

# Instrucciones

Se han creado 4 aplicaciones:

- Stack LAMP:
  - API con Lumen: <https://github.com/carlosgmr/tfm-api-lumen>
  - APP web con Laravel: <https://github.com/carlosgmr/tfm-app-laravel>
- Stack MEAN
  - API con Express: <https://github.com/carlosgmr/tfm-api-express>
  - APP web con Angular: <https://github.com/carlosgmr/tfm-app-angular>

En el readme.md de cada uno de los proyectos viene especificada las instrucciones para instalarlas. Se recomienda seguir las instrucciones desde GitHub porque están más claras al tener formato.

**IMPORTANTE 1:** para el despliegue de las aplicaciones se utiliza el sistema de contenedores Docker. Para instalar Docker Community Edition (CE) en su sistema operativo consulte la página <https://docs.docker.com/install/>

**IMPORTANTE 2:** las instrucciones vienen descritas suponiendo que se trabaja en una distribución Linux con base Debian (por ejemplo Ubuntu). Si se trabaja desde otro sistema operativo, las instrucciones pueden cambiar.

**IMPORTANTE 3:** las aplicaciones tipo API trabajan con una base de datos MySQL. Se puede instalar MySQL en el propio sistema operativo o en un contenedor de Docker. Una vez se tenga funcionando la base de datos, seguir las instrucciones del siguiente apartado.

**IMPORTANTE 4:** para cualquier duda o problema durante la instalación de los proyectos, contactad con el autor: Carlos Giovanni Molina Ronceros ([giovanni.molina@edu.uah.es](mailto:giovanni.molina@edu.uah.es)).

## Instalación Base de Datos

1. Conectarse a la base de datos.
2. Crear un esquema, por ejemplo "tfm".
3. Ejecutar el script "database/model.sql". Este script creará todas las tablas e insertará datos para realizar pruebas.
4. Después de realizar la importación de datos, se tienen las siguientes cuentas para acceder a las aplicaciones
  1. Tipo de cuenta "Administrador":
    - **email:** administrator@tfm.com **password:** 1234
    - **email:** administrator2@tfm.com **password:** 1234
    - **email:** administrator3@tfm.com **password:** 1234
  2. Tipo de cuenta "Instructor":
    - **email:** instructor@tfm.com **password:** 1234
    - **email:** instructor2@tfm.com **password:** 1234
    - **email:** instructor3@tfm.com **password:** 1234
  3. Tipo de cuenta "Usuario":
    - **email:** user@tfm.com **password:** 1234
    - **email:** user2@tfm.com **password:** 1234
    - **email:** user3@tfm.com **password:** 1234

# Instalación API Lumen

1. Clonar o descargar proyecto desde GitHub:

```
git clone https://github.com/carlosgrmr/tfm-api-lumen.git api-lumen
```

2. Entrar en la carpeta del proyecto:

```
cd api-lumen
```

3. Copiar el archivo ".env.example" y renombrarlo a ".env":

```
cp .env.example .env
```

4. Abrir el archivo ".env" y completar las siguientes variables de configuración:

1. **APP\_KEY:** cadena aleatoria que se utilizará para tareas de encriptación. Se recomienda que tenga una longitud mínima de 32 caracteres y que contenga letras en mayúsculas y minúsculas y números.
2. **DB\_HOST:** IP o nombre de dominio donde se encuentra la base de datos. Importante Si la base de datos se encuentra en el mismo equipo, no utilizar localhost o 127.0.0.1, sino la IP que tiene el equipo en la red local (utilizad ifconfig para averiguarla).
3. **DB\_PORT:** puerto de conexión con la base de datos.
4. **DB\_DATABASE:** nombre del esquema de la base de datos que se utilizará.
5. **DB\_USERNAME:** nombre de usuario para conectar a la base de datos.
6. **DB\_PASSWORD:** contraseña de usuario para conectar a la base de datos.
7. **JWT\_SECRET:** cadena aleatoria que se utilizará para firmar los token de autenticación JWT. Se recomienda que tenga una longitud mínima de 32 caracteres y que contenga letras en mayúsculas y minúsculas y números.
8. **PASSWORD\_ALGO:** algoritmo para encriptar las contraseñas de usuario. Los valores admitidos son sha1 y bcrypt. Para compatibilidad total entre clientes web y APIs, utilizad sha1.

