



# **Curso complementario Desarrollo de Back-end con Node.js - MongoDB**

**ING - DIEGO CASALLAS**

**22810019-1**



[www.sena.edu.co](http://www.sena.edu.co)

## Cargar archivos

- Validar las dependencias
  - **npm install multer**

```
11  "keywords": [],
12  "author": "",
13  "license": "ISC",
14  "description": "",
15  "dependencies": {
16    "bcrypt": "^6.0.0",
17    "body-parse": "^0.1.0",
18    "cookie-parse": "^0.4.0",
19    "dotenv": "^17.2.0",
20    "ejs": "^3.1.10",
21    "express": "^5.1.0",
22    "jsonwebtoken": "^9.0.2",
23    "mongoose": "^8.16.3",
24    "multer": "^2.0.2",
25    "nodemailer": "^7.0.5"
26  },
27  "devDependencies": {
28    "@types/express": "^5.0.3",
29    "nodemon": "^3.1.10"
30  }
```

# Node.js - Files

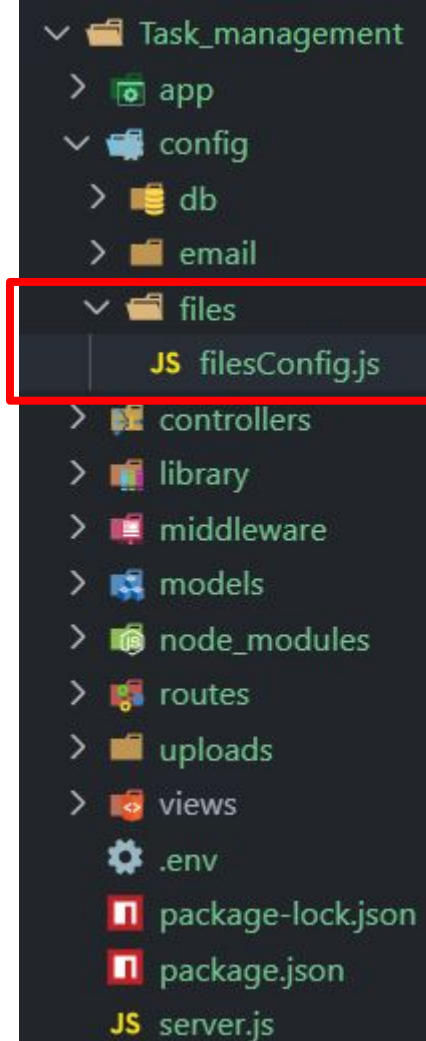


## Crear la carpeta files

- config/files->filesConfig.js

```
Task_management > config > files > JS filesConfig.js > [e] ensureUploadsDir
1  import multer from 'multer';
2  import path from 'path';
3  import { fileURLToPath } from 'url';
4  import fs from 'fs/promises';
5
6  // Get the current directory and file path
7  const __filename = fileURLToPath(import.meta.url);
8  const __dirname = path.dirname(__filename);
9
10 // Configurations for file uploads
11 const UPLOAD_DIR = path.join(__dirname, '../..../uploads');
12 // Define the path for the uploads directory
13 export const getFilesPath = UPLOAD_DIR;
14
15 // Create the uploads directory if it doesn't exist
16 const ensureUploadsDir = async () => {
17   try {
18     await fs.access(UPLOAD_DIR);
19   } catch {
20     await fs.mkdir(UPLOAD_DIR, { recursive: true });
21   }
22 };
23
```

CARPETA



```
Task_management
├── app
├── config
│   ├── db
│   ├── email
│   └── files
│       └── JS filesConfig.js
├── controllers
├── library
├── middleware
├── models
├── node_modules
├── routes
├── uploads
├── views
├── .env
├── package-lock.json
├── package.json
└── JS server.js
```

# Node.js - Files



- config/files->filesConfig.js

```
24 // Configure Multer storage
25 const storage = multer.diskStorage({
26   destination: async (req, file, cb) => {
27     await ensureUploadsDir();
28     cb(null, UPLOAD_DIR);
29   },
30   filename: (req, file, cb) => {
31     const uniqueSuffix = Date.now() + '-' + Math.round(Math.random() * 1E9);
32     const sanitized_name = file.originalname.replace(/^[a-zA-Z0-9-_.]/g, '');
33     cb(null, `${uniqueSuffix}-${sanitized_name}`);
34   }
35 });
36
37 // Filter to allow only specific file types
38 const fileFilter = (req, file, cb) => {
39   const allowedTypes = ['image/jpeg', 'image/png', 'image/gif', 'application/pdf'];
40   if (allowedTypes.includes(file.mimetype)) {
41     cb(null, true);
42   } else {
43     cb(new Error('Allowed file types: JPEG, PNG, GIF and PDF.'), false);
44   }
45 };
46
47 // Configure Multer upload
48 export const upload = multer({
49   storage,
50   limits: { fileSize: 5 * 1024 * 1024 }, // Límite de 5MB
51   fileFilter
52 });
53
54 // Function to get the full path of a file
55 export const getFilePath = (filename) => path.join(UPLOAD_DIR, filename);
56
57 // Function to check if a file exists
58 export const fileExists = async (filename) => {
59   try {
60     await fs.access(getFilePath(filename));
61     return true;
62   } catch {
63     return false;
64   }
65 };
```

