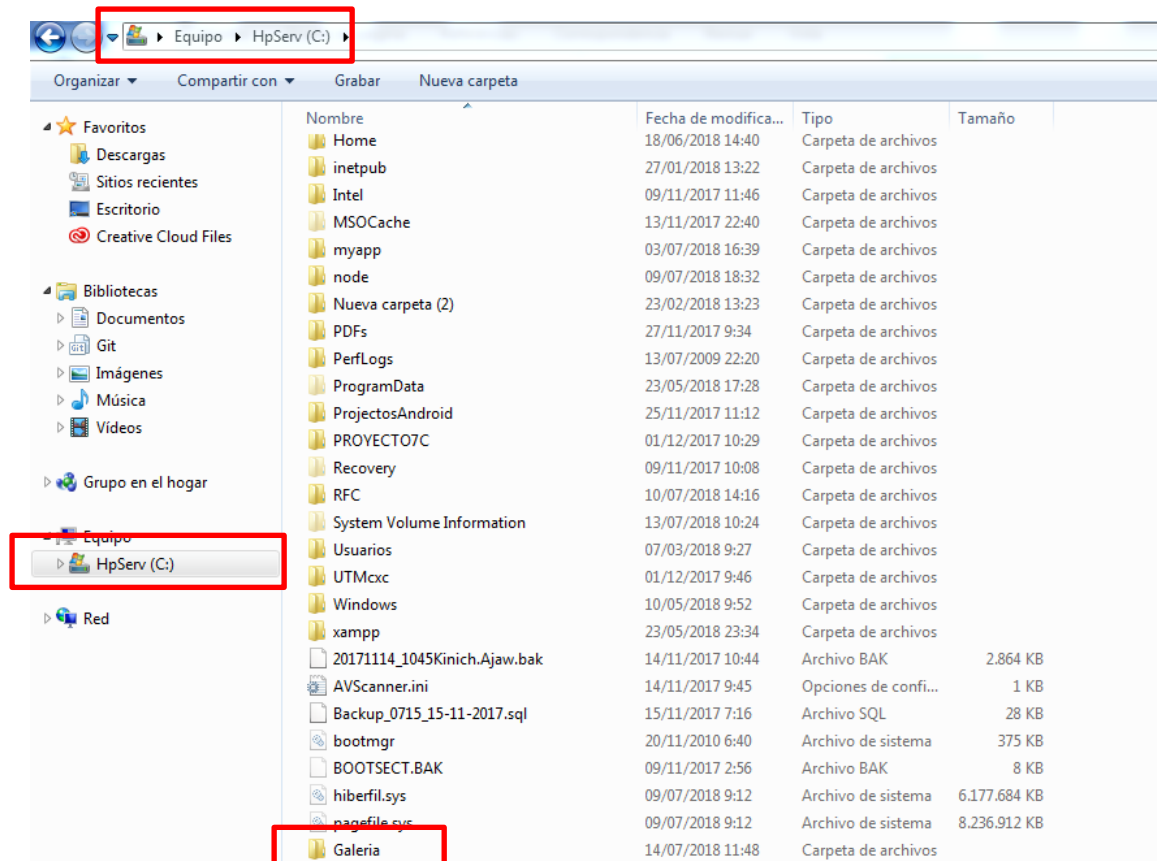


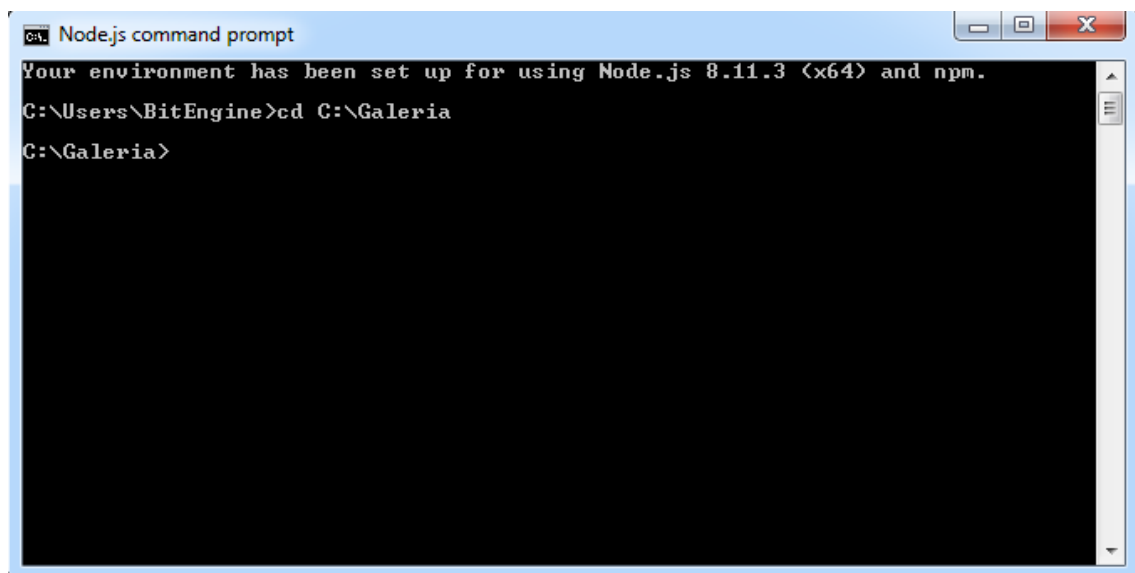
Manual para crear una galería de imágenes con (node.js, express y phpMyAdmin).



