

Universidad San Carlos de Guatemala  
Centro Universitario de Occidente  
División de Ciencias de la Ingeniería  
Manejo e implementación de archivos  
Ing. Christian Quiroa

**Byron Fernando Torres Ajxup 201731523**

**Manual Tecnico** GRAFILES

# GRAFILES

Quetzaltenango 31 de OCTUBRE de 2024

# Indice

<b>Indice</b>	<b>2</b>
<b>Introduccion</b>	<b>3</b>
<b>Requerimientos Técnicos</b>	<b>4</b>
Resumen del Proyecto	4
Estructura del Proyecto	5
La estructura del proyecto es la siguiente:	5
<b>Dockerfile del Backend</b>	<b>6</b>
<b>Dockerfile de Frontend</b>	
<b># Usa una imagen de Node.js</b>	<b>7</b>
<b>FROM node:16.14.0</b>	<b>7</b>
<b># Configura el directorio de trabajo</b>	<b>7</b>
<b>WORKDIR /app</b>	<b>7</b>
<b># Copia el package.json y package-lock.json</b>	<b>7</b>
<b>COPY package*.json ./</b>	<b>7</b>
<b># Instala dependencias</b>	<b>7</b>
<b>RUN npm install</b>	<b>7</b>
<b># Copia el resto del código</b>	<b>7</b>
<b>COPY . .</b>	<b>7</b>
<b># Expone el puerto de Vite</b>	<b>7</b>
<b>EXPOSE 5173</b>	<b>7</b>
<b># Comando para iniciar Vite</b>	<b>7</b>
<b>CMD ["npm", "run", "dev"]</b>	<b>7</b>
<b>Levantar el proyecto:</b>	<b>8</b>
Acceder a la Aplicación	8
<b>Detener y Eliminar los Contenedores</b>	<b>8</b>
<b>Consideraciones</b>	<b>8</b>
<b>Problemas Comunes</b>	<b>8</b>
<b>Scripts mongo</b>	<b>9</b>
<b>Conclusiones</b>	<b>11</b>

Quetzaltenango 26 de Septiembre de 2024

## Introduccion

Este manual técnico está diseñado para proporcionar una guía detallada sobre la configuración y ejecución de Grafiles, un gestor de archivos que permite a los usuarios crear carpetas y manejar archivos en formatos como TXT, DOCX, así como imágenes en PNG y JPG. La aplicación cuenta con roles de usuario diferenciados: empleado y admin, lo que permite gestionar el acceso y las funciones disponibles según el perfil del usuario.

## Requerimientos Técnicos

**Docker:** Asegúrate de tener Docker instalado. Puedes descargarlo desde Docker.

**Docker Compose:** Asegúrate de que Docker Compose está incluido en tu instalación de Docker.

## Resumen del Proyecto

Este proyecto es una aplicación web construida utilizando Node.js v16.14.0 y Express. La aplicación se gestiona con Docker, lo que permite un entorno de desarrollo y producción consistente. Las principales dependencias utilizadas son:

bcryptjs: Para la encriptación de contraseñas.

cors: Para permitir solicitudes de recursos cruzados.

dotenv: Para manejar variables de entorno.

express: Framework web para Node.js.

express-session: Middleware para gestionar sesiones.

mongoose: ODM para MongoDB.

nodemon: Herramienta para reiniciar automáticamente la aplicación durante el desarrollo.

## Estructura del Proyecto

La estructura del proyecto es la siguiente:

ARCHIVOS/

```
|
|└─ Backend/
|  |└─ config/
|  |└─ controllers/
|  |└─ models/
|  |└─ node_modules/
|  |└─ package.json
|  |└─ router/
|  |└─ index.js
|  └─ Dockerfile
|
|└─ Frontend/
|  |└─ src/
|  |└─ node_modules/
|  |└─ package.json
|  └─ vite.config.js
|
└─ docker-compose.yml
```