A file explorer view of a project directory. The 'files' folder is selected and highlighted with a red rectangle. Inside the 'files' folder, 'JS filesConfig.js' is listed. Other files and folders visible include 'Task\_management', 'app', 'config', 'db', 'email', 'controllers', 'library', 'middleware', 'models', 'node\_modules', 'routes', 'uploads', 'views', '.env', 'package-lock.json', 'package.json', and 'server.js'.

```
Task_management
├── app
├── config
│   ├── db
│   └── email
├── files
│   └── JS filesConfig.js
├── controllers
├── library
├── middleware
├── models
├── node_modules
├── routes
├── uploads
├── views
├── .env
├── package-lock.json
├── package.json
└── JS server.js
```

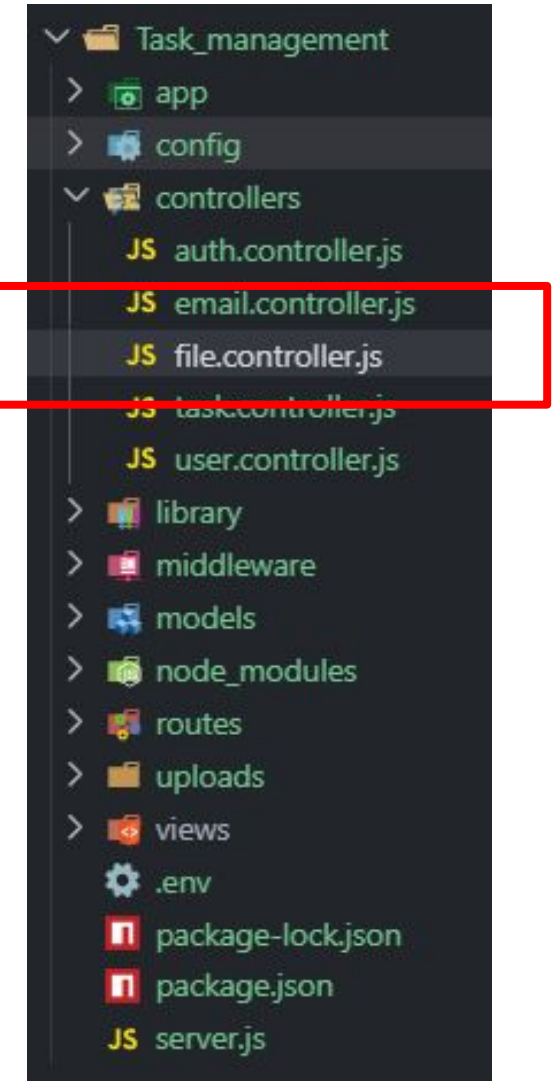
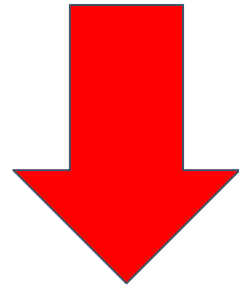
# Node.js - Files



Crear controlador

- controllers->files.controller.js

```
Task_management > controllers > JS file.controller.js > [icon] fileController > [icon] uploadFile > [icon] <function> > [icon] error
1  import path from 'path';
2  import { upload, getFilePath, fileExists, getFilesPath } from '../config/files/filesConfig.js';
3  import fs from 'fs';
4
```





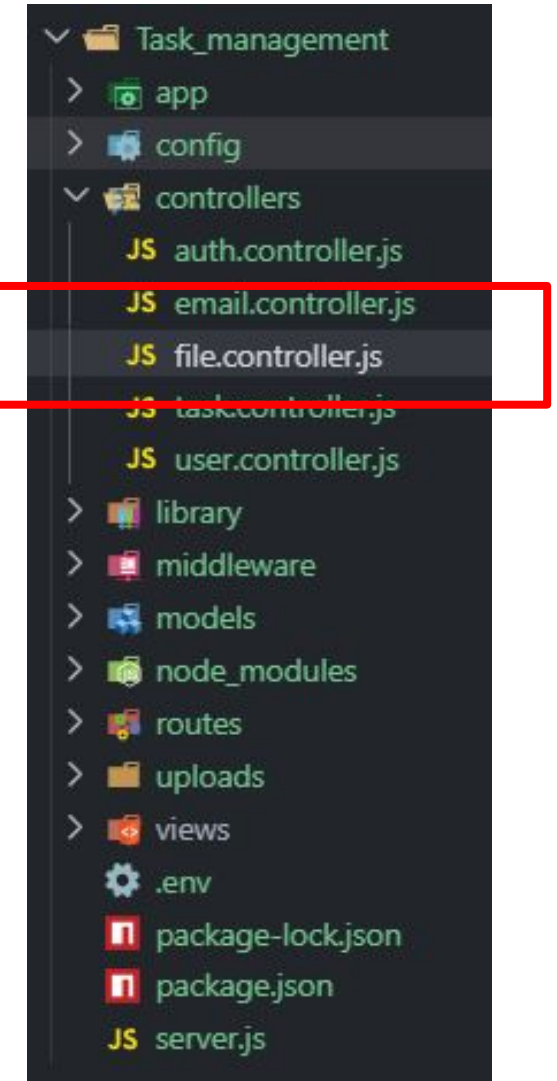
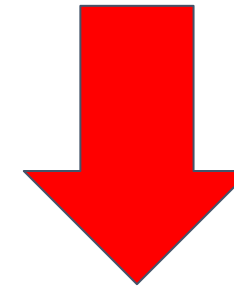
# Node.js - Files



