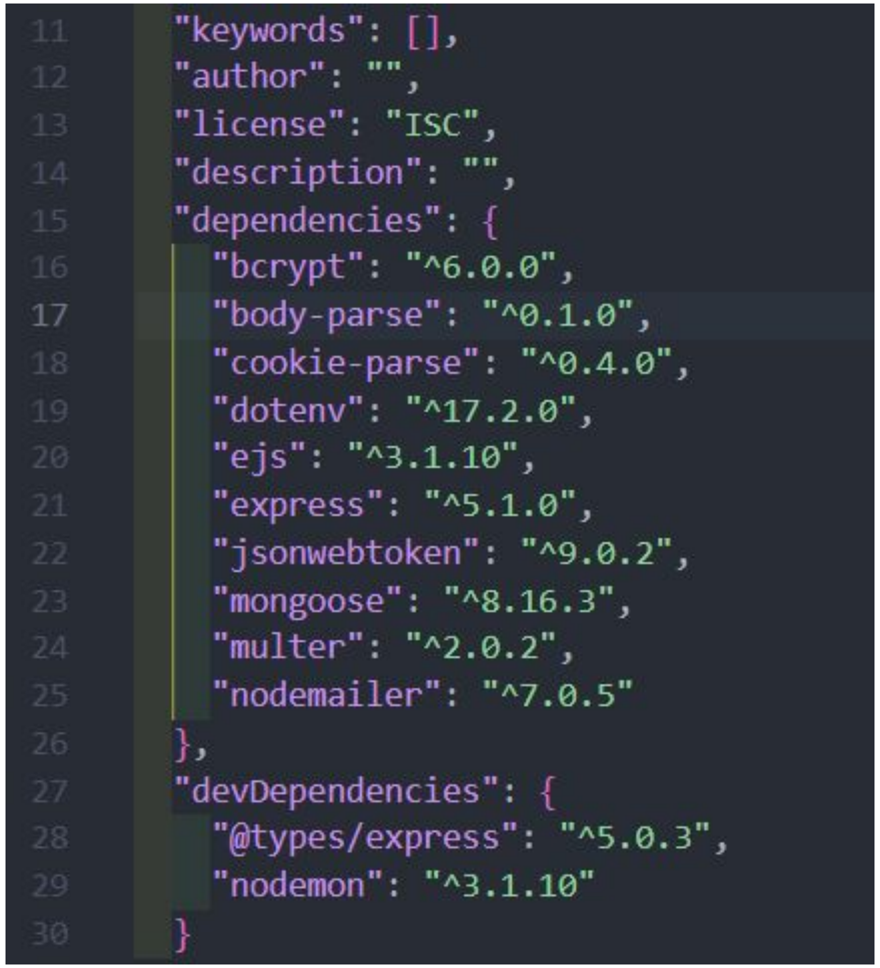
Curso complementario Desarrollo de Back-end con Node.js - MongoDB

22810019-1

Mongodb

Cargar archivos

- Validar las dependencias
  - npm install multer

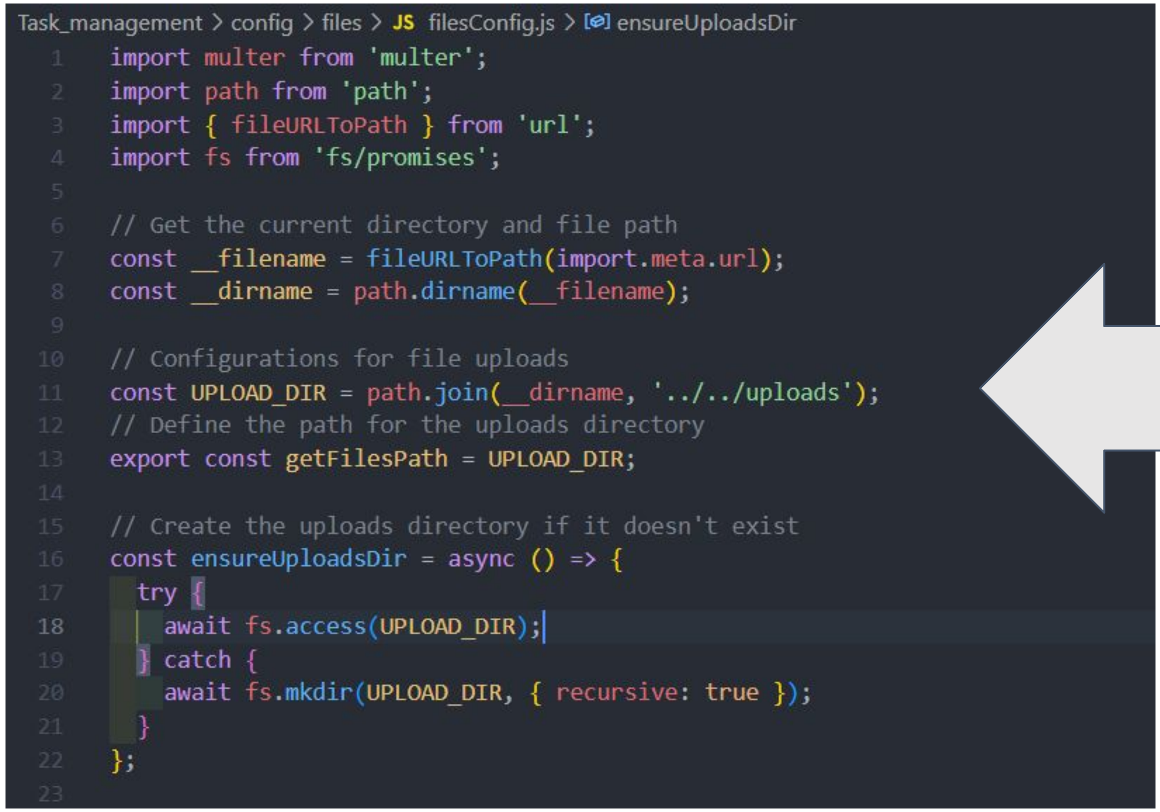


```
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "bcrypt": "^6.0.0",
    "body-parse": "^0.1.0",
    "cookie-parse": "^0.4.0",
    "dotenv": "^17.2.0",
    "ejs": "^3.1.10",
    "express": "^5.1.0",
    "jsonwebtoken": "^9.0.2",
    "mongoose": "^8.16.3",
    "multer": "^2.0.2",
    "nodemailer": "^7.0.5"
  },
  "devDependencies": {
    "@types/express": "^5.0.3",
    "nodemon": "^3.1.10"
  }
```

Mongodb

Crear la carpeta files

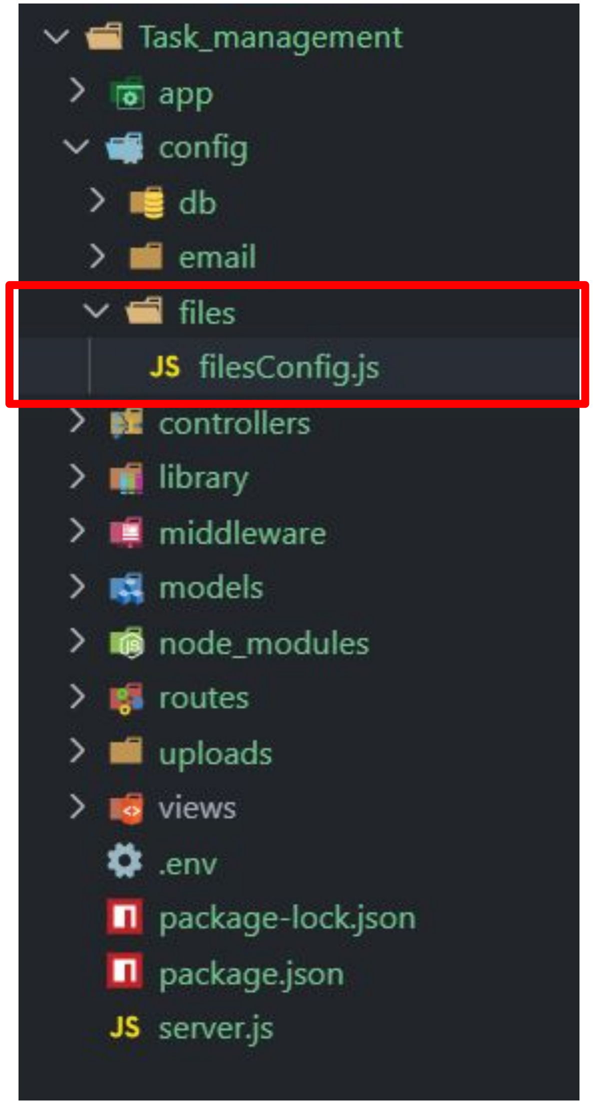
- config/files->filesConfig.js



Task_management > config > files > JS filesConfig.js > ensureUploadsDir

```javascript
import multer from 'multer';
import path from 'path';
import { fileURLToPath } from 'url';
import fs from 'fs/promises';

// Get the current directory and file path
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);

// Configurations for file uploads
const UPLOAD_DIR = path.join(__dirname, '../../uploads');
// Define the path for the uploads directory
export const getFilesPath = UPLOAD_DIR;

// Create the uploads directory if it doesn't exist
const ensureUploadsDir = async () => {
  try {
    await fs.access(UPLOAD_DIR);
  } catch {
    await fs.mkdir(UPLOAD_DIR, { recursive: true });
  }
};
```

