# Creo el proyecto

$npm init –yes

$git init

Instalo los package:

Express: para el servidor

Motor de plantilla: para enviar html se usan los motores de plantilla por ejemplo: handlebars, pub, ejs, etc.

Multer: modulo muy popular para subir imágenes en nodejs

Nodemon: para q reinicie automáticamente el server

*$npm i express ejs multer*

## Empezando

Trabajando en src/inde.js creando el servidor y configuración básica

Configuro ejs, son html mejorados, express tiene una integración por defecto con ejs

Express busca la carpeta VIEWS en la raíz asi que debo darle un nuevo PATH

Tambien le debo decir al servidor que USE el modulo para subir imágenes: multer y su configuración.

Creo el archivo views/indes.ejs y un form básico con un input file. Al usar POST la route recibe por medio de req.file un objeto con lo siguiente:

{ fieldname: 'imageInput',

originalname: 'trailerVideos.png',

encoding: '7bit',

mimetype: 'image/png',

destination: 'D:\\Desarrollo\\nodejs-upload-img\\src\\public\\images',

filename: '48852f77e4f7e6b9488ed5a5da13ae65',

path:

'D:\\Desarrollo\\nodejs-upload-img\\src\\public\\images\\48852f77e4f7e6b9488ed5a5da13ae65',

size: 1106349 }

La imagen se subirá con el nombre filename y sin extensión, para renombrarla:

//config multer para q guarde la img con su nombre original. Le paso esto a Multer Middleware

const multerStorage=multer.diskStorage({

destination:path.join(\_\_dirname,'public/images'),

filename:(req,file,cb)=>{

//si no hay error le paso null

cb(null,file.originalname)

}

})

Y luego uso esta configuración:

app.use(multer({

storage:multerStorage,

dest: path.join(\_\_dirname,'public/images')

}).single('imageInput'))

## Refactoring

Creo una carpeta routes/index.routes.js creo las rutas en un archivo independiente, en este caso multer esta disponible para toda la app(index.js) pero si la quisiera en un solo archivo también lo muevo(en este caso solo lo ocupo en index.route.js)

## Static Files

Ahora quiero acceder a esas imágenes para ello configuro mi static files,

//STATIC FILES

//desde el navegador :3000/images/cualquier\_img.ext

app.use(express.static(path.join(\_\_dirname,'public')))

## Filtrar extension

//la configuracion de multer, puedo cambiar single('nombre\_del\_campo')

const upload=multer({

storage:multerStorage,

dest: path.join(\_\_dirname,'../public/images'),

limits:{fileSize:1000000},//size en bytes

fileFilter:(req,file,cb)=>{

//expresion regular /abc/

const fileType= /jpeg|jpg|png/

//comprueba q el tipo subido coincide con alguna de las esperadas

const mimetype=fileType.test(file.mimetype)

//ahora compruebo la extension con un metodo de path pasandole el originalname

const extname=fileType.test(path.extname(file.originalname))

if(mimetype && extname){

// el cb dice: no error, continua

cb(null,true)

}else{

//sino error

console.log(`${mimetype} y ${extname}`)

cb('Error: Archivo debe ser imagen')

}

}

}).single('imageInput')

## UUID

Modulo para generar un nombre para las imágenes basado en algoritmo.

<https://www.npmjs.com/package/uuid>

UUID tiene distintas versiones, en este caso usare la 4:

const uuid=require('uuid/v4')

para usar solo llamo al método y le concateno la extensión:

uuid()+path.extname(file.originalname).toLowerCase()

## Conclusion

No es comun subir las imágenes a tu servidor, lo común es usar un servicio dedicado como

<https://aws.amazon.com/es/s3/>

Este servicio me devuelve una url de donde se guardo la imagen y luego en mi App o BD solo guardo esa URL para acceder a ella.

Otro que esta basado en amazonS3 es

<https://cloudinary.com/>