## Crear controlador

- controllers->files.controller.js

```
5 export const fileController = {
6   /**
7    * uploadFile
8    */
9   uploadFile: [
10    upload.single('file'),
11    async (req, res) => {
12      try {
13        if (!req.file) {
14          return res.status(400).json({ error: 'Did not upload any files' });
15        }
16        res.status(201).json({
17          message: 'File uploaded successfully',
18          name: req.file.filename,
19          originalName: req.file.originalname,
20          size: req.file.size,
21          mimetype: req.file.mimetype,
22          url: getFilePath(req.file.filename),
23          extension: path.extname(req.file.filename).toLowerCase(),
24          uploadedAt: new Date()
25        });
26      } catch (error) {
27        console.error('Error uploading file :', error);
28        res.status(500).json({
29          error: error.message || 'Error uploading file.'
30        });
31      }
32    }
33  ],
34 }
```



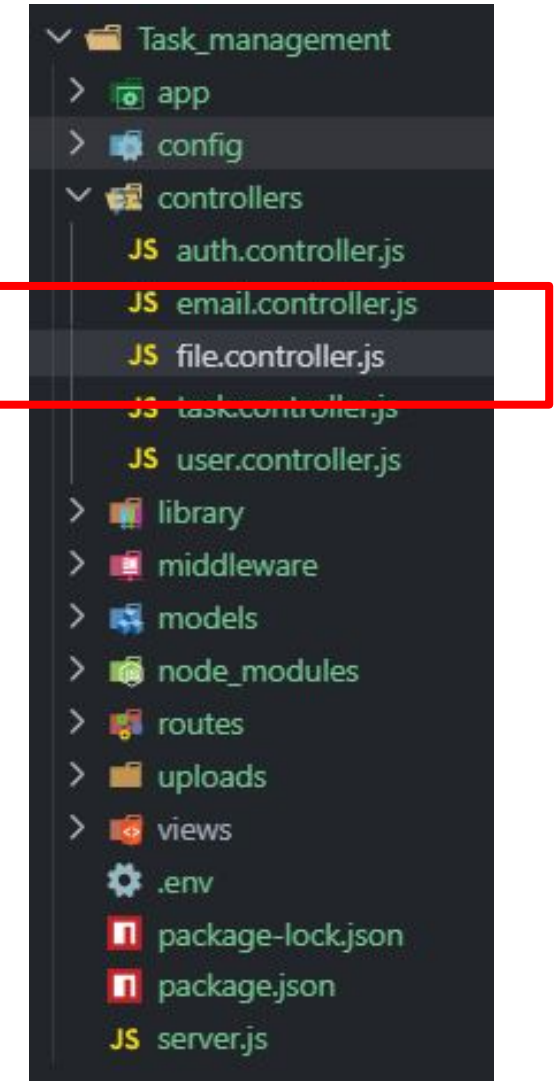
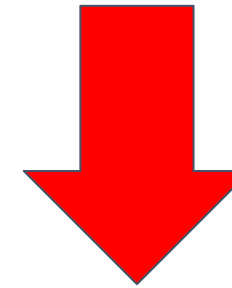
# Node.js - Files



## Crear controlador

- controllers->files.controller.js

```
38  getFiles: async (req, res) => {
39    try {
40      const filePath = getFilePath;
41      const fileStats = await fs.readdirSync(filePath);
42      if (fileStats.length === 0) {
43        return res.status(404).json({ error: 'No files found.' });
44      }
45      res.json({
46        files: fileStats.map(file => {
47          const filePath = getFilePath(file);
48          const stats = fs.statSync(filePath);
49          return {
50            name: file,
51            originalName: stats.originalName,
52            size: stats.size,
53            createdAt: stats.birthtime,
54            updatedAt: stats.mtime,
55            uploadedAt: stats.ctime,
56            mimetype: stats.mimetype || 'application/octet-stream', // Default type, can be improved
57            extension: path.extname(file).toLowerCase(),
58            url: filePath,
59          }
60        })
61      });
62    } catch (error) {
63      console.error('Error to search the file:', error);
64      res.status(500).json({ error: 'Error searching the file.' });
65    }
66  },
67 }
```