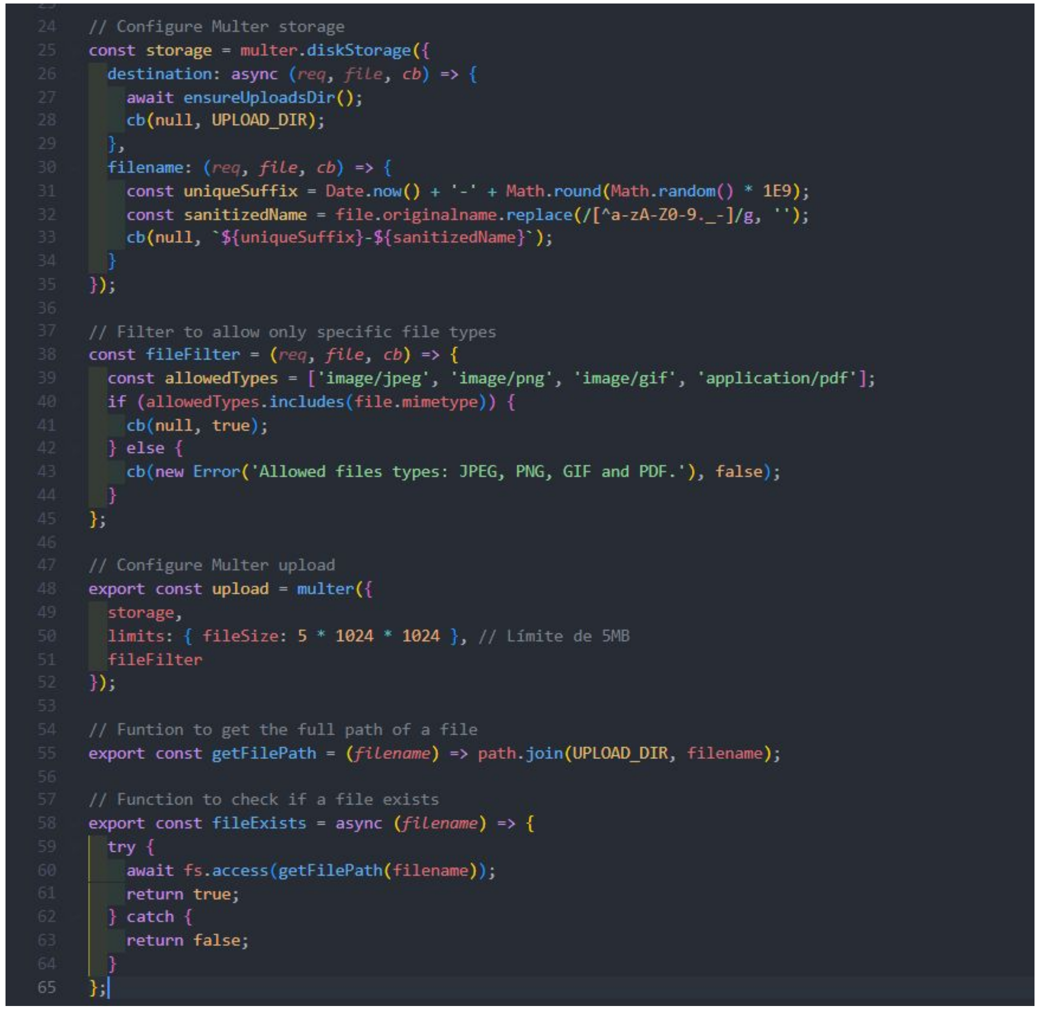
CARPETA



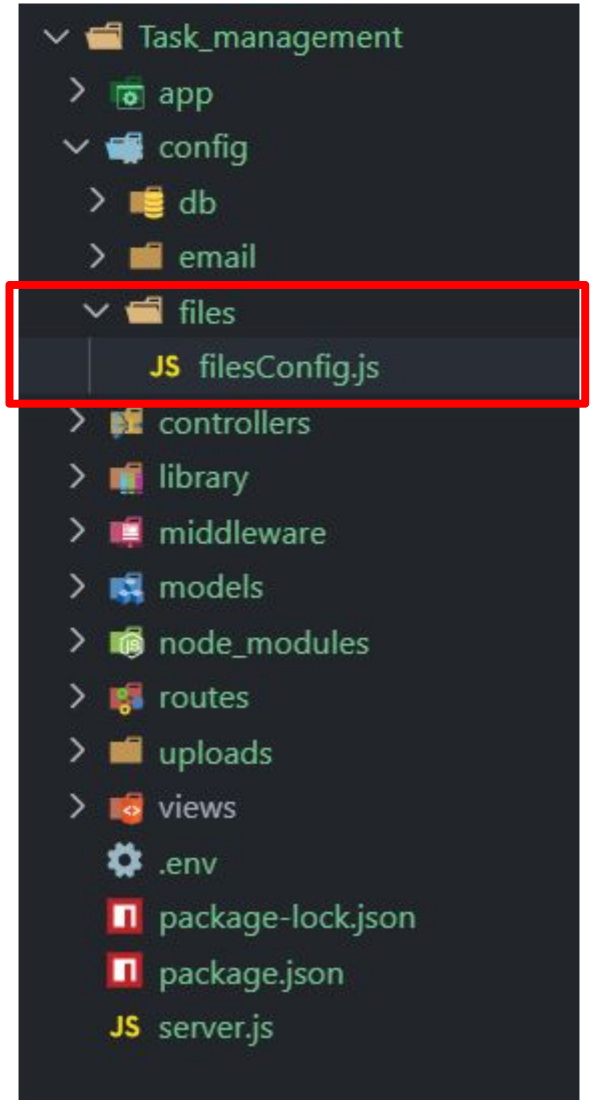
```
Task_management
  app
  config
    db
    email
    files
      filesConfig.js
  controllers
  library
  middleware
  models
  node_modules
  routes
  uploads
  views
  .env
  package-lock.json
  package.json
  server.js
```

Mongodb

- config/files->filesConfig.js



```javascript
// Configure Multer storage
const storage = multer.diskStorage({
  destination: async (req, file, cb) => {
    await ensureUploadsDir();
    cb(null, UPLOAD_DIR);
  },
  filename: (req, file, cb) => {
    const uniqueSuffix = Date.now() + '-' + Math.round(Math.random() * 1E9);
    const sanitizedName = file.originalname.replace(/[^a-zA-Z0-9._-]/g, '');
    cb(null, `${uniqueSuffix}-${sanitizedName}`);
  }
});

// Filter to allow only specific file types
const fileFilter = (req, file, cb) => {
  const allowedTypes = ['image/jpeg', 'image/png', 'image/gif', 'application/pdf'];
  if (allowedTypes.includes(file.mimetype)) {
    cb(null, true);
  } else {
    cb(new Error('Allowed files types: JPEG, PNG, GIF and PDF.'), false);
  }
};

// Configure Multer upload
export const upload = multer({
  storage,
  limits: { fileSize: 5 * 1024 * 1024 }, // Límite de 5MB
  fileFilter
});

// Funtion to get the full path of a file
export const getFilePath = (filename) => path.join(UPLOAD_DIR, filename);

// Function to check if a file exists
export const fileExists = async (filename) => {
  try {
    await fs.access(getFilePath(filename));
    return true;
  } catch {
    return false;
  }
};
```

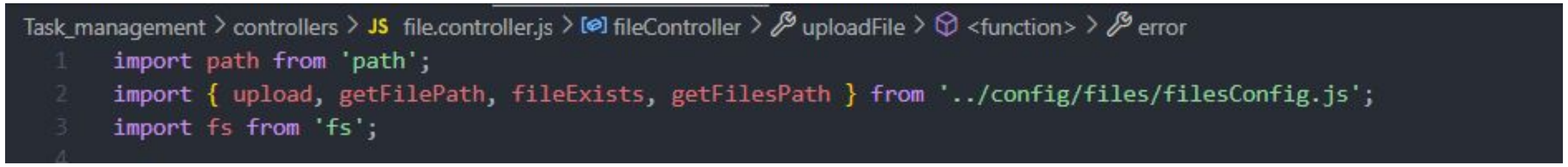


```
Task_management
  app
  config
    db
    email
    files
      filesConfig.js
  controllers
  library
  middleware
  models
  node_modules
  routes
  uploads
  views
  .env
  package-lock.json
  package.json
  server.js
```

Mongodb

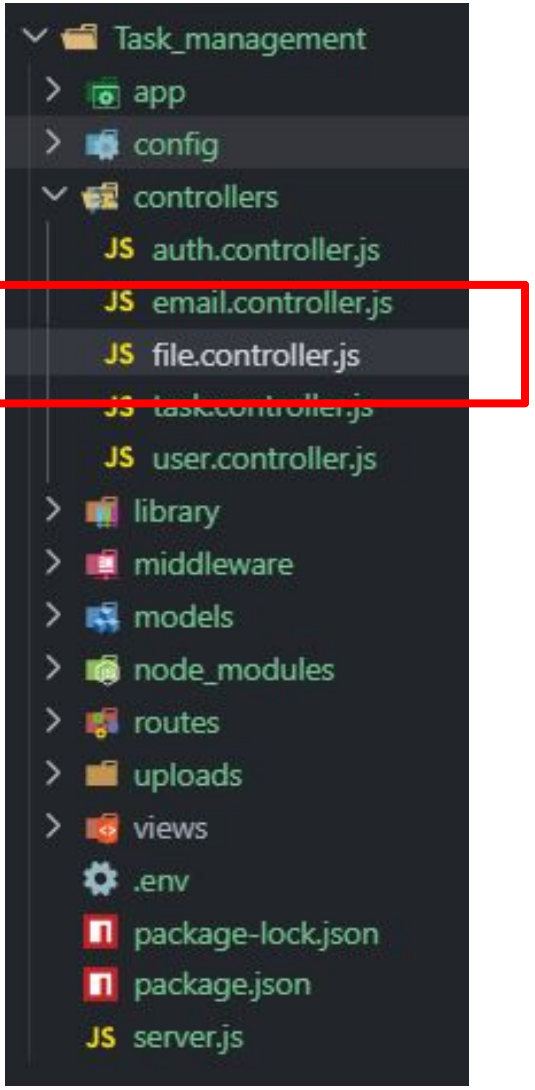
Crear controlador

- controllers->files.controller.js



Task_management > controllers > JS file.controller.js > fileController > uploadFile > <function> > error

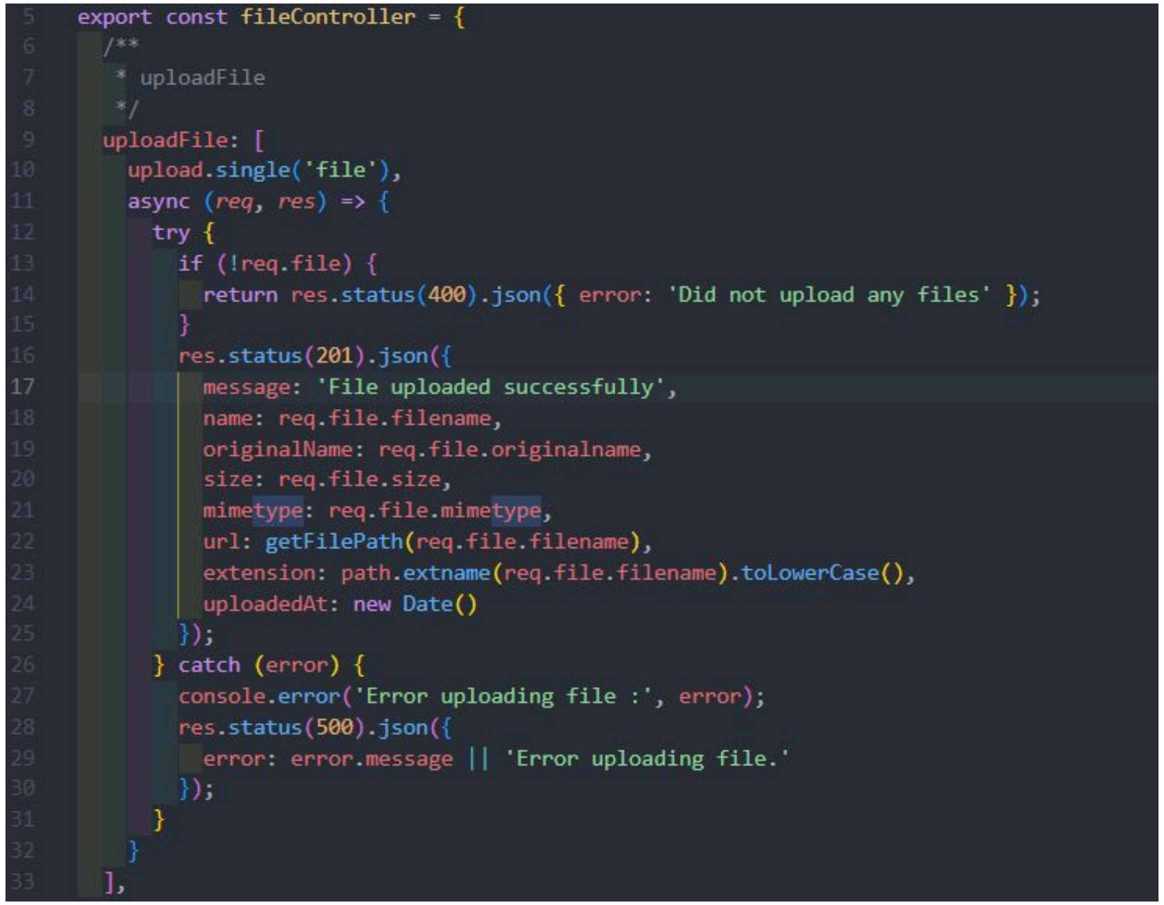
```
import path from 'path';
import { upload, getFilePath, fileExists, getFilesPath } from '../config/files/filesConfig.js';
import fs from 'fs';
```