1.- Crearemos una carpeta llamada Galería en nuestro disco (C:)



2.- Abriremos la línea de comandos y escribiremos lo siguiente:



3.- Instalaremos express con la siguiente instrucción:

```

C:\Users\BitEngine>cd C:\Galeria
C:\Galeria>npm install express
npm WARN saveError ENOENT: no such file or directory, open 'C:\Galeria\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\Galeria\package.json'
npm WARN Galeria No description
npm WARN Galeria No repository field.
npm WARN Galeria No README data
npm WARN Galeria No license field.

+ express@4.16.3
added 50 packages in 18.557s
C:\Galeria>

```

4.- Creamos la siguiente instrucción, esto con el objetivo de crear un proyecto con express:

```

C:\Users\BitEngine>cd C:\Galeria
C:\Galeria>npm install express
npm WARN saveError ENOENT: no such file or directory, open 'C:\Galeria\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\Galeria\package.json'
npm WARN Galeria No description
npm WARN Galeria No repository field.
npm WARN Galeria No README data
npm WARN Galeria No license field.

+ express@4.16.3
added 50 packages in 18.557s
C:\Galeria>npm install express-generator -g
C:\Users\BitEngine\AppData\Roaming\npm\express -> C:\Users\BitEngine\AppData\Roaming\npm\node_modules\express-generator\bin\express-cli.js
+ express-generator@4.16.0
updated 1 package in 3.457s
C:\Galeria>

```

5.- Como siguiente paso crearemos toda la estructura del sitio web con la siguiente instrucción:

```

C:\Galeria>express --view=pug myapp

create : myapp\
create : myapp\public\
create : myapp\public\javascripts\
create : myapp\public\images\
create : myapp\public\stylesheets\
create : myapp\public\stylesheets\style.css
create : myapp\routes\
create : myapp\routes\index.js
create : myapp\routes\users.js
create : myapp\views\
create : myapp\views\error.pug
create : myapp\views\index.pug
create : myapp\views\layout.pug
create : myapp\app.js
create : myapp\package.json
create : myapp\bin\
create : myapp\bin\www

change directory:
> cd myapp

install dependencies:

```

6.- Instalaremos “cookie-parser” con la siguiente instrucción:

```

C:\Galeria>npm install cookie-parser

npm WARN enoent ENOENT: no such file or directory, open 'C:\Galeria\package.json'
npm WARN Galeria No description
npm WARN Galeria No repository field.
npm WARN Galeria No README data
npm WARN Galeria No license field.

+ morgan@1.9.0
added 3 packages in 2.145s

C:\Galeria>npm install cookie-parser

npm WARN enoent ENOENT: no such file or directory, open 'C:\Galeria\package.json'
npm WARN Galeria No description
npm WARN Galeria No repository field.
npm WARN Galeria No README data
npm WARN Galeria No license field.

+ cookie-parser@1.4.3
added 1 package in 1.631s

C:\Galeria>

```

7.- Instalaremos “morgan” con la siguiente instrucción:

```

Node.js command prompt

change directory:
> cd myapp

install dependencies:
> npm install

run the app:
> SET DEBUG=myapp:* & npm start

C:\Galeria>npm install morgan
npm WARN saveError ENOENT: no such file or directory, open 'C:\Galeria\package.j
son'
npm WARN enoent ENOENT: no such file or directory, open 'C:\Galeria\package.json'
npm WARN Galeria No description
npm WARN Galeria No repository field.
npm WARN Galeria No README data
npm WARN Galeria No license field.

+ morgan@1.9.0
added 3 packages in 2.145s
C:\Galeria>

```

8.- Como último punto instalaremos el motor de plantillas ejs para trabajar con las views así que escribiremos lo siguiente:

```

+ ejs@2.6.1
added 1 package in 1.158s
C:\Galeria>

```

9.- Una vez terminado todo el procedimiento de creación del proyecto escribimos lo siguiente:

```

C:\Galeria>npm start
> myapp@0.0.0 start C:\Galeria
> node ./bin/www

```

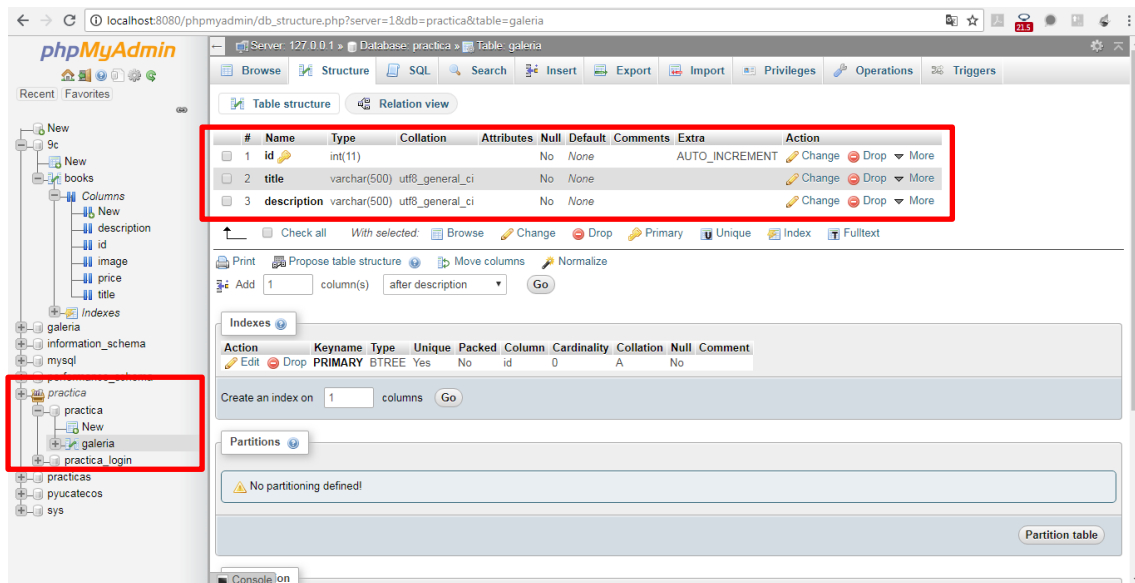
10.- Abrimos el navegador y escribimos la siguiente ruta para cotejar la instalación de express, si te salió en el navegador de la siguiente forma ya estarás listo para empezar con express js:



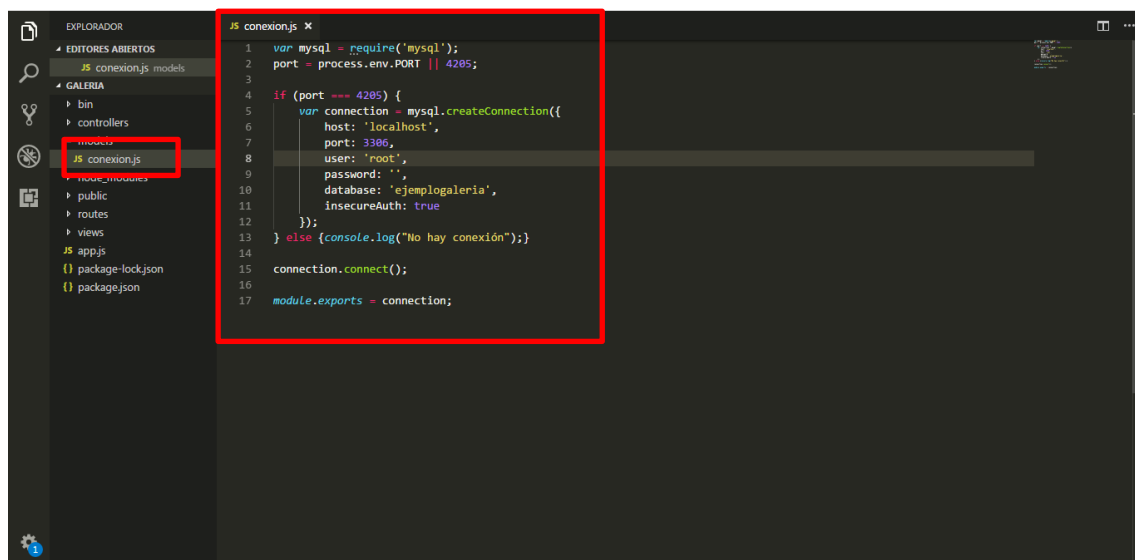
Express

Welcome to Express

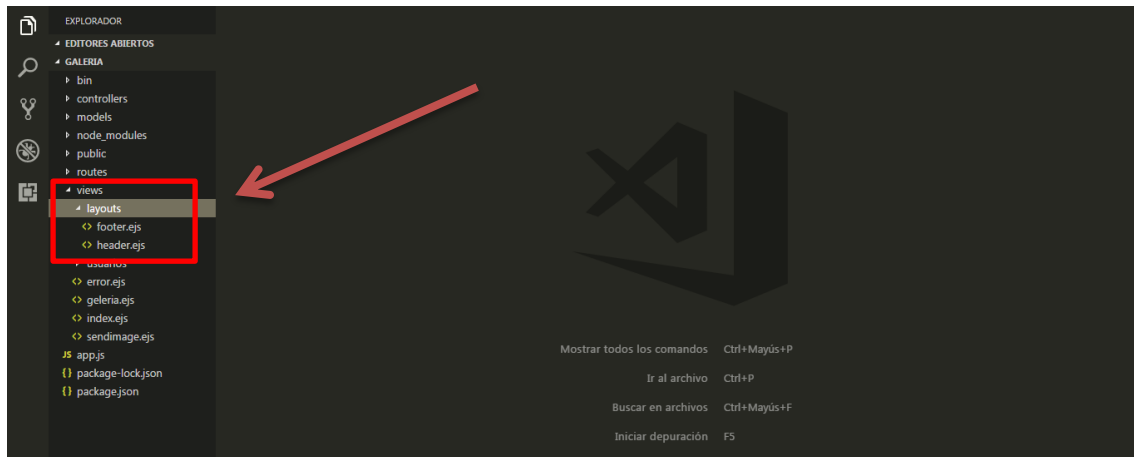
11.- Como siguiente paso abrimos phpMyAdmin para crear nuestra Base de Datos que alimentara a la galería de imágenes:



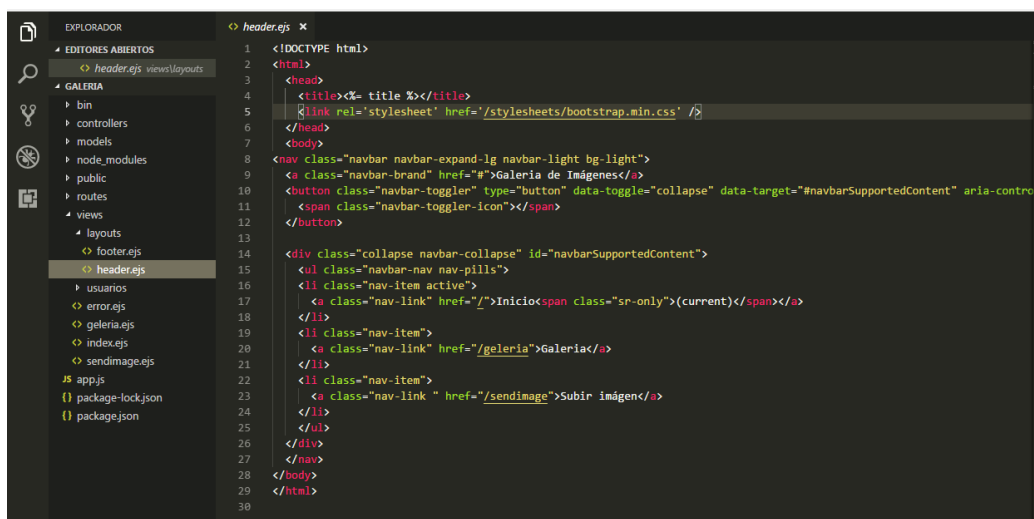
12.- Después crearemos una clase llamada “conexión.js” que será la encargada de conectarnos con la Base de Datos y obtener la información correspondiente:



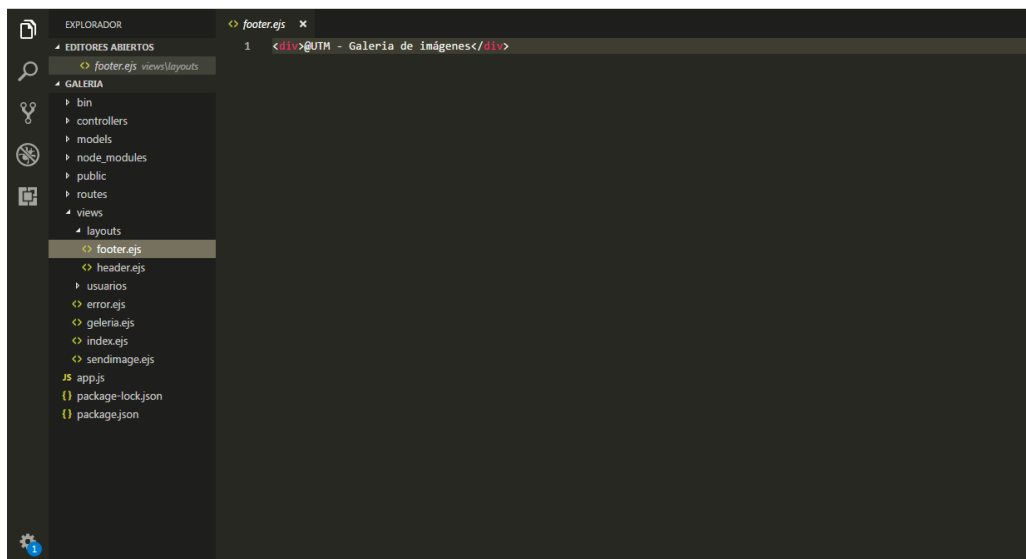
12.- Después crearemos una carpeta llamada “layouts” que contendrá dos clases llamadas “footer.ejs” y “header.ejs” las cuales le darán una mejor presentación a nuestra galería:



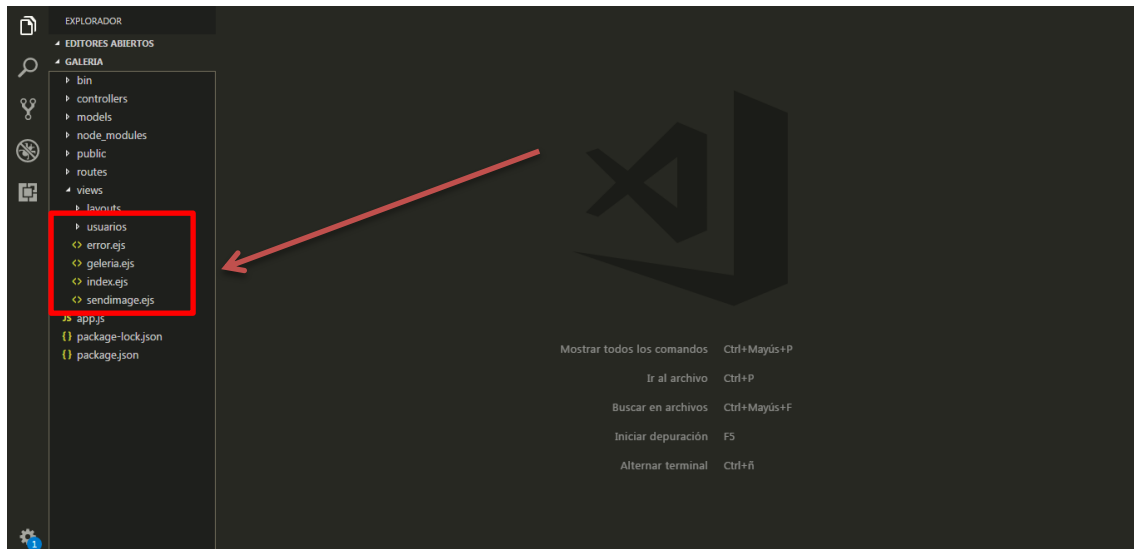
12.- La clase “header.ejs” contendrá lo siguiente:



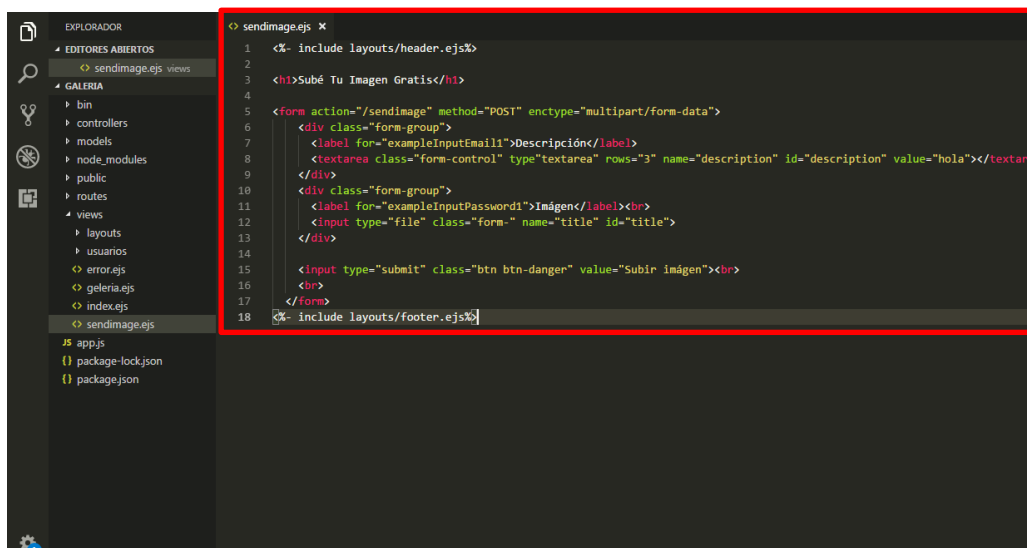
12.- La clase “footer.ejs” contendrá lo siguiente:



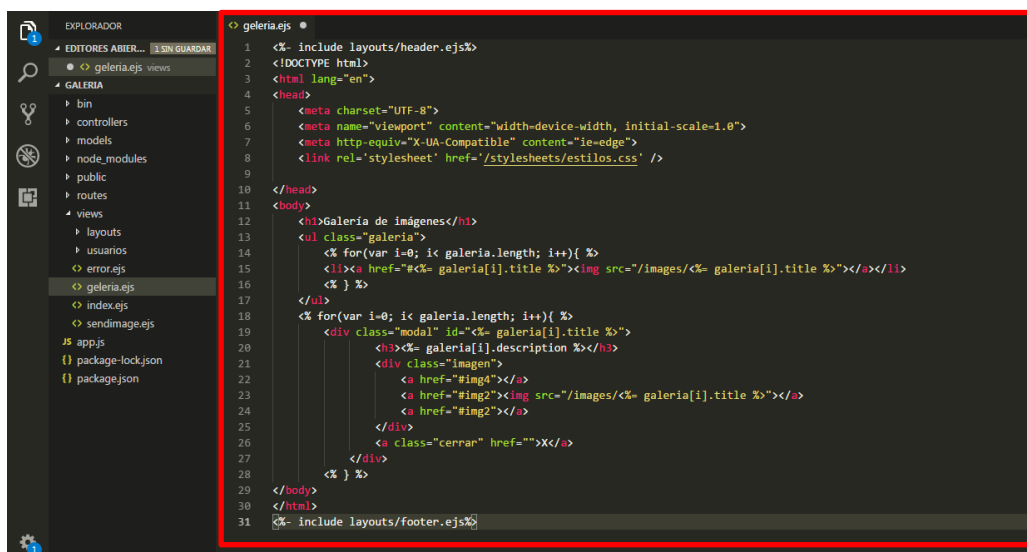
12.- Continuando en la carpeta “views” crearemos las clases “galeria.ejs” y “sendimage.ejs”, que serán las encargadas de la lógica de la galería:



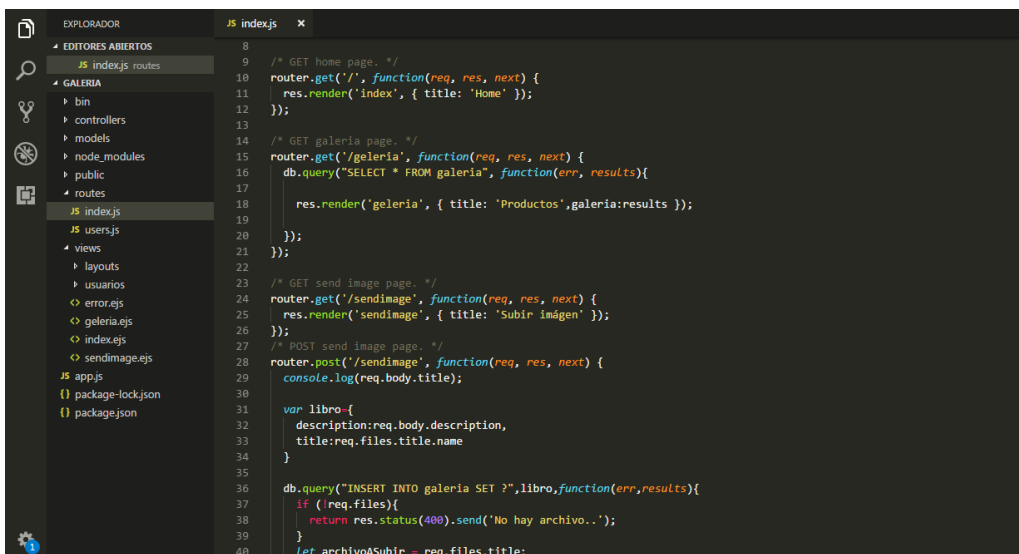
13.- La clase “sendimage.ejs” contendrá lo siguiente:



14.- La clase “sendimage.ejs” contendrá lo siguiente:



15.- En la clase “index.js” de la carpeta “routes” agregaremos las siguientes líneas de código que serán las que se encargaran de la lógica de nuestra galería:



```

8
9  /* GET home page. */
10 router.get('/', function(req, res, next) {
11   res.render('index', { title: 'Home' });
12 });
13
14 /* GET galeria page. */
15 router.get('/galeria', function(req, res, next) {
16   db.query("SELECT * FROM galeria", function(err, results){
17     res.render('galeria', { title: 'Productos', galeria: results });
18   });
19 });
20
21
22
23 /* GET send image page. */
24 router.get('/sendimage', function(req, res, next) {
25   res.render('sendimage', { title: 'Subir imagen' });
26 });
27
28 /* POST send image page. */
29 router.post('/sendimage', function(req, res, next) {
30   console.log(req.body.title);
31
32   var libro={
33     description:req.body.description,
34     title:req.files.title.name
35   }
36
37   db.query("INSERT INTO galeria SET ?",libro,function(err,results){
38     if (!req.files){
39       return res.status(400).send('No hay archivo..');
40     }
41     let archivoASubir = req.files.title;

```

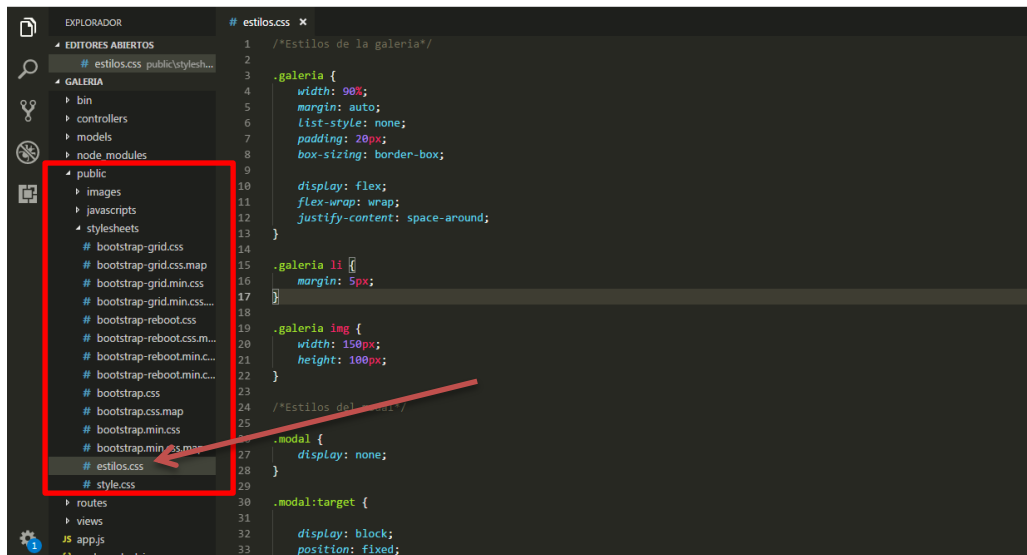


```

35
36   db.query("INSERT INTO galeria SET ?",libro,function(err,results){
37     if (!req.files){
38       return res.status(400).send('No hay archivo..');
39     }
40     let archivoASubir = req.files.title;
41
42     // Use the mv() method to place the file somewhere on your server
43     archivoASubir.mv('public/images/'+req.files.title.name, function(err) {
44       if (err)
45         return res.status(500).send(err);
46     });
47     res.render('sendimage', { title: 'Galeria' });
48   });
49
50
51
52 });
53
54 // Datos del arreglo books
55 /*var books=[
56   {id:1,title:"Node.js",description:"Node para todos",price:"300.00",img:"nodejs.jpg"},
57   {id:2,title:"Android",description:"Android para todos",price:"700.00",img:"libroAndroid.jpg"},
58   {id:3,title:"HTML5",description:"HTML5 para todos",price:"500.00",img:"html5.jpg"},
59 ],*/
60
61 //res.render('productos', { title: 'Productos', books: books });
62
63
64 module.exports = router;
65