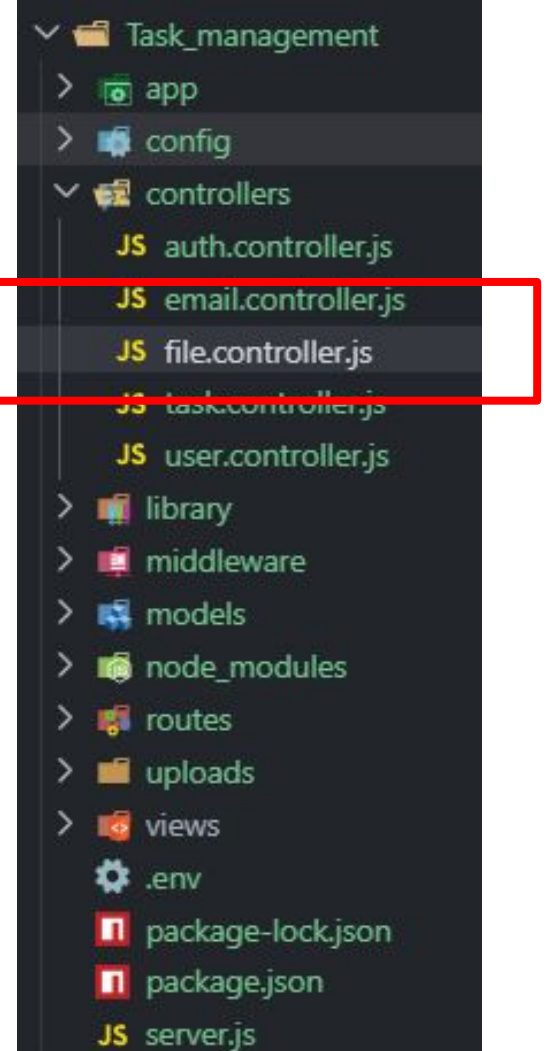
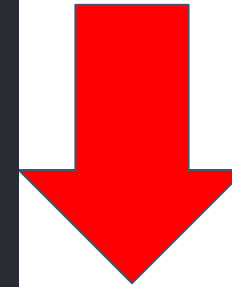
# Node.js - Files



## Crear controlador

- controllers->files.controller.js

```
69  /**
70   * Search for a file by filename and return its details
71   */
72  getFile: async (req, res) => {
73    try {
74      const { filename } = req.params;
75      if (!await fileExists(filename)) {
76        return res.status(404).json({ error: 'File not found' });
77      }
78      const filePath = getFilePath(filename);
79      const fileStats = await fs.statSync(filePath);
80      res.json({
81        name: filename,
82        url: filePath,
83        size: fileStats.size,
84        extension: path.extname(filename).toLowerCase(),
85        createdAt: fileStats.birthtime,
86        updatedAt: fileStats.mtime
87      });
88    } catch (error) {
89      console.error('Error al buscar archivo:', error);
90      res.status(500).json({ error: 'Error al buscar el archivo.' });
91    }
92  },
93
```





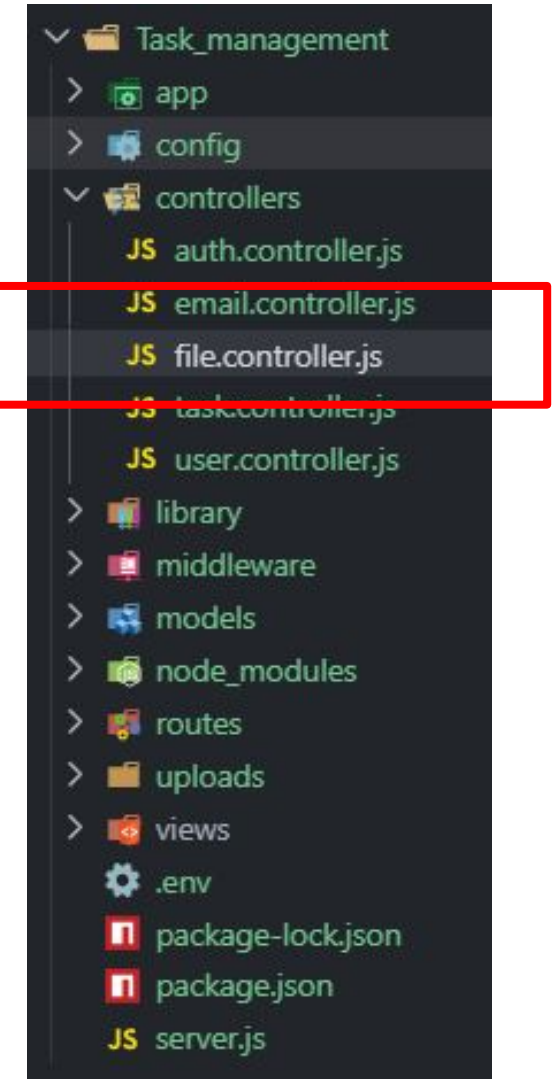
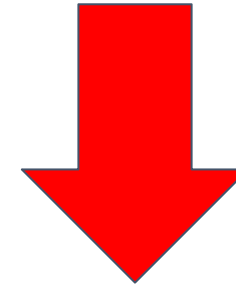
# Node.js - Files