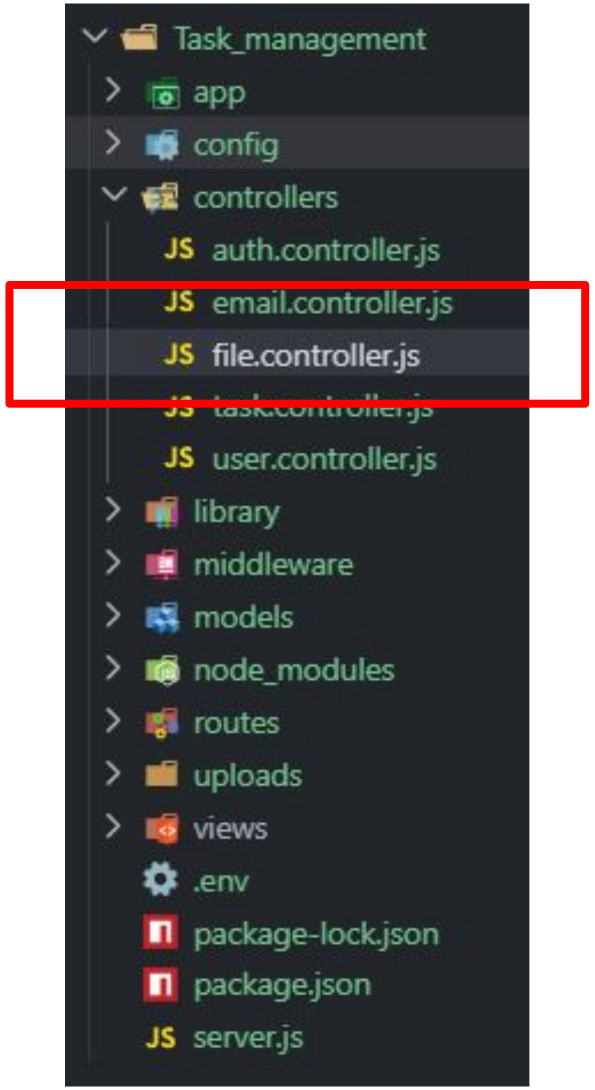
Mongodb

Crear controlador

- controllers->files.controller.js



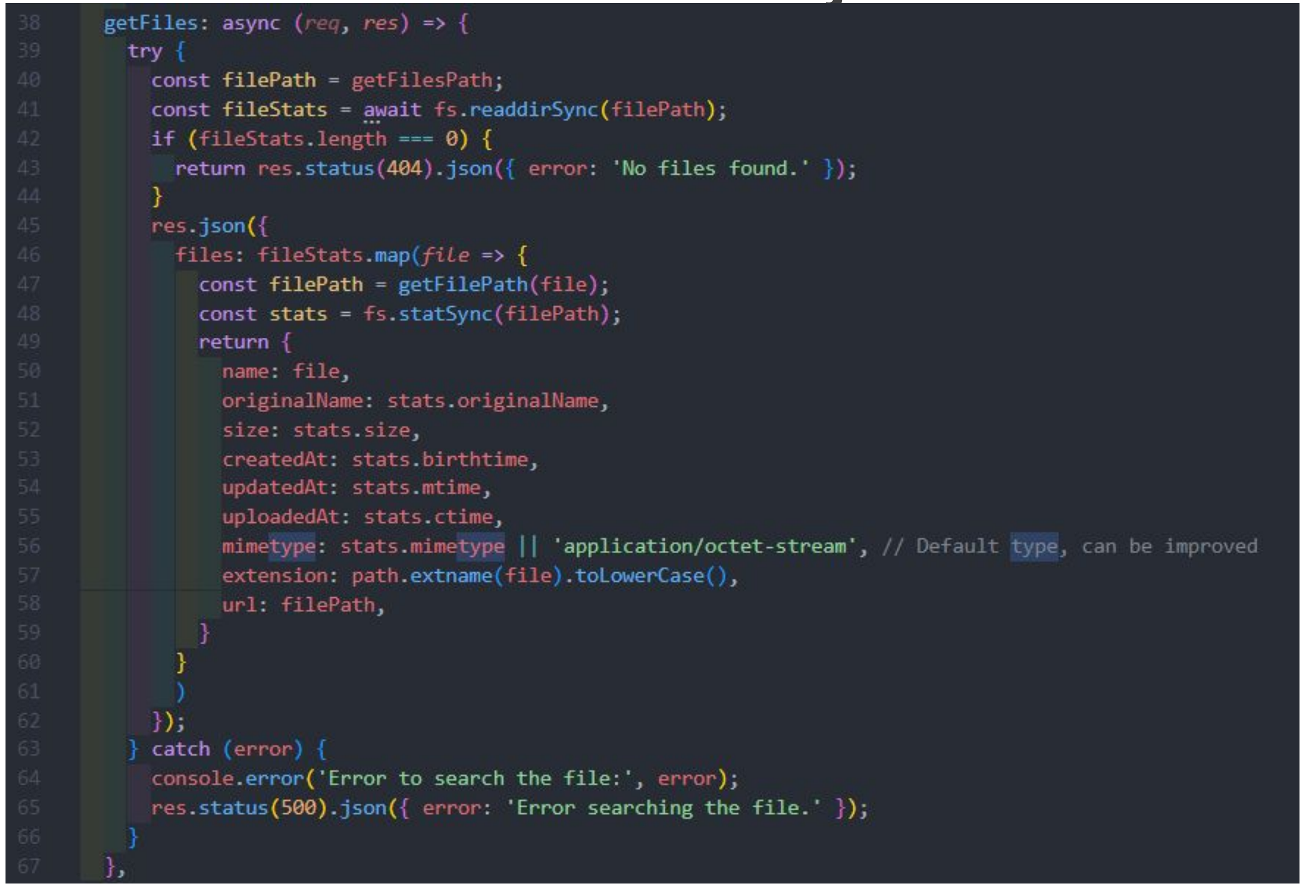
```
export const fileController = {
  /**
   * uploadFile
   */
  uploadFile: [
    upload.single('file'),
    async (req, res) => {
      try {
        if (!req.file) {
          return res.status(400).json({ error: 'Did not upload any files' });
        }
        res.status(201).json({
          message: 'File uploaded successfully',
          name: req.file.filename,
          originalName: req.file.originalname,
          size: req.file.size,
          mimetype: req.file.mimetype,
          url: getFilePath(req.file.filename),
          extension: path.extname(req.file.filename).toLowerCase(),
          uploadedAt: new Date()
        });
      } catch (error) {
        console.error('Error uploading file :', error);
        res.status(500).json({
          error: error.message || 'Error uploading file.'
        });
      }
    }
  ],
```



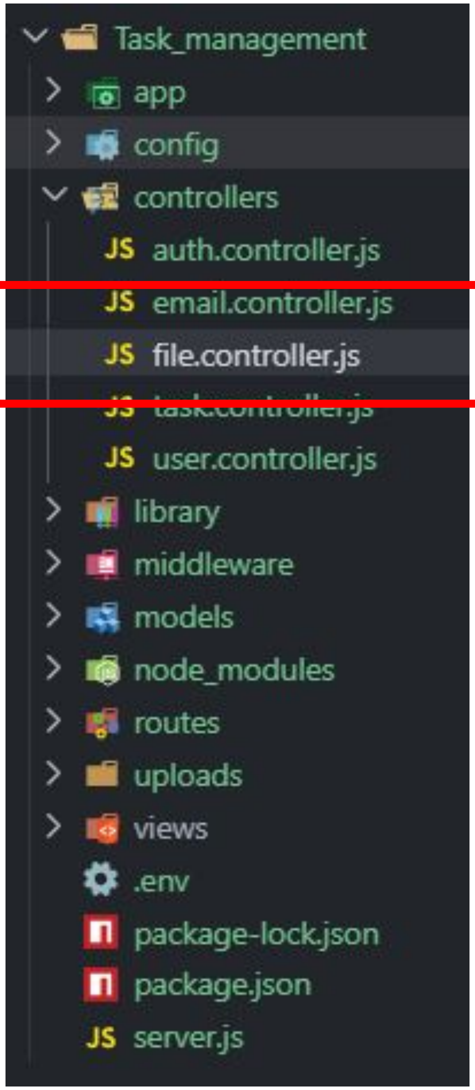
Mongodb

Crear controlador

- controllers->files.controller.js



```javascript
getFiles: async (req, res) => {
  try {
    const filePath = getFilesPath;
    const fileStats = await fs.readdirSync(filePath);
    if (fileStats.length === 0) {
      return res.status(404).json({ error: 'No files found.' });
    }
    res.json({
      files: fileStats.map(file => {
        const filePath = getFilePath(file);
        const stats = fs.statSync(filePath);
        return {
          name: file,
          originalName: stats.originalName,
          size: stats.size,
          createdAt: stats.birthtime,
          updatedAt: stats.mtime,
          uploadedAt: stats.ctime,
          mimetype: stats.mimetype || 'application/octet-stream', // Default type, can be improved
          extension: path.extname(file).toLowerCase(),
          url: filePath,
        }
      }
      )
    });
  } catch (error) {
    console.error('Error to search the file:', error);
    res.status(500).json({ error: 'Error searching the file.' });
  }
},
```

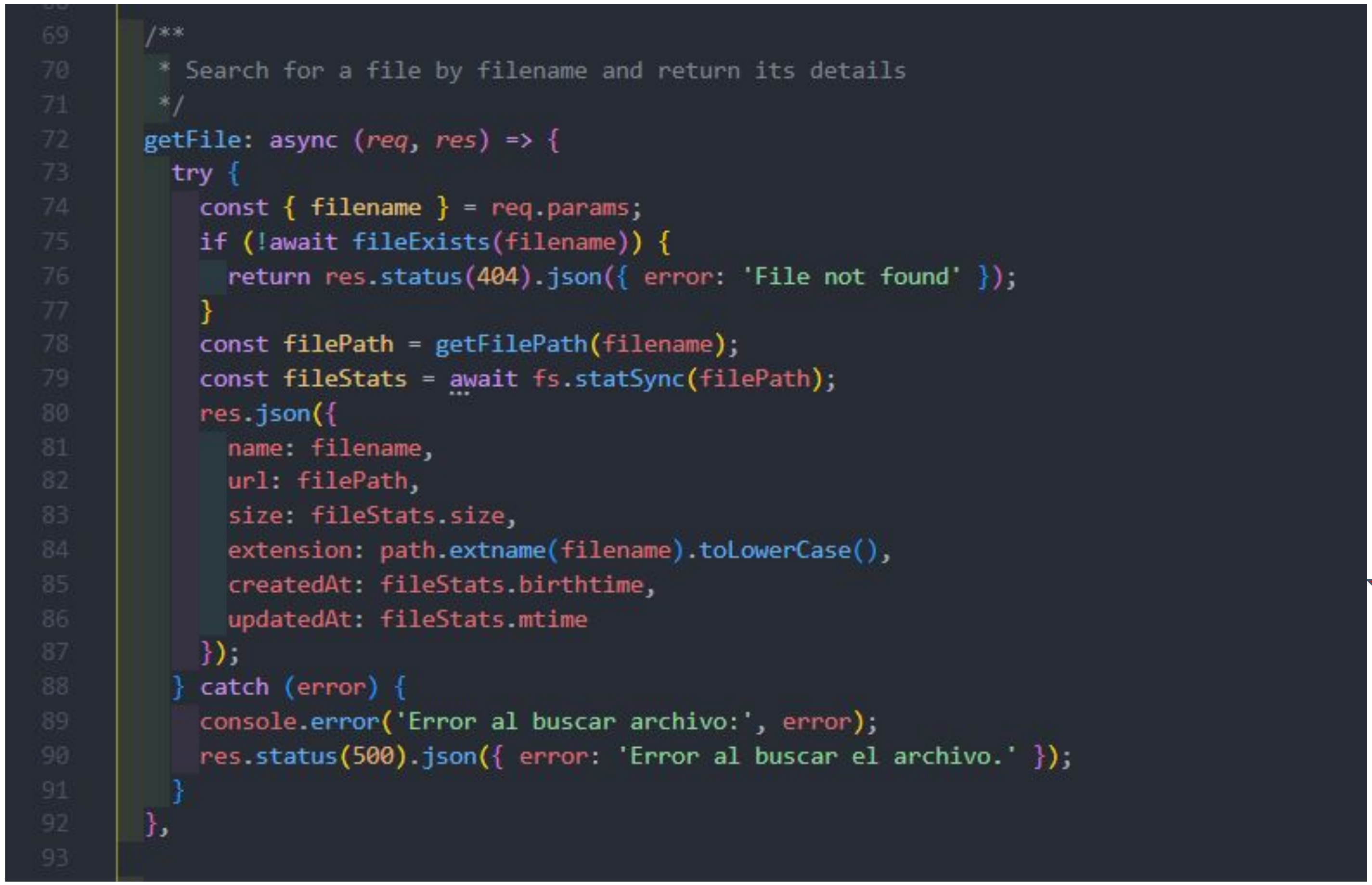


```
Task_management
  app
  config
  controllers
    auth.controller.js
    email.controller.js
    file.controller.js
    task.controller.js
    user.controller.js
  library
  middleware
  models
  node_modules
  routes
  uploads
  views
  .env
  package-lock.json
  package.json
  server.js
```