```

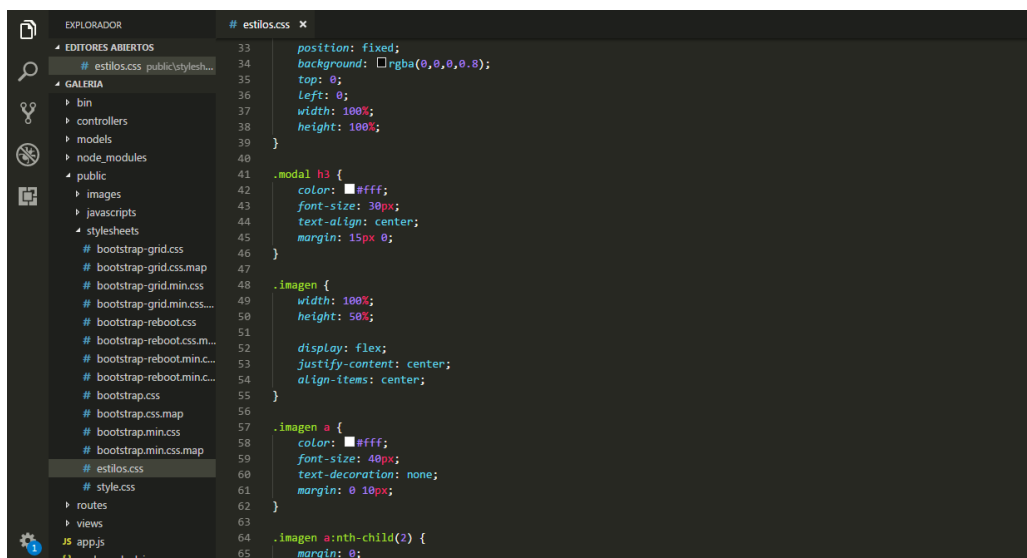
15.- Después crearemos el “estito.css” en la carpeta “stylesheets” la cual será la encargada de darle estilo a nuestra galería y contendrá lo siguiente:



```

# estilos.css
1 /*Estilos de la galeria*/
2
3 .galeria {
4   width: 90%;
5   margin: auto;
6   list-style: none;
7   padding: 20px;
8   box-sizing: border-box;
9
10  display: flex;
11  flex-wrap: wrap;
12  justify-content: space-around;
13 }
14
15 .galeria li {
16   margin: 5px;
17 }
18
19 .galeria img {
20   width: 150px;
21   height: 100px;
22 }
23
24 /*Estilos del modal*/
25
26 .modal {
27   display: none;
28 }
29
30 .modal:target {
31
32   display: block;
33   position: fixed;

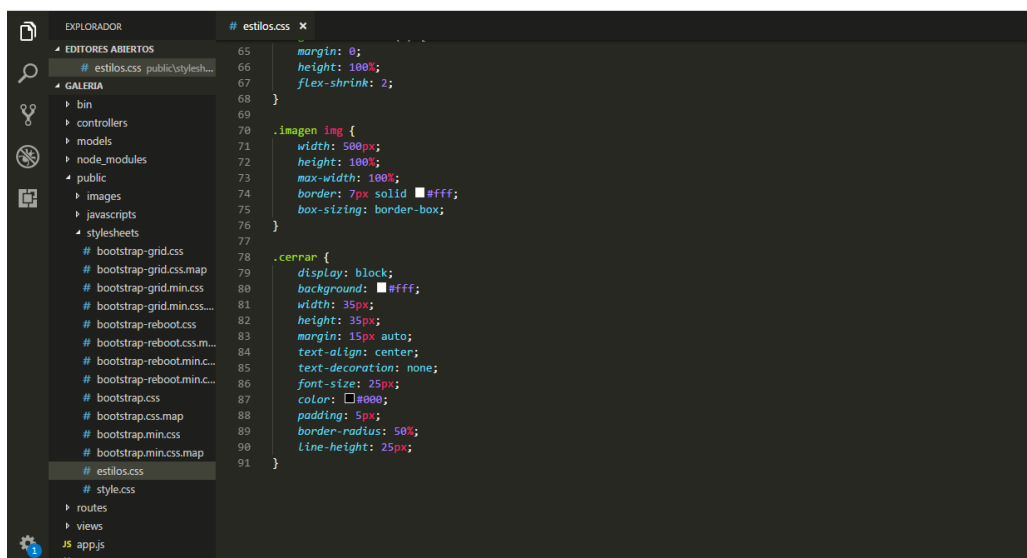
```



```

34   position: fixed;
35   background: rgba(0,0,0,0.8);
36   top: 0;
37   left: 0;
38   width: 100%;
39   height: 100%;
40 }
41
42 .modal h3 {
43   color: #fff;
44   font-size: 30px;
45   text-align: center;
46   margin: 15px 0;
47 }
48
49 .imagen {
50   width: 100%;
51   height: 50%;
52
53   display: flex;
54   justify-content: center;
55   align-items: center;
56 }
57
58 .imagen a {
59   color: #fff;
60   font-size: 40px;
61   text-decoration: none;
62   margin: 0 10px;
63 }
64
65 .imagen a:nth-child(2) {
66   margin: 0;

```



```

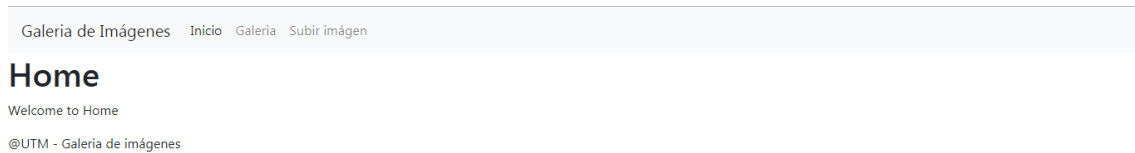
67   margin: 0;
68   height: 100%;
69   flex-shrink: 2;
70 }
71
72 .imagen img {
73   width: 500px;
74   height: 100%;
75   max-width: 100%;
76   border: 7px solid #fff;
77   box-sizing: border-box;
78 }
79
80 .cerrar {
81   display: block;
82   background: #fff;
83   width: 35px;
84   height: 35px;
85   margin: 15px auto;
86   text-align: center;
87   text-decoration: none;
88   font-size: 25px;
89   color: #000;
90   padding: 5px;
91   border-radius: 50%;
92   line-height: 25px;