## Crear controlador

- controllers->files.controller.js

```
94  /**
95   * Delete a file by filename
96   */
97  deleteFile: async (req, res) => {
98    try {
99      const { filename } = req.params;
100     if (!await fileExists(filename)) {
101       return res.status(404).json({ error: 'Archivo no encontrado.' });
102     }
103     const filePath = getFilePath(filename);
104     await fs.unlinkSync(filePath);
105     res.json({
106       message: 'delete file: ' + filePath + ' successfully',
107       filename
108     });
109   } catch (error) {
110     console.error('Error delete file:', error);
111     res.status(500).json({ error: 'Error delete file.' });
112   }
113 },
```



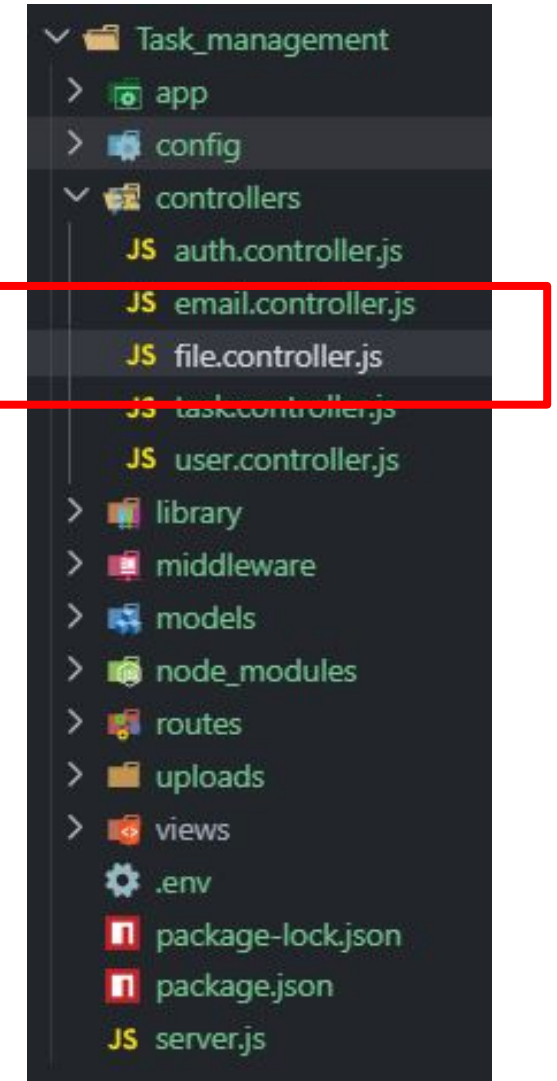
# Node.js - Files



## Crear controlador

- controllers->files.controller.js

```
115  /**
116   * Download a file by filename
117   */
118  downloadFile: async (req, res) => {
119    try {
120      const { filename } = req.params;
121      const filePath = getFilePath(filename);
122      if (!await fileExists(filename)) {
123        return res.status(404).json({ error: 'File not found.' });
124      }
125      res.download(filePath, filename, (err) => {
126        if (err) {
127          console.error('Error Downloading file:', err);
128          res.status(500).json({ error: 'Error downloading file.' });
129        }
130      });
131    } catch (error) {
132      console.error('Error download file:', error);
133      res.status(500).json({ error: 'Error downloading file.' });
134    }
135  }
```