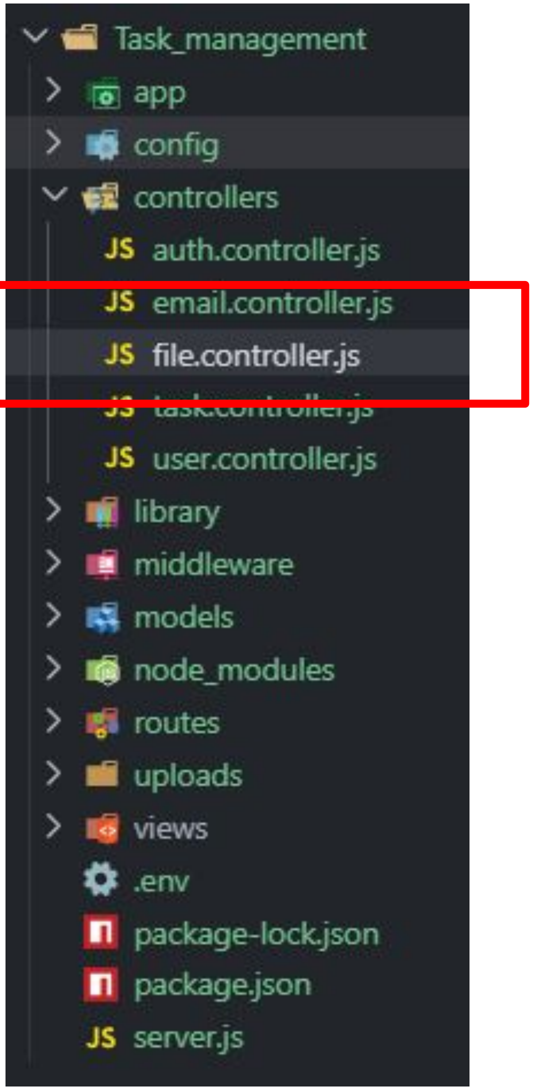
Mongodb

Crear controlador

- controllers->files.controller.js



```
/**
 * Search for a file by filename and return its details
 */
getFile: async (req, res) => {
  try {
    const { filename } = req.params;
    if (!await fileExists(filename)) {
      return res.status(404).json({ error: 'File not found' });
    }
    const filePath = getFilePath(filename);
    const fileStats = await fs.statSync(filePath);
    res.json({
      name: filename,
      url: filePath,
      size: fileStats.size,
      extension: path.extname(filename).toLowerCase(),
      createdAt: fileStats.birthtime,
      updatedAt: fileStats.mtime
    });
  } catch (error) {
    console.error('Error al buscar archivo:', error);
    res.status(500).json({ error: 'Error al buscar el archivo.' });
  }
},
```

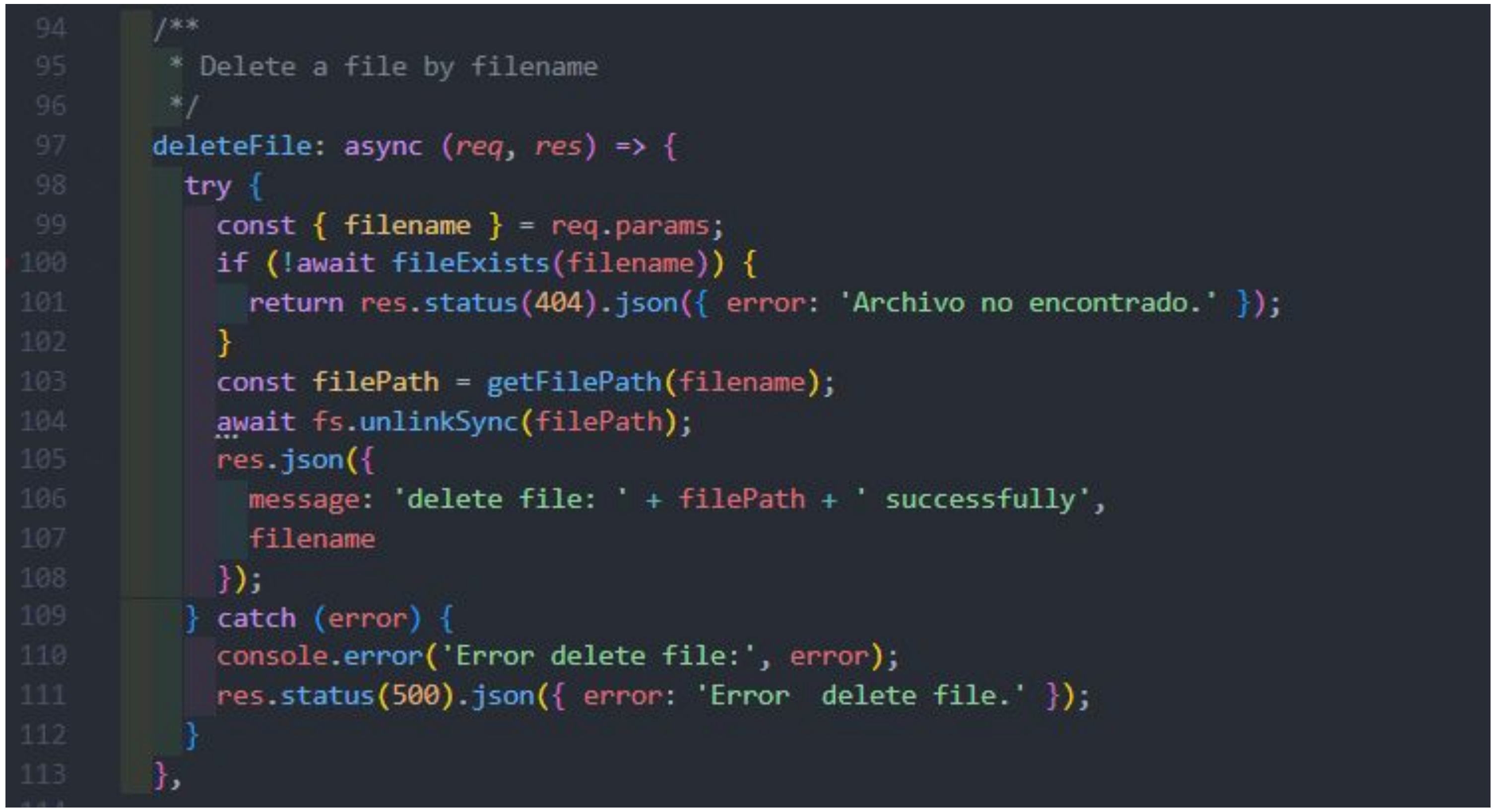


```
Task_management
  app
  config
  controllers
    auth.controller.js
    email.controller.js
    file.controller.js
    task.controller.js
    user.controller.js
  library
  middleware
  models
  node_modules
  routes
  uploads
  views
  .env
  package-lock.json
  package.json
  server.js
```

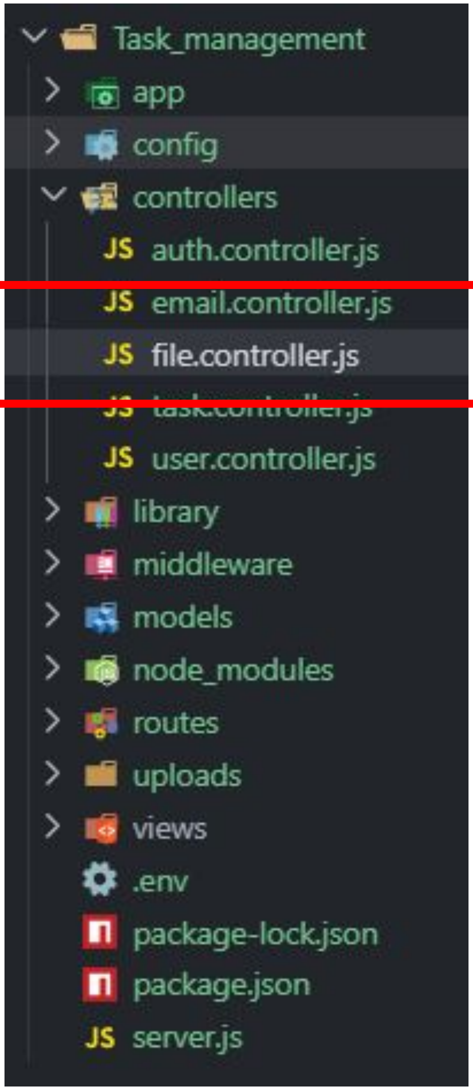
Mongodb

Crear controlador

- controllers->files.controller.js



```
/**
 * Delete a file by filename
 */
deleteFile: async (req, res) => {
  try {
    const { filename } = req.params;
    if (!await fileExists(filename)) {
      return res.status(404).json({ error: 'Archivo no encontrado.' });
    }
    const filePath = getFilePath(filename);
    await fs.unlinkSync(filePath);
    res.json({
      message: 'delete file: ' + filePath + ' successfully',
      filename
    });
  } catch (error) {
    console.error('Error delete file:', error);
    res.status(500).json({ error: 'Error  delete file.' });
  }
},
```

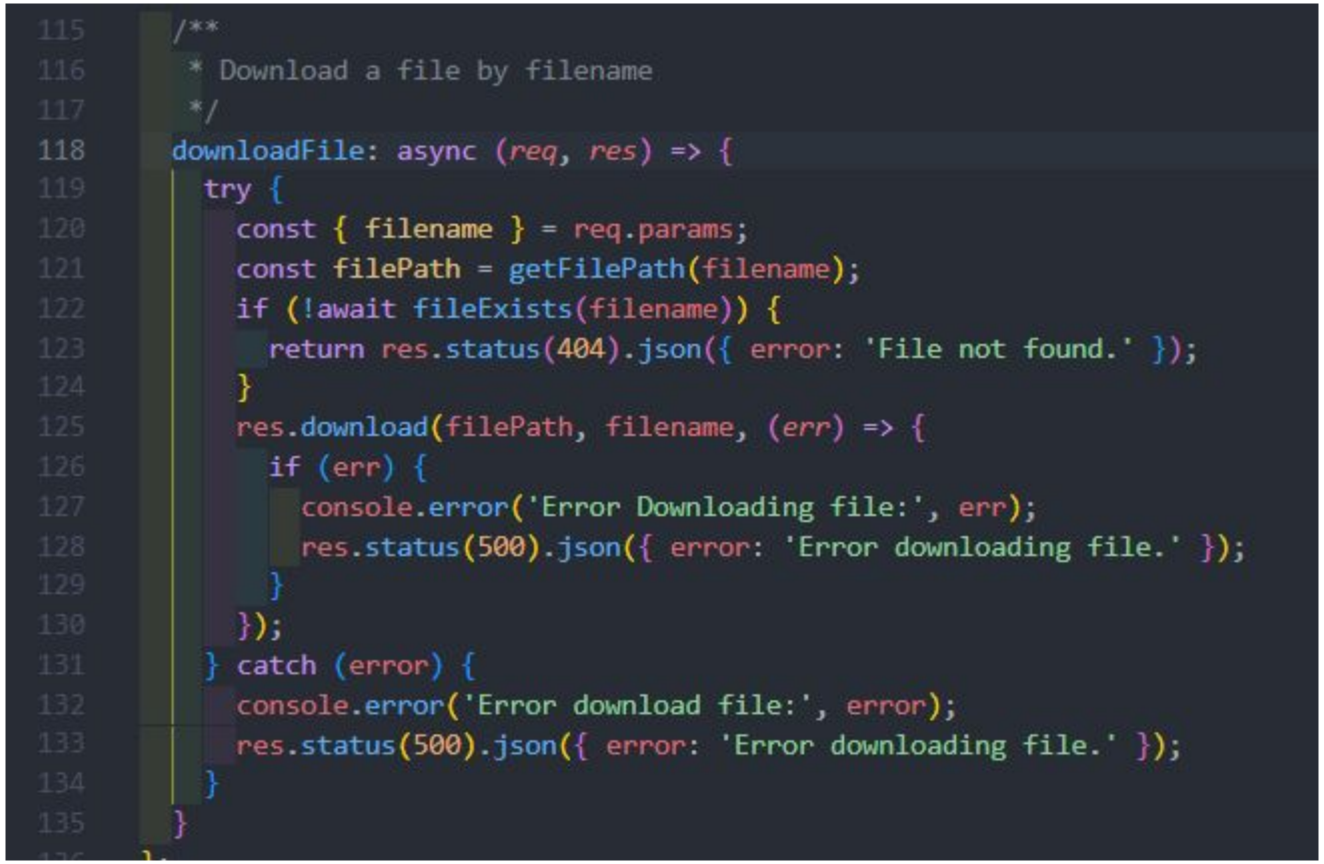


```
Task_management
  app
  config
  controllers
    auth.controller.js
    email.controller.js
    file.controller.js
    task.controller.js
    user.controller.js
  library
  middleware
  models
  node_modules
  routes
  uploads
  views
  .env
  package-lock.json
  package.json
  server.js
```