```

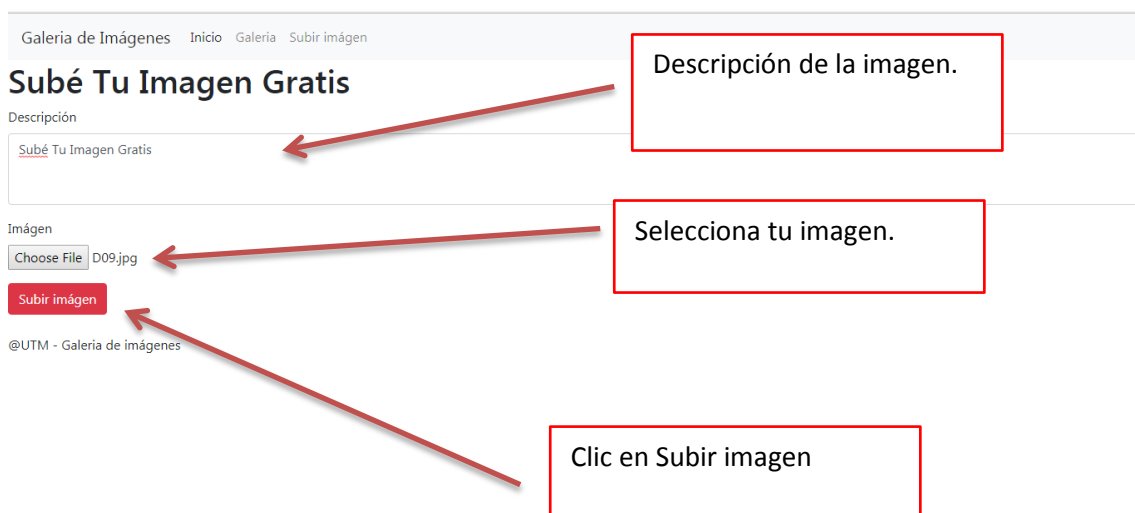
16.- Para finalizar iniciamos nuestro proyecto con el siguiente comando:

```
C:\RFC>npm start
> sitioweb@0.0.0 start C:\RFC
> node ./bin/www
```

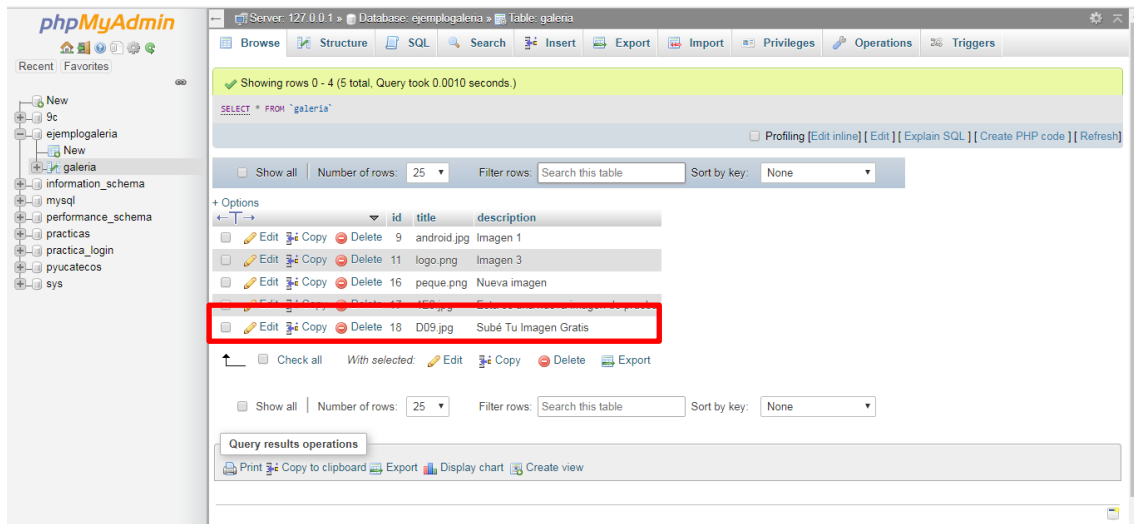
16.- Nos dirigimos a nuestro navegador y colocamos la siguiente dirección <http://localhost:3000/>:



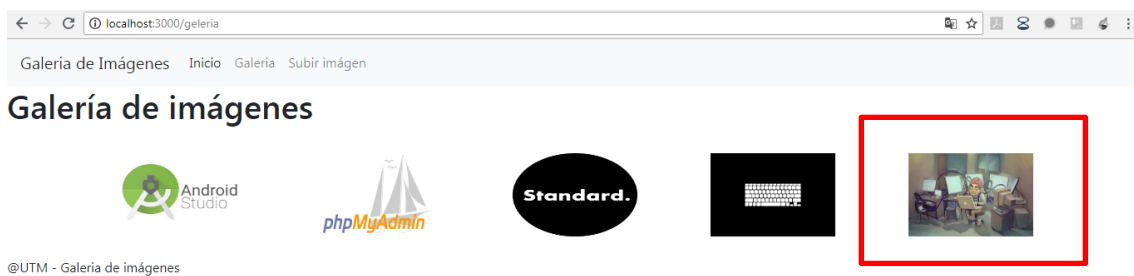
17.- Nos dirigimos a la pestaña “Subir imagen y llenamos los campos para insertar una imagen a la Base de Datos phpMyAdmin”:



18.- Verificamos en phpMyAdmin que se guardó el registro:



19.- Nos dirigimos a la pestaña “Galería” y podemos ver nuestras imagenes ya registradas en el BD:



20.- Si le damos clic a nuestra imagen registrada podremos ver una mejor vista de la imagen con la descripción que le pusimos:

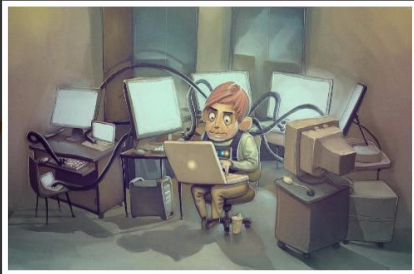
Galería de Imágenes Inicio Galería Subir imagen Subé Tu Imagen Gratis

Galería de imágenes

Android Studio

phpM

@UTM - Galería de imágenes



X