# Dockerfile del Backend

```
# Usa una imagen de Node.js
FROM node:16.14.0

# Configura el directorio de trabajo
WORKDIR /app

# Copia el package.json y package-lock.json
COPY package*.json ./

# Instala dependencias
RUN npm install

# Copia el resto del código
COPY . .

# Expone el puerto de la API
EXPOSE 3000

# Comando para iniciar la aplicación
CMD ["npm", "start"]
```

# Dockerfile de Frontend

# Usa una imagen de Node.js

```
FROM node:16.14.0
```

```
# Configura el directorio de trabajo
```

```
WORKDIR /app
```

```
# Copia el package.json y package-lock.json
```

```
COPY package*.json ./
```

```
# Instala dependencias
```

```
RUN npm install
```

```
# Copia el resto del código
```

```
COPY . .
```

```
# Expone el puerto de Vite
```

```
EXPOSE 5173
```

```
# Comando para iniciar Vite
```

```
CMD ["npm", "run", "dev"]
```

## Levantar el proyecto:

`sudo docker-compose up --build`

## Acceder a la Aplicación

**Frontend:** <http://localhost:5173/>

**Backend:** <http://localhost:3000/> (si es necesario realizar pruebas de API).

## Detener y Eliminar los Contenedores

`sudo docker-compose down`

## Consideraciones

Asegúrate de tener los permisos necesarios para ejecutar Docker. Es posible que necesites utilizar **sudo** en sistemas basados en Unix.

Revisa la configuración de tu red si encuentras problemas para acceder a las URL. Asegúrate de que no haya conflictos de puertos.

## Problemas Comunes

Error al conectar a MongoDB: Asegúrate de que el contenedor de MongoDB está funcionando y que tu aplicación backend se está conectando a la URL correcta.

Problemas de CORS: Si estás haciendo peticiones desde el frontend al backend, asegúrate de que el middleware CORS esté correctamente configurado en tu aplicación Express.



## Scripts mongo

Usuarios:

use grafilesDB

```
db.usuarios.insertOne({
  nombre: "Admin User",
  username: "admin01",
  password: "hashed_password_here", // Recuerda que debes encriptarla previamente
  rol: "ADMIN",
  rootId: null,
  sharedId: null
})
```

Files

```
db.files.insertOne({
  name: "Root Folder",
  type: "folder",
  parentId: null,
  createdBy: ObjectId("ID_del_usuario"), // Reemplazar con el _id real del usuario
  visible: true,
  createdAt: new Date(),
  updatedAt: new Date()
})
```

```
}}
```

Compartidos:

```
db.share.insertOne({  
  name: "Shared Document",  
  type: "file",  
  extension: "txt",  
  content: "Este documento ha sido compartido.",  
  parentId: null,  
  createdBy: ObjectId("ID_del_usuario"),  
  sharedBy: ObjectId("ID_usuario_destino"),  
  visible: true,  
  createdAt: new Date(),  
  updatedAt: new Date()  
})
```

Consultas:

```
db.files.find({ parentId: ObjectId("ID_de_la_carpeta") })
```

```
db.files.find({ createdBy: ObjectId("ID_del_usuario"), type: { $in: ["file", "image"] } })
```

```
db.share.find({ sharedBy: ObjectId("ID_usuario_destino") })
```

```
db.basura.find({ visible: false })
```

```
db.usuarios.find({ rol: "EMPLEADO" })
```

```
db.files.updateOne({ _id: ObjectId("ID_del_archivo") }, { $set: { visible: false } })
```

```
db.usuarios.updateOne({ _id: ObjectId("ID_del_usuario") }, { $set: { rootId:  
ObjectId("ID_de_la_nueva_carpeta") } })
```

```
db.share.updateOne(  
  { _id: ObjectId("ID_del_archivo_compartido") },  
  { $set: { sharedBy: ObjectId("ID_usuario_destino") } }  
)
```

## Conclusiones

Este manual debería proporcionar toda la información necesaria para que otro programador pueda levantar y trabajar con el proyecto en su máquina. Si tienes más preguntas o necesitas ayuda adicional, no dudes en consultarlo.