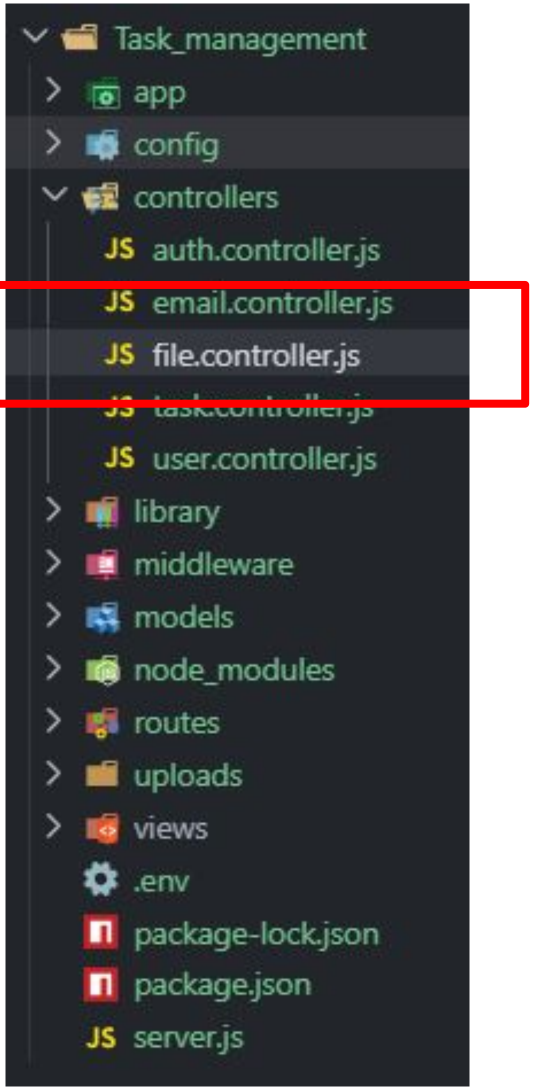
Mongodb

Crear controlador

- controllers->files.controller.js



```
/**
 * Download a file by filename
 */
downloadFile: async (req, res) => {
  try {
    const { filename } = req.params;
    const filePath = getFilePath(filename);
    if (!await fileExists(filename)) {
      return res.status(404).json({ error: 'File not found.' });
    }
    res.download(filePath, filename, (err) => {
      if (err) {
        console.error('Error Downloading file:', err);
        res.status(500).json({ error: 'Error downloading file.' });
      }
    });
  } catch (error) {
    console.error('Error download file:', error);
    res.status(500).json({ error: 'Error downloading file.' });
  }
}
```

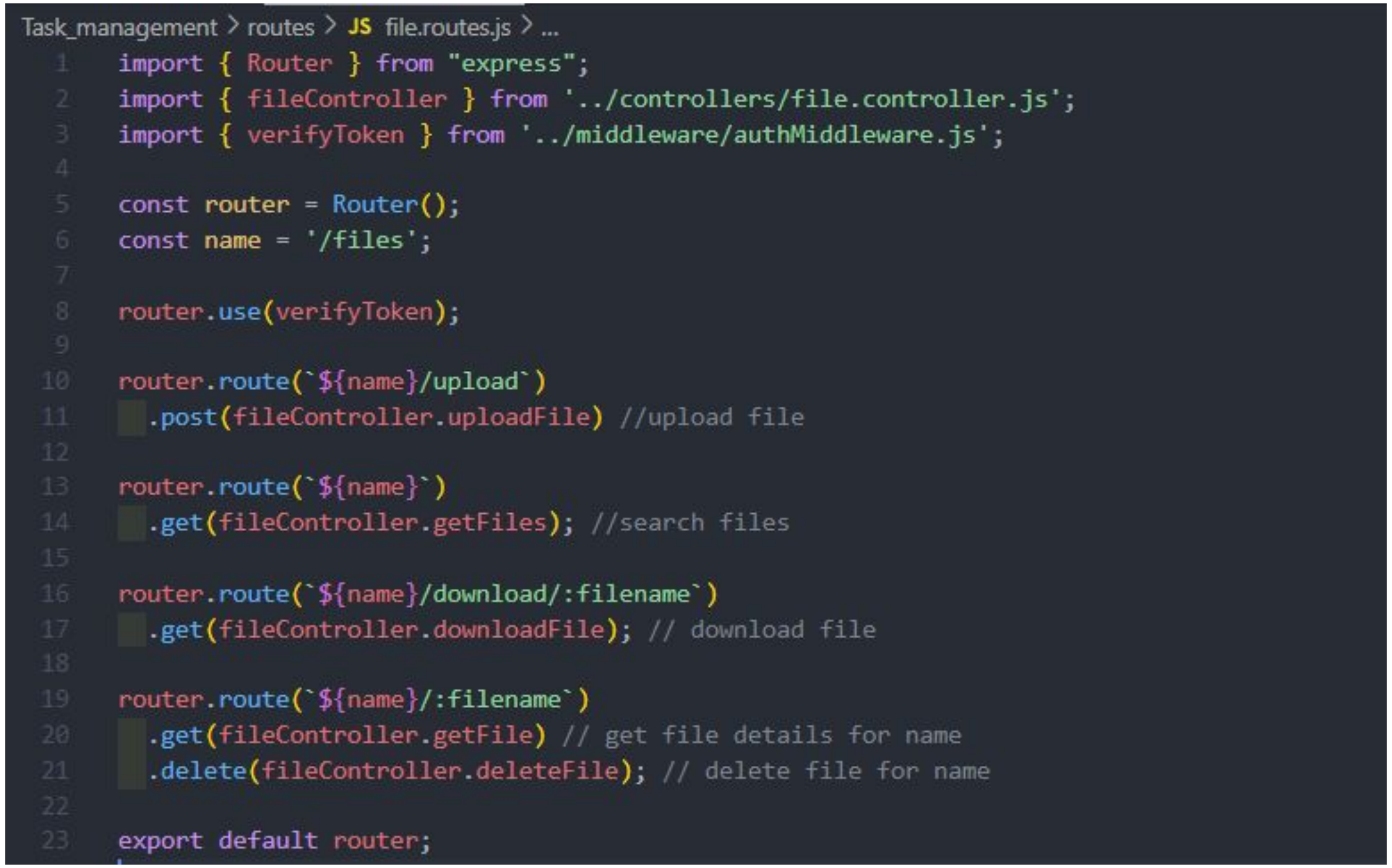


```
Task_management
  app
  config
  controllers
    auth.controller.js
    email.controller.js
    file.controller.js
    task.controller.js
    user.controller.js
  library
  middleware
  models
  node_modules
  routes
  uploads
  views
  .env
  package-lock.json
  package.json
  server.js
```

Mongodb

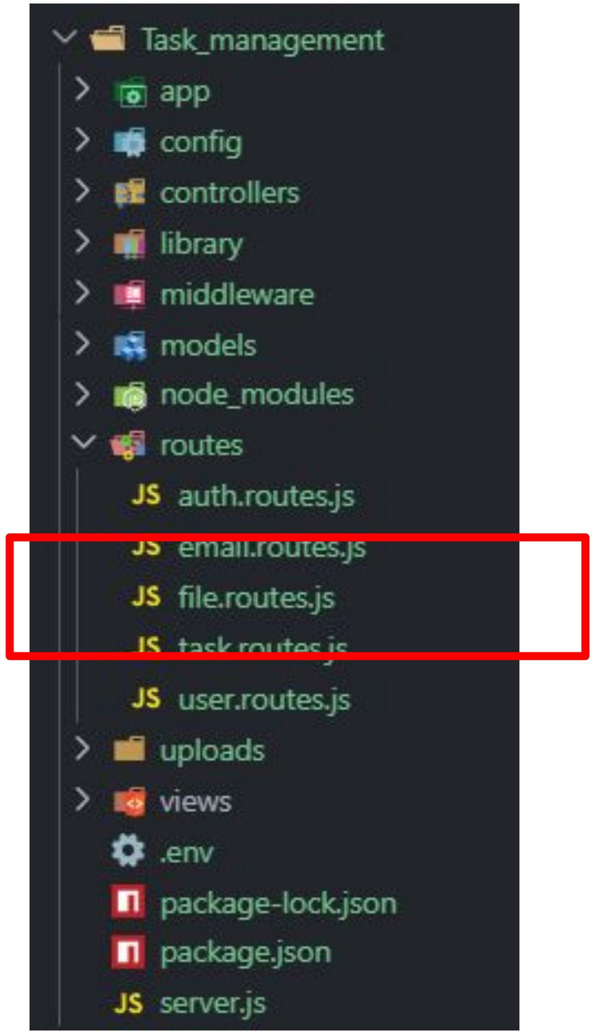
Crear las rutas

- routes->file.routes.js



Task_management > routes > JS file.routes.js > ...

```javascript
import { Router } from "express";
import { fileController } from '../controllers/file.controller.js';
import { verifyToken } from '../middleware/authMiddleware.js';

const router = Router();
const name = '/files';

router.use(verifyToken);

router.route(`${name}/upload`)
    .post(fileController.uploadFile) //upload file

router.route(`${name}`)
    .get(fileController.getFiles); //search files

router.route(`${name}/download/:filename`)
    .get(fileController.downloadFile); // download file

router.route(`${name}/:filename`)
    .get(fileController.getFile) // get file details for name
    .delete(fileController.deleteFile); // delete file for name

export default router;
```

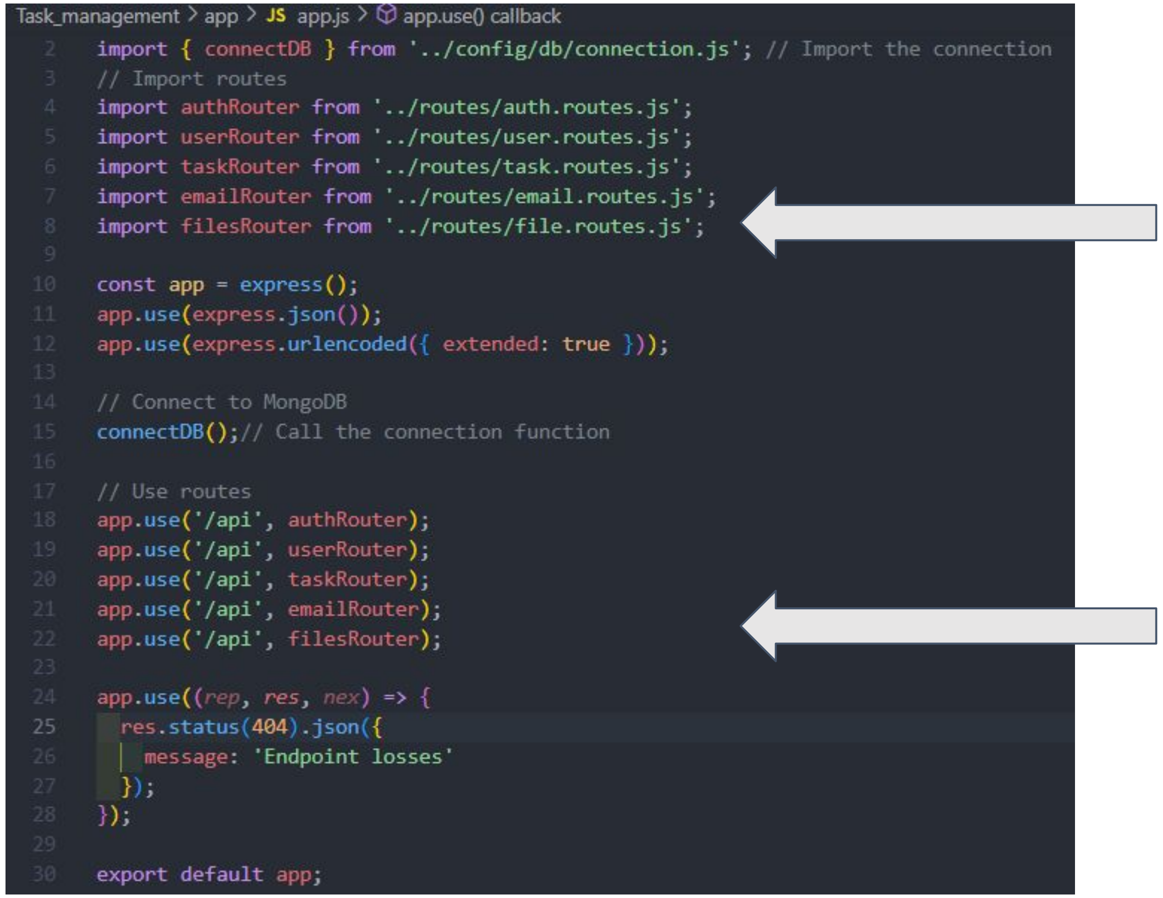


```
Task_management
  app
  config
  controllers
  library
  middleware
  models
  node_modules
  routes
    auth.routes.js
    email.routes.js
    file.routes.js
    task.routes.js
    user.routes.js
  uploads
  views
  .env
  package-lock.json
  package.json
  server.js
```