5. Construir la imagen de Docker. Hay que estar dentro de la carpeta del proyecto:

```
docker build -t carlosgrmr/api-lumen .
```

6. Construir contenedor Docker con la imagen anterior y ejecutarlo:

```
docker run -p 8970:80 --detach --memory 1g --name api-lumen carlosgrmr/api-lumen
```

7. La aplicación se encuentra accesible desde <http://localhost:8970>

# Instalación APP Laravel

1. Clonar o descargar proyecto desde GitHub:

```
git clone https://github.com/carlosgmr/tfm-app-laravel.git app-laravel
```

2. Entrar en la carpeta del proyecto:

```
cd app-laravel
```

3. Copiar el archivo ".env.example" y renombrarlo a ".env":

```
cp .env.example .env
```

4. Abrir el archivo ".env" y completar las siguientes variables de configuración:

1. **APP\_KEY:** cadena aleatoria que se utilizará para tareas de encriptación. Se recomienda que tenga una longitud mínima de 32 caracteres y que contenga letras en mayúsculas y minúsculas y números.
2. **API\_URL:** URL donde está desplegada la API Lumen. Importante Si API se encuentra en el mismo equipo, no utilizar localhost o 127.0.0.1, sino la IP que tiene el equipo en la red local (utilizad ifconfig para averiguarla). No olvidar añadir una barra inclinada (/) al final de la URL. Ejemplo:  
`http://192.168.1.61:8970/`

5. Construir la imagen de Docker. Hay que estar dentro de la carpeta del proyecto:

```
docker build -t carlosgmr/app-laravel .
```

6. Construir contenedor Docker con la imagen anterior y ejecutarlo:

```
docker run -p 8972:80 --detach --memory 1g --name app-laravel carlosgmr/app-laravel
```

7. La aplicación se encuentra accesible desde <http://localhost:8972>

# Instalación API Express

1. Clonar o descargar proyecto desde GitHub:

```
git clone https://github.com/carlosgmr/tfm-api-express.git api-express
```

2. Entrar en la carpeta del proyecto:

```
cd api-express
```

3. Crear el fichero "config.js"

```
touch config.js
```

Su contenido será el siguiente:

```
module.exports = {
  'db': {
    'host': '192.168.1.XXX', // IP o nombre de dominio donde se encuentra la base de datos.
    // Importante: Si la base de datos se encuentra en el mismo equipo, no utilizar localhost o 127.0.0.1,
    // sino la IP que tiene el equipo en la red local (utilizad ifconfig para averiguarla)
    'port': 0000, // puerto de conexión con la base de datos
    'user': 'usuario', // nombre de usuario para conectar a la base de datos
    'password': 'contraseña', // contraseña de usuario para conectar a la base de datos
    'database': 'base de datos', // nombre del esquema de la base de datos que se utilizará
    'dateStrings': true
  },
  'jwt': {
    'issuer': 'tfm-api-express',
    'secret': 'your-secret-key' // cadena aleatoria que se utilizará para firmar los token de
    // autenticación JWT. Se recomienda que tenga una longitud mínima de 32 caracteres y que contenga letras
    // en mayúsculas y minúsculas y números
  },
  'passwordAlgo': 'sha1|bcrypt' // algoritmo para encriptar las contraseñas de usuario. Los valores
  // admitidos son sha1 y bcrypt. Para compatibilidad total entre clientes web y APIs, utilizad sha1
};
```

4. Construir la imagen de Docker. Hay que estar dentro de la carpeta del proyecto:

```
docker build -t carlosgmr/api-express .
```

5. Construir contenedor Docker con la imagen anterior y ejecutarlo:

```
docker run -p 8971:3000 --detach --memory 1g --name api-express carlosgmr/api-express
```

6. La aplicación se encuentra accesible desde <http://localhost:8971>

# Instalación APP Angular

1. Instalar Node.js y npm:

```
sudo apt-get install -y nodejs
```

Si el despliegue se realiza en un sistema operativo o distribución Linux distinta, consultar esta página <https://nodejs.org/es/download/package-manager/>

2. Instalar globalmente Angular Cli:

```
npm install -g @angular/cli
```

3. Clonar o descargar proyecto desde GitHub:

```
git clone https://github.com/carlosgmr/tfm-app-angular.git app-angular
```

4. Entrar en la carpeta del proyecto:

```
cd app-angular
```

5. Instalar dependencias del proyecto:

```
npm install
```

6. Abrir el archivo "src/environments/environment.prod.ts"

7. Modificar la clave **api** > **url** y poner la URL donde se encuentra desplegada la API Express. Importante: Si API se encuentra en el mismo equipo, no utilizar localhost o 127.0.0.1, sino la IP que tiene el equipo en la red local (utilizad ifconfig para averiguarla). No olvidar añadir una barra inclinada (/) al final de la URL. Ejemplo:  
`http://192.168.1.61:8971/`

8. Generar contenido de despliegue:

```
ng build --prod --build-optimizer
```

9. Construir la imagen de Docker. Hay que estar dentro de la carpeta del proyecto:

```
docker build -t carlosgmr/app-angular .
```

10. Construir contenedor Docker con la imagen anterior y ejecutarlo:

```
docker run -p 8973:80 --detach --memory 1g --name app-angular carlosgmr/app-angular
```

11. La aplicación se encuentra accesible desde <http://localhost:8973>