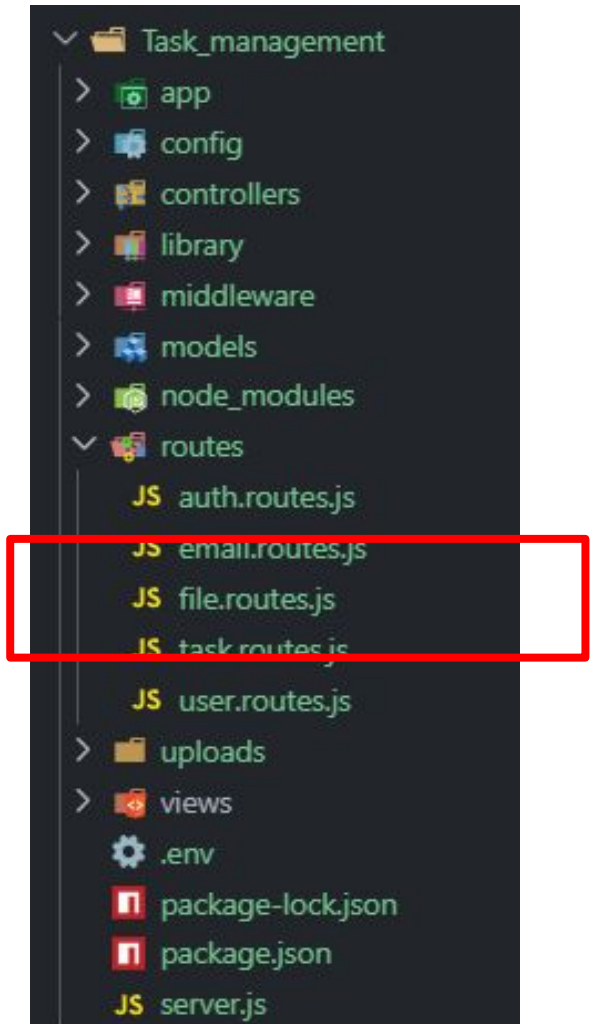
# Node.js - Files



## Crear las rutas

- routes->file.routes.js

```
Task_management > routes > JS file.routes.js > ...
1  import { Router } from "express";
2  import { fileController } from '../controllers/file.controller.js';
3  import { verifyToken } from '../middleware/authMiddleware.js';
4
5  const router = Router();
6  const name = '/files';
7
8  router.use(verifyToken);
9
10 router.route(`${name}/upload`)
11   .post(fileController.uploadFile) //upload file
12
13 router.route(`${name}`)
14   .get(fileController.GetFiles); //search files
15
16 router.route(`${name}/download/:filename`)
17   .get(fileController.downloadFile); // download file
18
19 router.route(`${name}/:filename`)
20   .get(fileController.getFile) // get file details for name
21   .delete(fileController.deleteFile); // delete file for name
22
23 export default router;
```



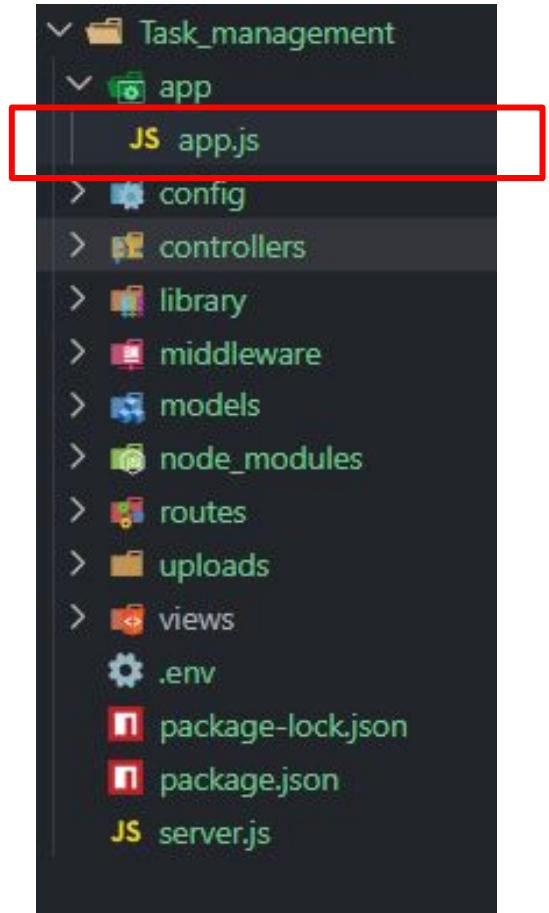
# Node.js - Files



## Agregar las rutas a la aplicación

- app->app.js

```
Task_management > app > JS app.js > app.use() callback
2 import { connectDB } from '../config/db/connection.js'; // Import the connection
3 // Import routes
4 import authRouter from '../routes/auth.routes.js';
5 import userRouter from '../routes/user.routes.js';
6 import taskRouter from '../routes/task.routes.js';
7 import emailRouter from '../routes/email.routes.js';
8 import filesRouter from '../routes/file.routes.js';
9
10 const app = express();
11 app.use(express.json());
12 app.use(express.urlencoded({ extended: true }));
13
14 // Connect to MongoDB
15 connectDB();// Call the connection function
16
17 // Use routes
18 app.use('/api', authRouter);
19 app.use('/api', userRouter);
20 app.use('/api', taskRouter);
21 app.use('/api', emailRouter);
22 app.use('/api', filesRouter);
23
24 app.use((req, res, next) => {
25   res.status(404).json({
26     message: 'Endpoint losses'
27   });
28 });
29
30 export default app;
```



