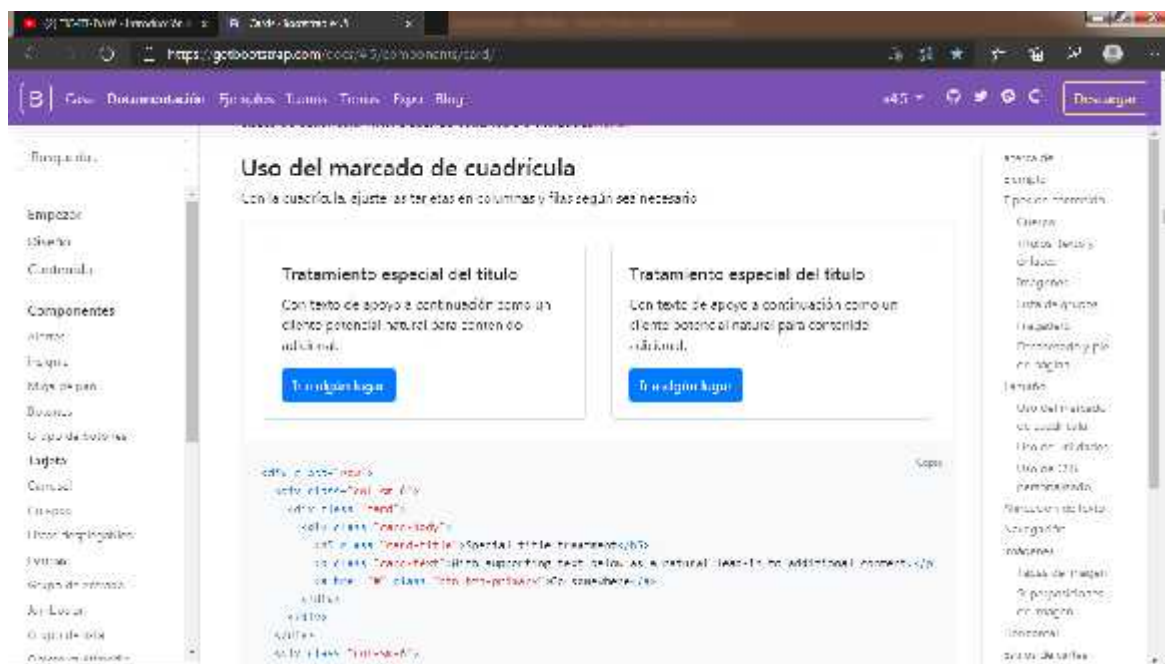




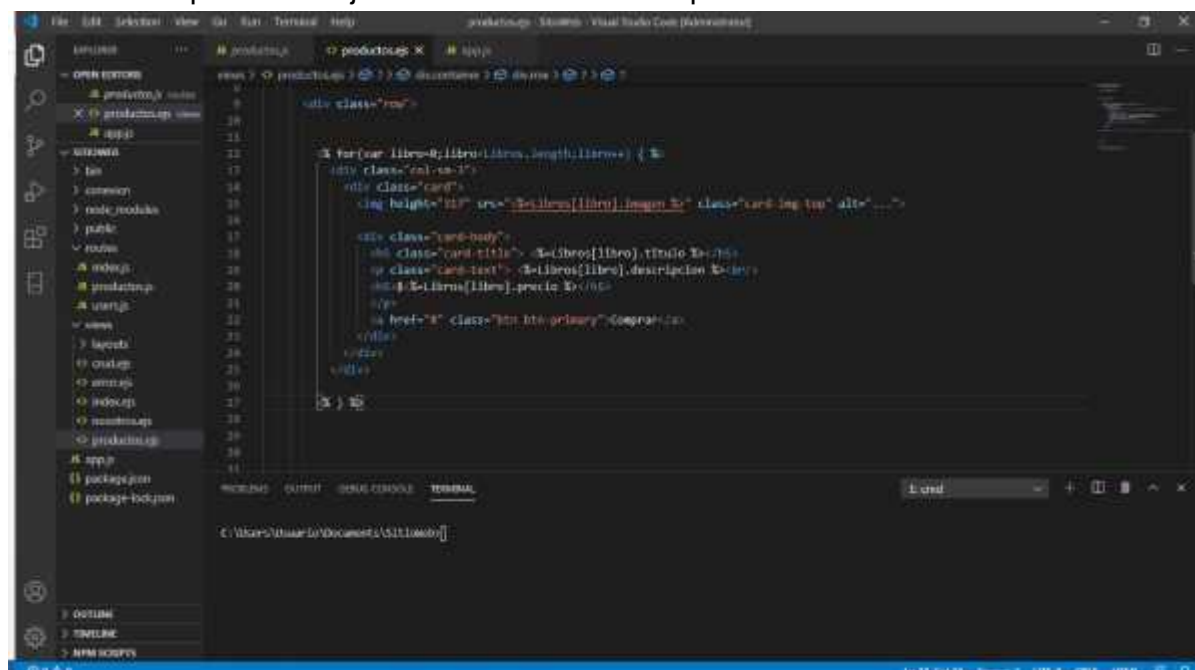
## Creacion de el archivo.ejs

Para empezar se creara un archivo en la parte de vistas (views) el cual se llamara productos.ejs este será la vista en donde se mostrara la galería de imágenes.

**Paso 1:** se dirigirá a la página de bootstrap (<https://getbootstrap.com/>) el cual servirá para darle una forma más personalizada a la