Mongodb

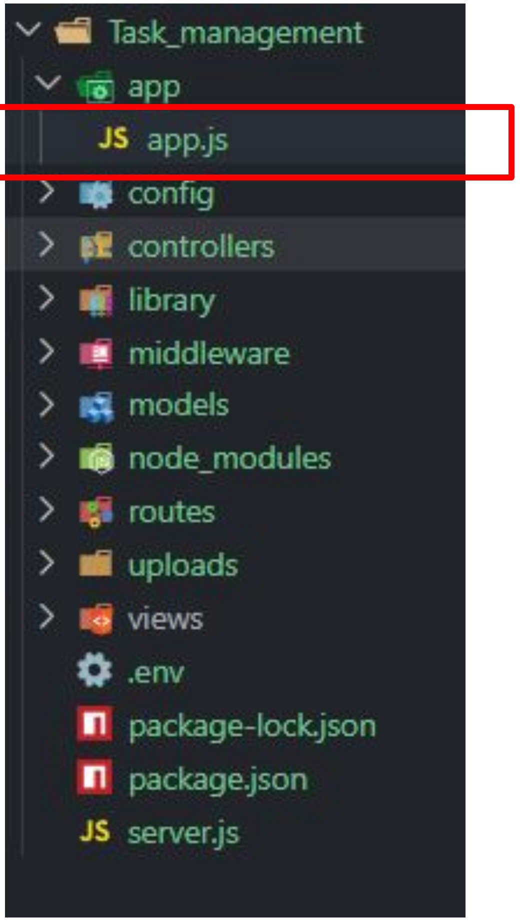
Agregar las rutas a la aplicación

- **app->app.js**



Task_management > app > JS app.js > app.use() callback

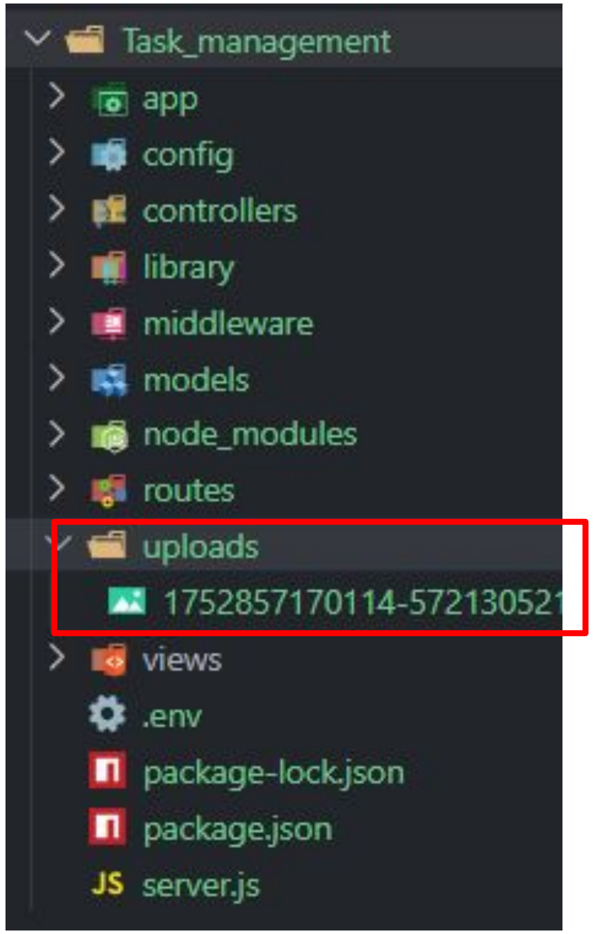
```
import { connectDB } from '../config/db/connection.js'; // Import the connection
// Import routes
import authRouter from '../routes/auth.routes.js';
import userRouter from '../routes/user.routes.js';
import taskRouter from '../routes/task.routes.js';
import emailRouter from '../routes/email.routes.js';
import filesRouter from '../routes/file.routes.js';

const app = express();
app.use(express.json());
app.use(express.urlencoded({ extended: true }));

// Connect to MongoDB
connectDB();// Call the connection function

// Use routes
app.use('/api', authRouter);
app.use('/api', userRouter);
app.use('/api', taskRouter);
app.use('/api', emailRouter);
app.use('/api', filesRouter);

app.use((rep, res, nex) => {
  res.status(404).json({
    message: 'Endpoint losses'
  });
});

export default app;
```



Mongodb

Validar la carpeta

- uploads

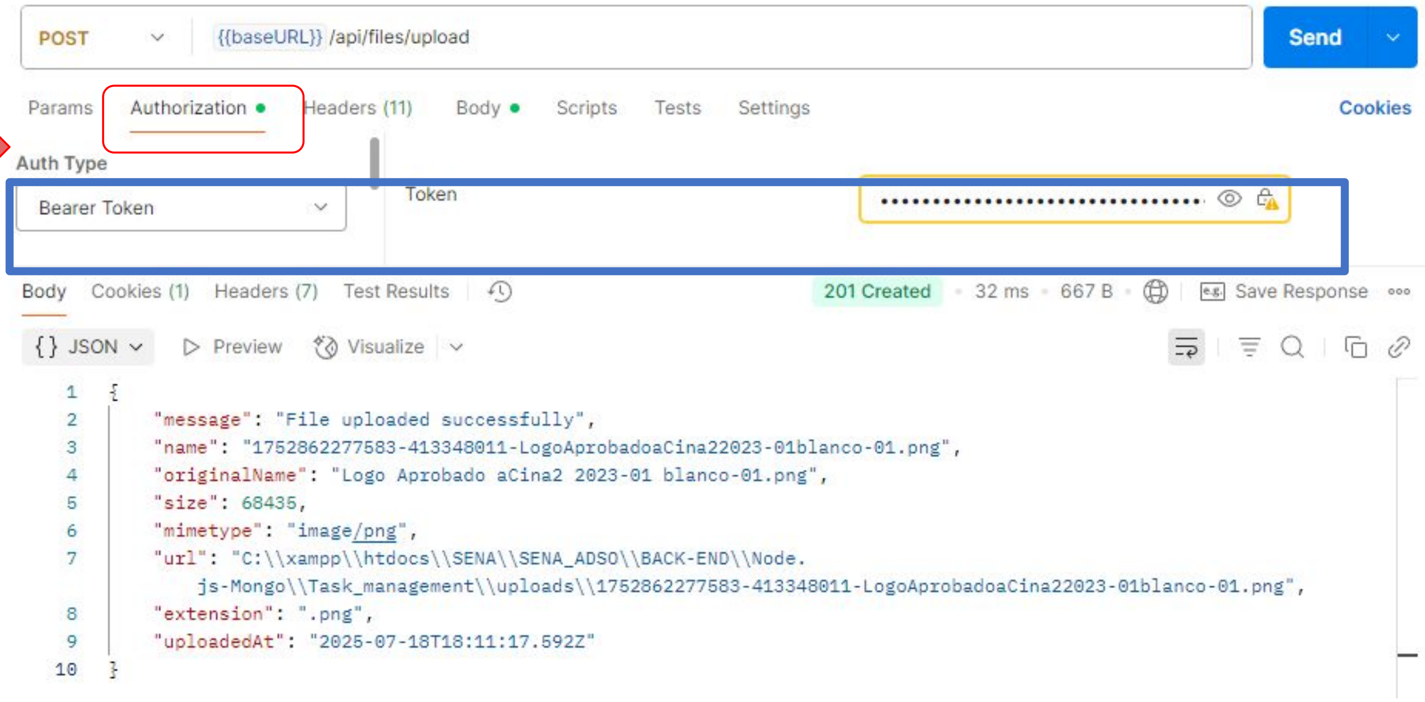


```
Task_management
  app
  config
  controllers
  library
  middleware
  models
  node_modules
  routes
  uploads
    1752857170114-572130521
  views
  .env
  package-lock.json
  package.json
  server.js
```

Mongodb

Pruebas

TOKEN



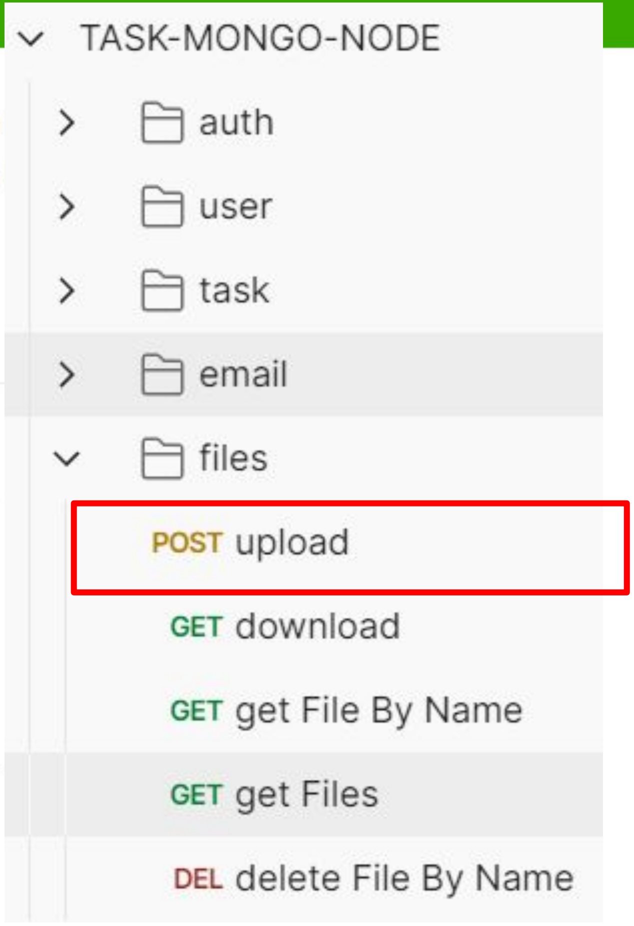
POST {{baseURL}} /api/files/upload

Auth Type: Bearer Token

Token: ••••••••••••••••••••••••••••••••••

201 Created · 32 ms · 667 B

```
{
    "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"
}
```



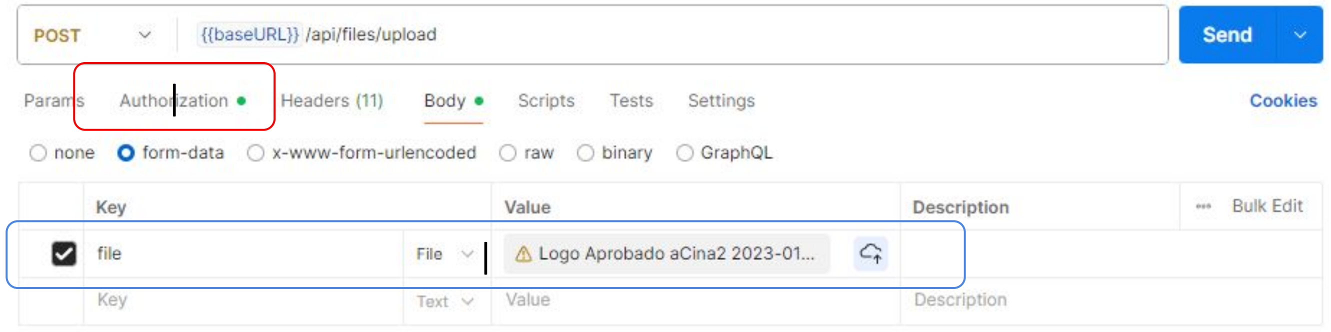
```
TASK-MONGO-NODE
  > auth
  > user
  > task
  > email
  v files
      POST upload
      GET download
      GET get File By Name
      GET get Files
      DEL delete File By Name
```