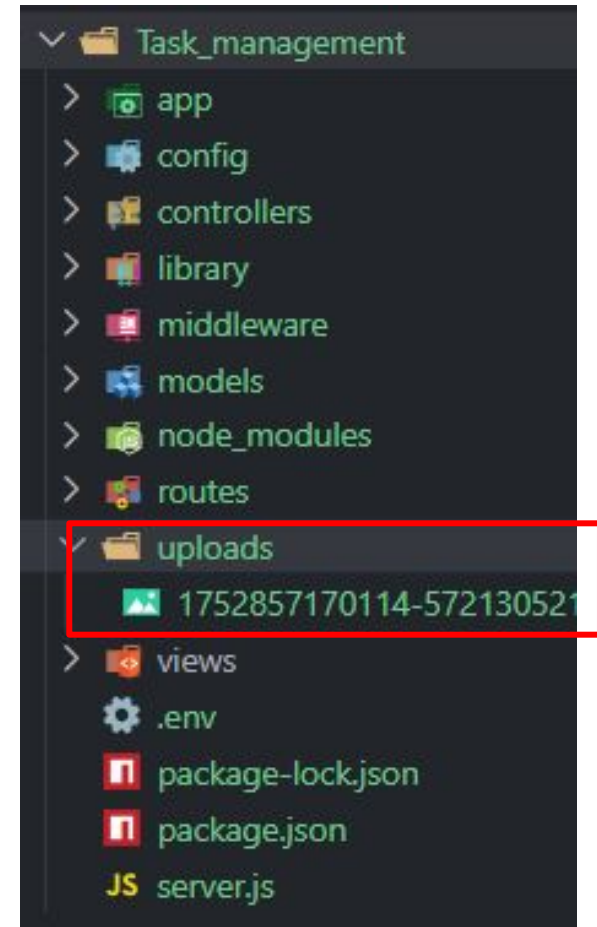


# Node.js - Files



Validar la carpeta

- uploads

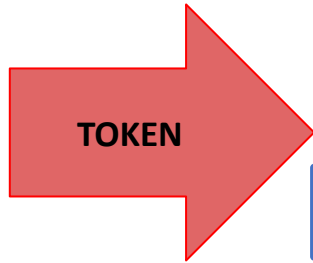




# Node.js - Files



## Pruebas



POST {{baseUrl}} /api/files/upload Send

Params Authorization Headers (11) Body Scripts Tests Settings Cookies

Auth Type Bearer Token Token

Body Cookies (1) Headers (7) Test Results 201 Created 32 ms 667 B Save Response

```
{
  "message": "File uploaded successfully",
  "name": "1752862277583-413348011-LogoAprobadoaCina22023-01blanco-01.png",
  "originalName": "Logo Aprobado aCina2 2023-01 blanco-01.png",
  "size": 68435,
  "mimetype": "image/png",
  "url": "C:\\xampp\\htdocs\\SENA\\SENA_ADSO\\BACK-END\\Node.js-Mongo\\Task_management\\uploads\\1752862277583-413348011-LogoAprobadoaCina22023-01blanco-01.png",
  "extension": ".png",
  "uploadedAt": "2025-07-18T18:11:17.592Z"
}
```

TASK-MONGO-NODE

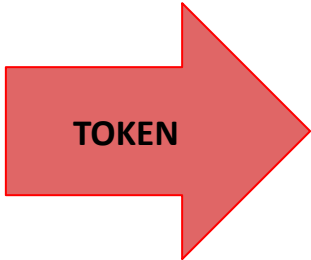
- > auth
- > user
- > task
- > email
- > files
  - POST upload
  - GET download
  - GET get File By Name
  - GET get Files
  - DEL delete File By Name

MÉTODO	RUTA	CONTROLADOR	MODELO
POST	http://localhost:3000/api/files/upload	Files	N/A

# Node.js - Files



## Pruebas



POST Send

Params Authorization Headers (11) Body Scripts Tests Settings Cookies

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key		Value	Description		Bulk Edit
<input checked="" type="checkbox"/>	file	File	Logo Aprobado aCina2 2023-01...			
	Key	Text	Value	Description		

- TASK-MONGO-NODE
  - auth
  - user
  - task
  - email
  - files
    - POST upload
    - GET download
    - GET get File By Name
    - GET get Files
    - DEL delete File By Name

MÉTODO	RUTA	CONTROLADOR	MODELO
POST	http://localhost:3000/api/files/upload	Files	N/A

# Node.js - Files



## Pruebas

POST

{{baseUrl}} /api/files/upload

Send

ParamsAuthorizationHeaders (11)BodyScriptsTestsSettings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Key	Value	Description
<input checked="" type="checkbox"/> file	File <div>Logo Aprobado aCina2 2023-01 bla...</div>	
Key	Text	Description

+ New file from local machine

BodyCookies (1)Headers (7)Test Results

{ } JSON

Preview

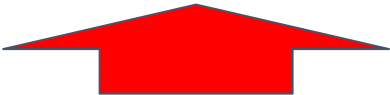
Visualize

Files uploaded to team32 ms • 667 B • Save Response

PERCEPCIÓN DE LA FORMACIÓN PROFE...