vista de la galería, después hacer clic en empezar y dirigirse a la parte de componentes en donde se elegirá la opción de tarjeta(cards) y buscaremos la opción donde se puedan mostrar varias cards.

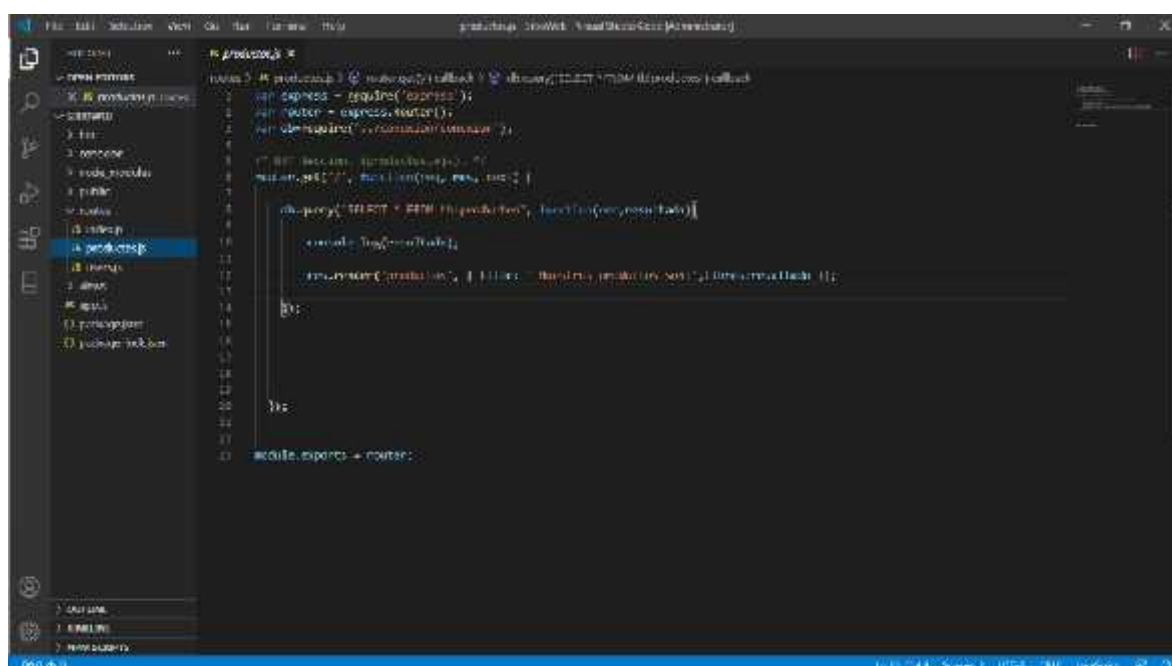


**Paso3:** Se copia el código que está en la parte de debajo del ejemplo y se pega el código en el archivo de productos.ejs. Este será nuestro template como se ve a continuación:



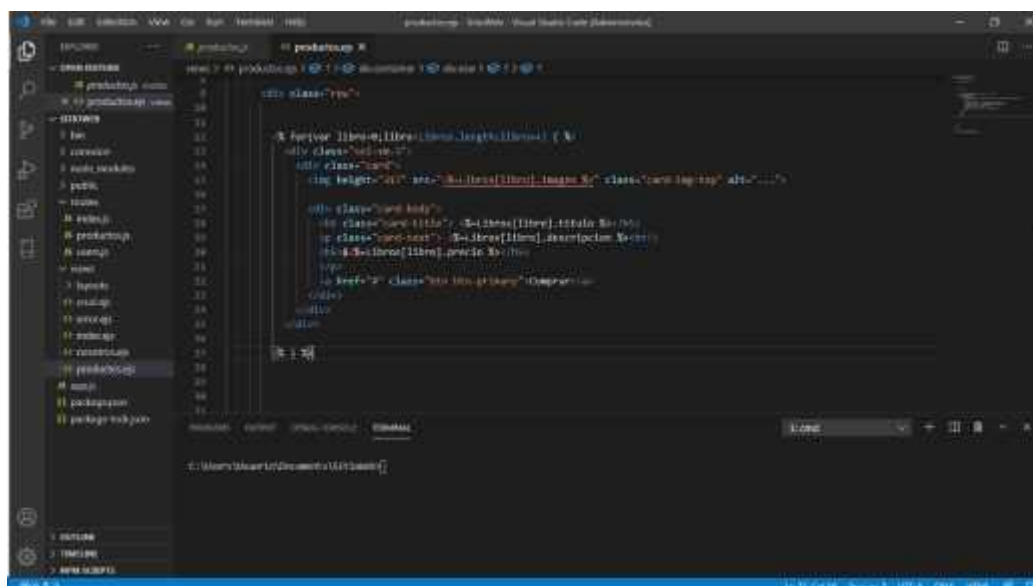
Una vez copiado el código se dirigen a la parte de rutas (routes)

Y creamos un archivo nuevo llamado productos.js en este caso lo que se realiza es la ruta hacia donde se visualizaran los datos del archivo productos.ejs esto primeramente desde el archivo app.js escribimos las rutas (en la parte debajo de `var usersRouter = require('./routes/users');`)  
`var productosRouter = require('./routes/productos');`  
después se escribirá el siguiente código



```
1 // routes/productos.js
2
3 const express = require('express');
4 const router = express.Router();
5 const productos = require('../models/productos');
6
7 // GET /productos
8 router.get('/', async (req, res, next) => {
9   const productos = await productos.find().exec();
10
11   res.render('productos', { productos });
12 });
13
14 module.exports = router;
```

**Paso5:** ya una vez hecha la conexión y la modificación del template nos debe quedar un archivo así



```
1 <%-- productos.ejs --%>
2
3 <%= title %>
4
5 <%= productos.length %> productos
6
7 <%=>
8   <div class="row">
9     <div class="col-md-3">
10       <img alt="Producto" data-bbox="100 100 200 200"/>
11     </div>
12     <div class="col-md-9">
13       <div class="card">
14         <div class="card-body">
15           <h3 class="card-title"><%= productos[0].nombre %></h3>
16           <p class="card-text"><%= productos[0].descripcion %></p>
17           <p class="card-text"><%= productos[0].precio %></p>
18         </div>
19       </div>
20     </div>
21   </div>
22 </%=>
```

## Conclusión

Ya terminado los archivos de conexión, rutas y témpate el resultado del proyecto nos debe mostrar esto