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

Mongodb

Pruebas

TOKEN



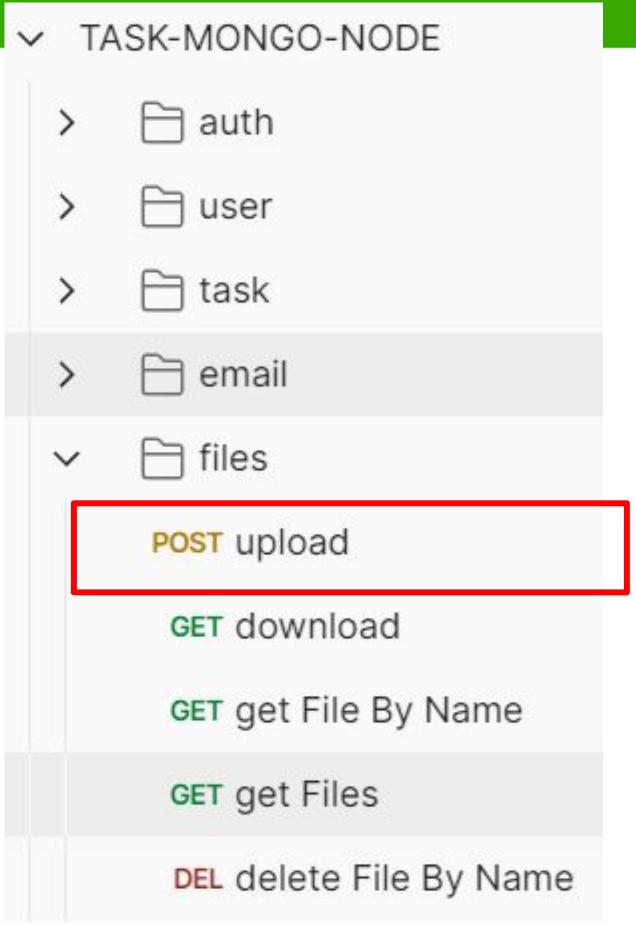
POST {{baseURL}} /api/files/upload

Send

Params | Authorization ● | Headers (11) | Body ● | Scripts | Tests | Settings | Cookies

○ none [✓ form-data] ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key		Value		Description
✓	file	File	⚠ Logo Aprobado aCina2 2023-01...		
	Key	Text	Value		Description

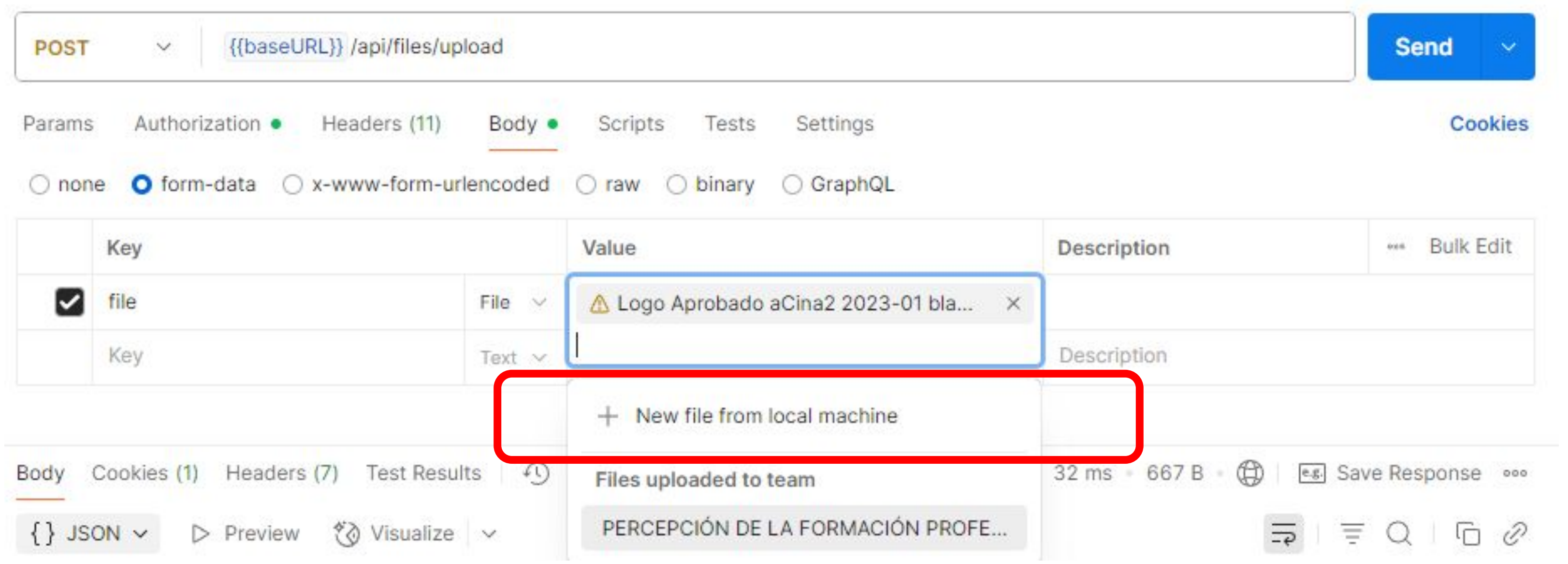


TASK-MONGO-NODE
- auth
- user
- task
- email
- files
  - POST upload
  - GET download
  - GET get File By Name
  - GET get Files
  - DEL delete File By Name

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

Mongodb

Pruebas


POST {{baseURL}} /api/files/upload

Send

Params | Authorization | Headers (11) | Body | Scripts | Tests | Settings | Cookies

none [✓ form-data] x-www-form-urlencoded raw binary GraphQL

	Key		Value	Description
✓	file	File	Logo Aprobado aCina2 2023-01 bla...	
	Key	Text		Description

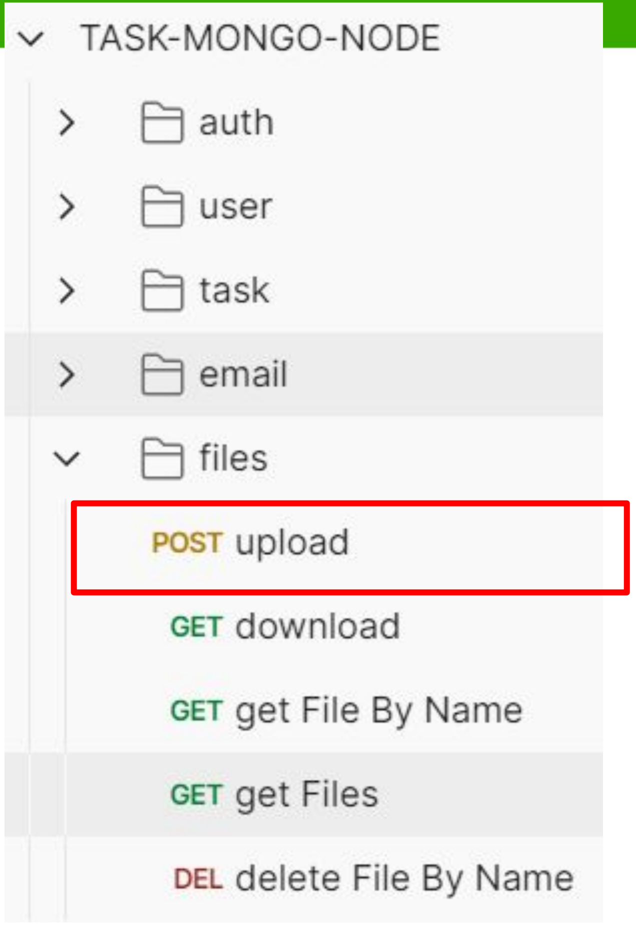
New file from local machine

Files uploaded to team

PERCEPCIÓN DE LA FORMACIÓN PROFE...

Body | Cookies (1) | Headers (7) | Test Results | 32 ms · 667 B | Save Response

{} JSON | Preview | Visualize

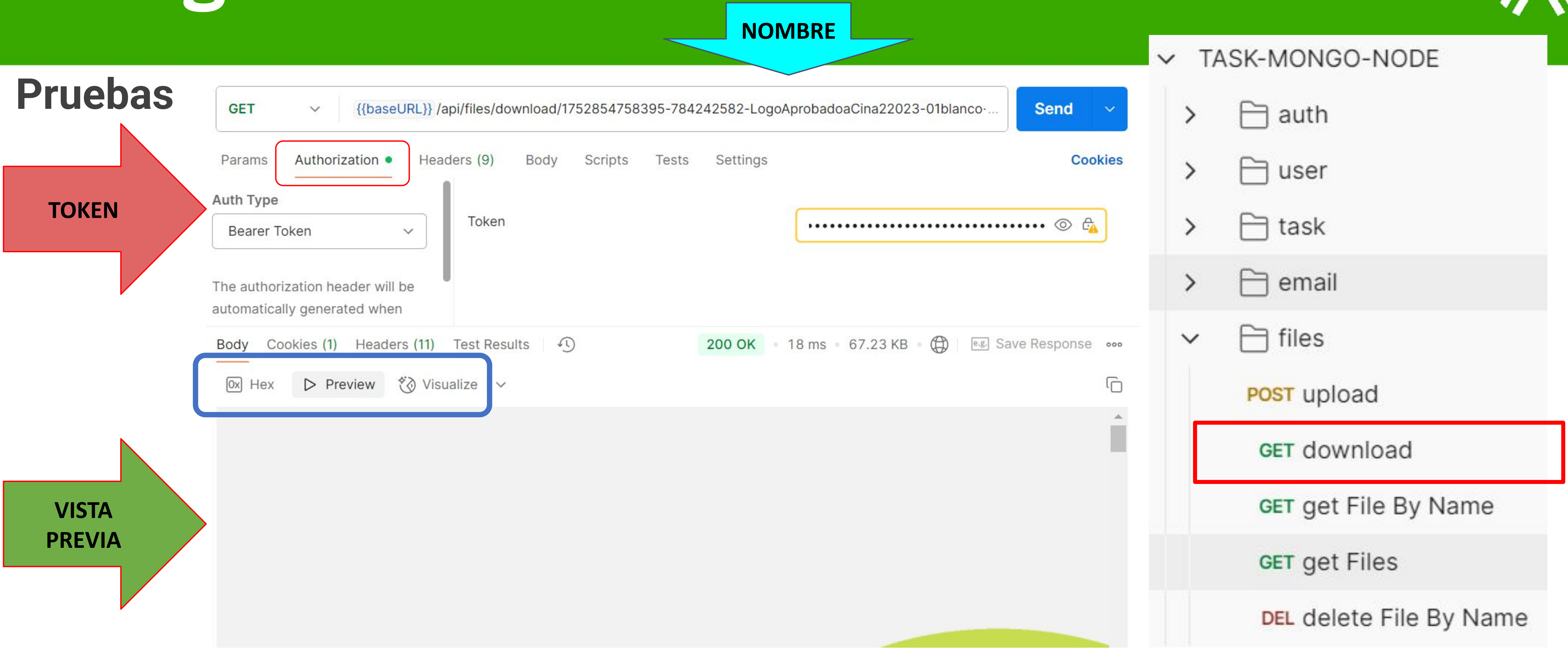

```
TASK-MONGO-NODE
  auth
  user
  task
  email
  files
    POST upload
    GET download
    GET get File By Name
    GET get Files
    DEL delete File By Name
```

Cargar el archivo

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

Mongodb

Pruebas



GET {{baseURL}} /api/files/download/1752854758395-784242582-LogoAprobadoaCina22023-01blanco-...