- TASK-MONGO-NODE
- > auth
  - > user
  - > task
  - > email
  - > files
    - POST upload
    - GET download
    - GET get File By Name
    - GET get Files
    - DEL delete File By Name

Cargar el archivo



MÉTODO	RUTA	CONTROLADOR	MODELO
POST	http://localhost:3000/api/files/upload	Files	N/A

# Node.js - Files



NOMBRE

## Pruebas

TOKEN

VISTA  
PREVIA

GET  Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

Auth Type  
Bearer Token

Token  
.....

The authorization header will be automatically generated when

Body Cookies (1) Headers (11) Test Results 200 OK • 18 ms • 67.23 KB • Save Response

Hex Preview Visualize

TASK-MONGO-NODE

- > auth
- > user
- > task
- > email
- > files
  - POST upload
  - GET download
  - GET get File By Name
  - GET get Files
  - DEL delete File By Name

MÉTODO	RUTA	CONTROLADOR	MODELO
GET	http://localhost:3000/api/files/download	Files	N/A

# Node.js - Files



NOMBRE

## Pruebas

TOKEN

RESPUESTA

GET `{{baseURL}} /api/files/1752857170114-572130521-LogoAprobadoaCina22023-01blanco-01.png` Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

Auth Type: Bearer Token

Token: `.....`

200 OK • 11 ms • 578 B • Save Response

```
{
  "name": "1752857170114-572130521-LogoAprobadoaCina22023-01blanco-01.png",
  "url": "C:\\xampp\\htdocs\\SENA\\SENA_ADSO\\BACK-END\\Node.js-Mongo\\Task_management\\uploads\\1752857170114-572130521-LogoAprobadoaCina22023-01blanco-01.png",
  "size": 68435,
  "extension": ".png",
  "createdAt": "2025-07-18T16:46:10.116Z",
  "updatedAt": "2025-07-18T16:46:10.121Z"
}
```

TASK-MONGO-NODE

- > auth
- > user
- > task
- > email
- > files
  - POST upload
  - GET download
  - GET get File By Name
  - GET get Files
  - DEL delete File By Name

MÉTODO	RUTA	CONTROLADOR	MODELO
GET	<code>http://localhost:3000/api/files/:filename</code>	Files	N/A



# Node.js - Files



## Pruebas

TOKEN

RESPUESTA

GET {{baseUrl}} /api/files/ Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

Auth Type Bearer Token Token

200 OK 15 ms 668 B Save Response

```
1 {
2   "files": [
3     {
4       "name": "1752857170114-572130521-LogoAprobadoaCina22023-01blanco-01.png",
5       "size": 68435,
6       "createdAt": "2025-07-18T16:46:10.116Z",
7       "updatedAt": "2025-07-18T16:46:10.121Z",
8       "uploadedAt": "2025-07-18T16:46:10.121Z",
9       "mimetype": "application/octet-stream",
10      "extension": ".png",
11      "url": "C:\\xampp\\htdocs\\SENA\\SENA_ADSO\\BACK-END\\Node.
js-Mongo\\Task_management\\uploads\\1752857170114-572130521-LogoAprobadoaCina22023-01blanco-01.png"
12    }
13  ]
14 }
```

### TASK-MONGO-NODE

- > auth
- > user
- > task
- > email
- > files
  - POST upload
  - GET download
  - GET get File By Name
  - GET get Files
  - DEL delete File By Name

MÉTODO	RUTA	CONTROLADOR	MODELO
GET	http://localhost:3000/api/files	Files	N/A

# Node.js - Files



## Pruebas

TOKEN

NOMBRE

RESPUESTA

**DELETE**  **Send**

Params **Authorization** Headers (9) Body Scripts Tests Settings Cookies

Auth Type  
Bearer Token

Token  
.....

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body Cookies (1) Headers (7) Test Results 200 OK • 16 ms • 502 B • Save Response

JSON Preview Visualize

```
1 {
2   "message": "delete file: C:\\xampp\\htdocs\\SENA\\SENA_ADSO\\BACK-END\\Node.
      js-Mongo\\Task_management\\uploads\\1752854758395-784242582-LogoAprobadoaCina22023-01blanco-01.
      png successfully",
3   "filename": "1752854758395-784242582-LogoAprobadoaCina22023-01blanco-01.png"
4 }
```

TASK-MONGO-NODE

- > auth
- > user
- > task
- > email
- > files
  - POST upload
  - GET download
  - GET get File By Name
  - GET get Files
  - DEL delete File By Name**

MÉTODO	RUTA	CONTROLADOR	MODELO
DELETE	http://localhost:3000/api/files/:filename	Files	N/A



**G R A C I A S**

Línea de atención al ciudadano: 01 8000 910270  
Línea de atención al empresario: 01 8000 910682



[www.sena.edu.co](http://www.sena.edu.co)