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

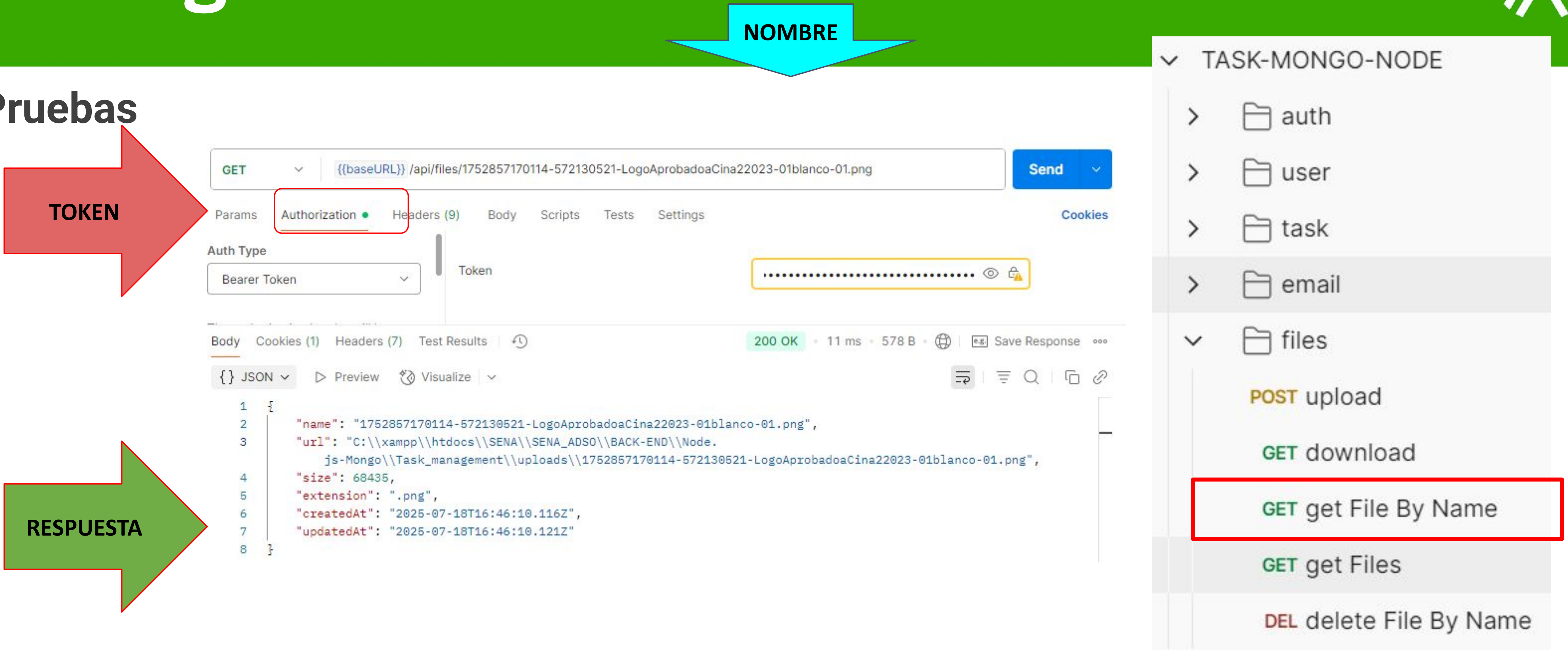
Hex | Preview | Visualize

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

Mongodb

Pruebas



```
{
    "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"
}
```

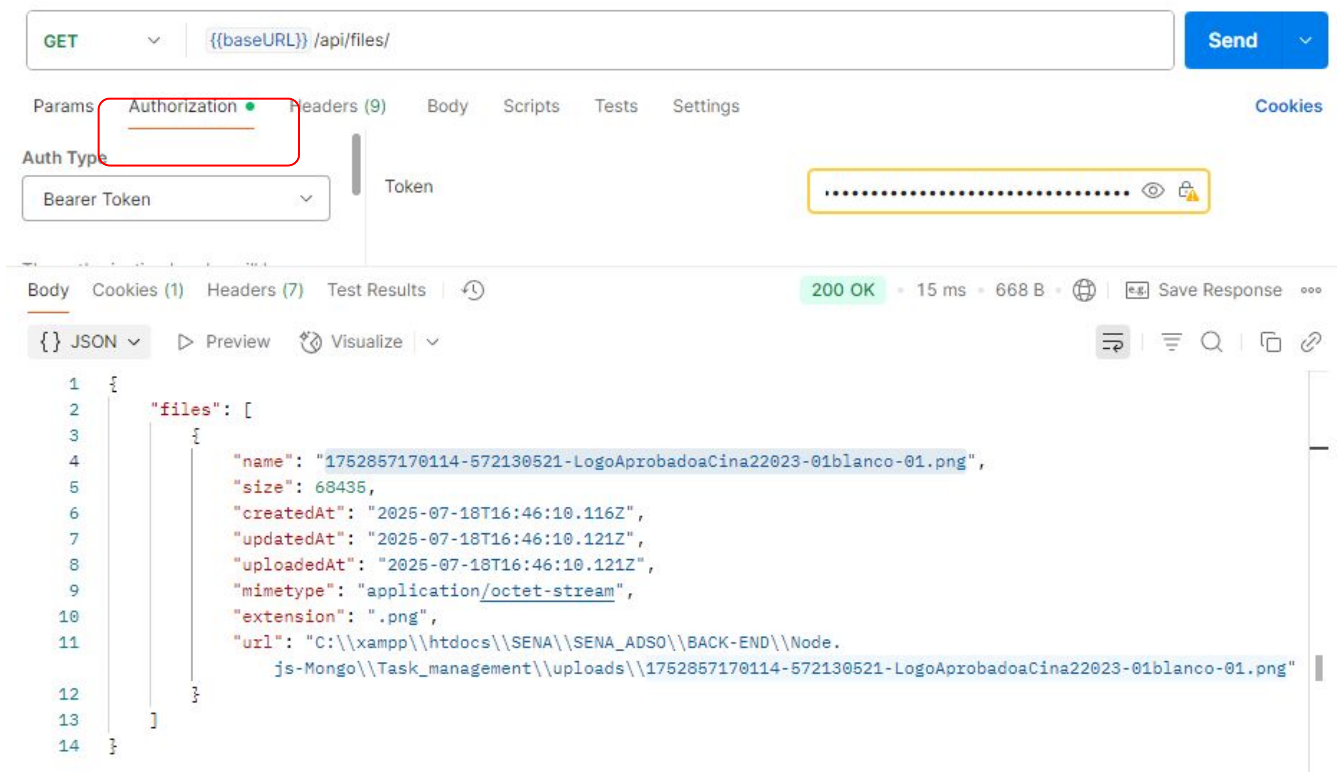
MÉTODO	RUTA	CONTROLADOR	MODELO
GET	http://localhost:3000/api/files/:filename	Files	N/A

Mongodb

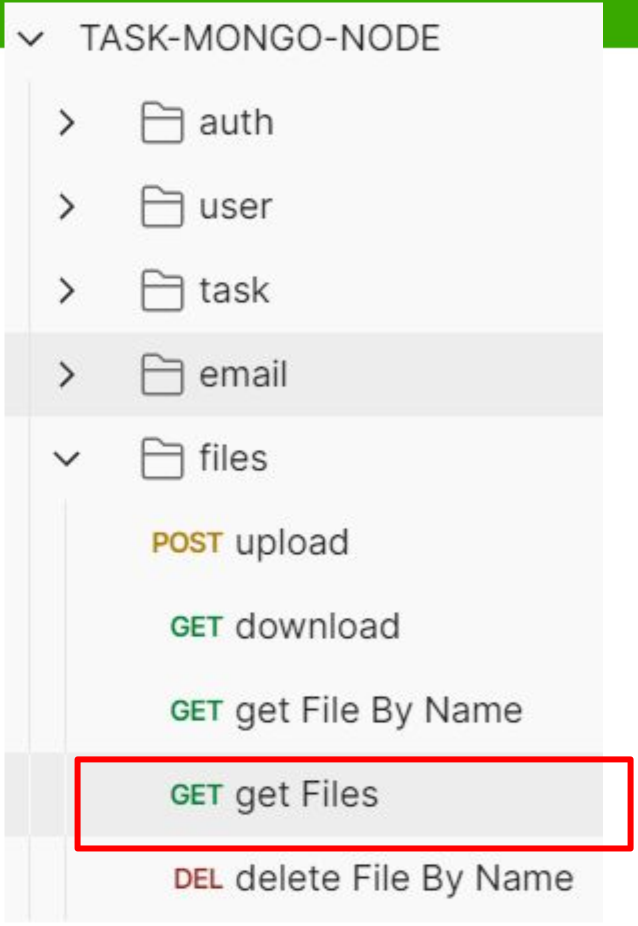
Pruebas

TOKEN

RESPUESTA



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



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

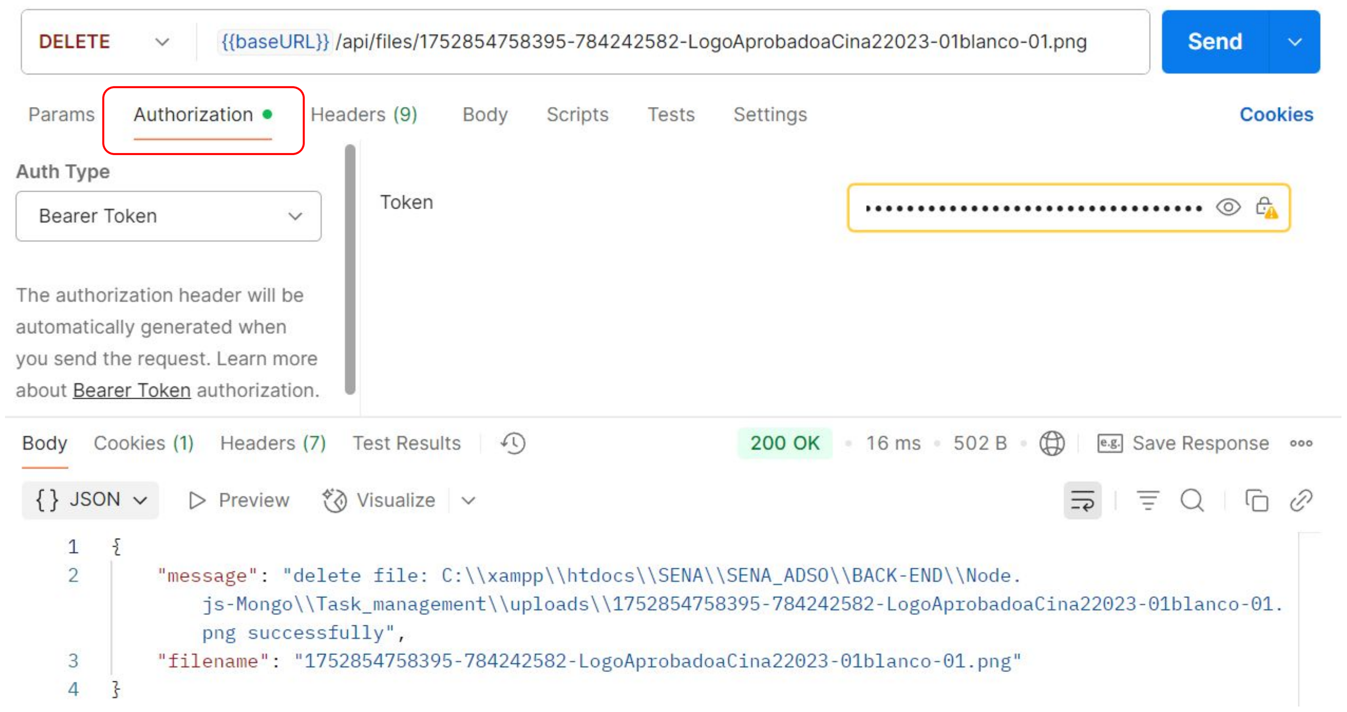
Mongodb

Pruebas

NOMBRE

TOKEN

RESPUESTA



DELETE {{baseURL}} /api/files/1752854758395-784242582-LogoAprobadoaCina22023-01blanco-01.png

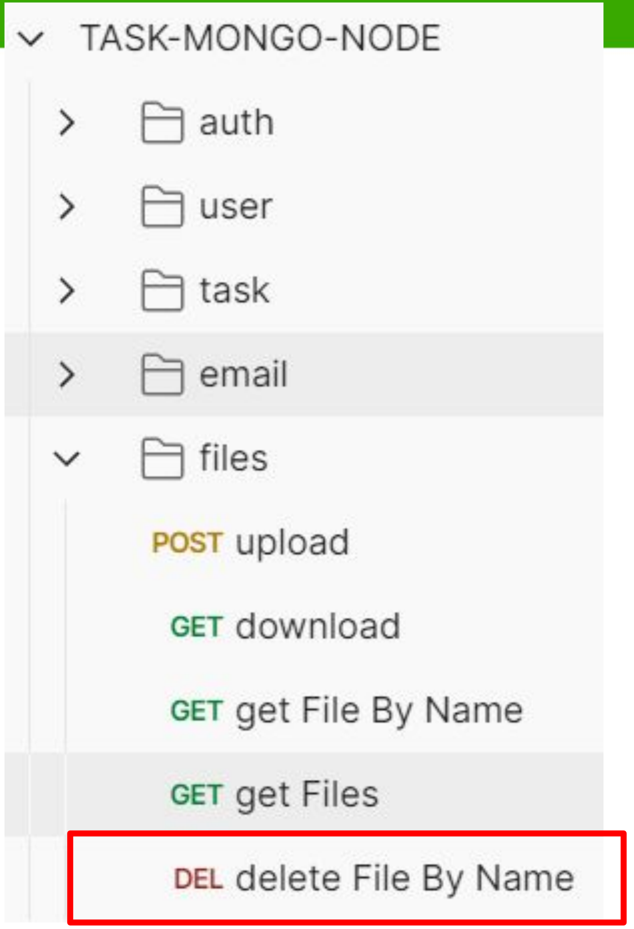
Auth Type: Bearer Token

Token: ••••••••••••••••••••••••••••••••••

The authorization header will be automatically generated when you send the request. Learn more about Bearer Token authorization.

200 OK · 16 ms · 502 B

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



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

SENA

G R A C I A S