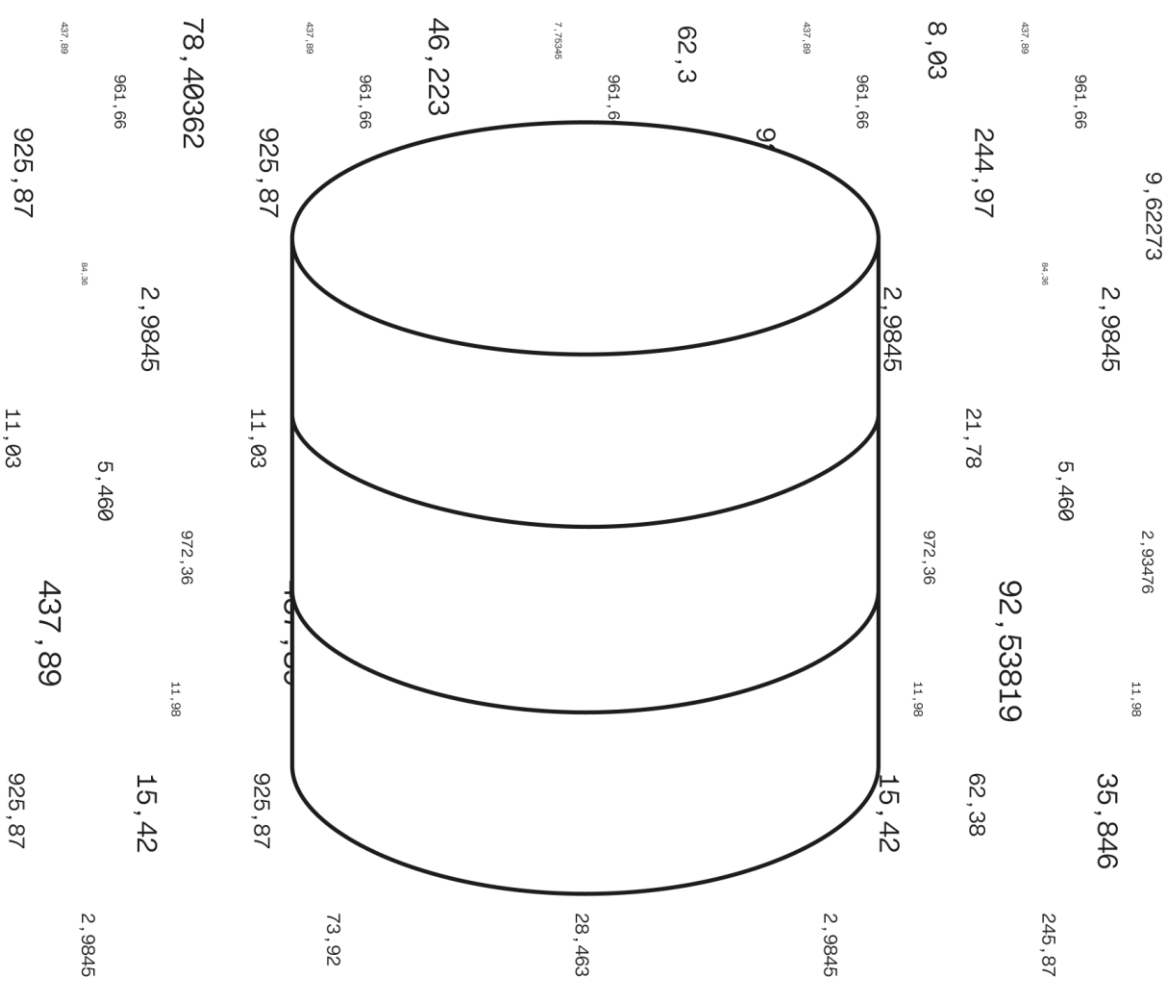


# GRAM ▼

## Control



GUÍA DE INSTALACIÓN



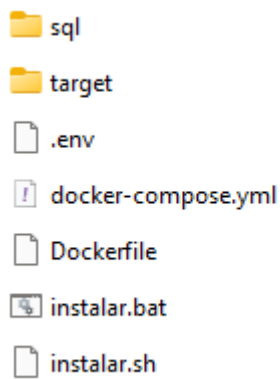
## ÍNDICE

INTRODUCCIÓN.....	4
PROCESO DE INSTALACIÓN.....	4
INSTALACIÓN DE LA API, BASE DE DATOS Y phpMyAdmin.....	7
ERRORES.....	10

## INTRODUCCIÓN

Esta guía describe el proceso necesario para poder utilizar la Z8i API. Adjunto a este manual se facilitan los archivos necesarios para la instalación.

Dichos archivos son los siguientes:

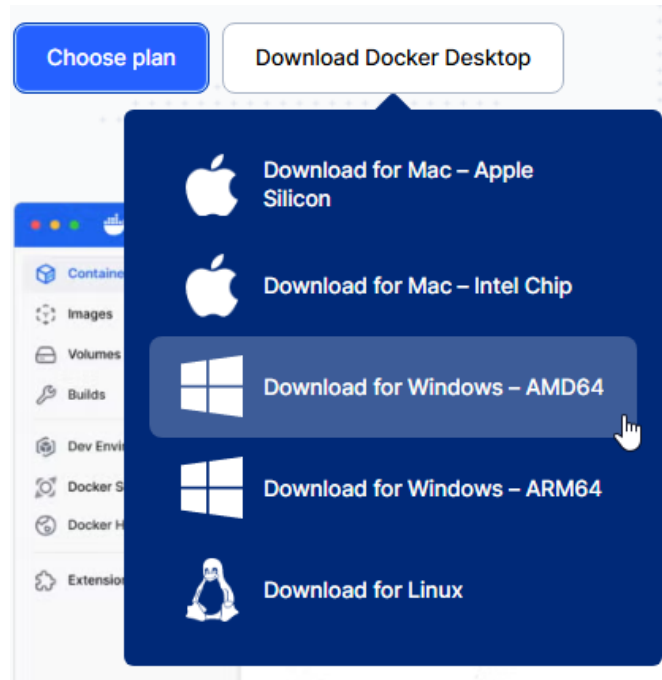


## PROCESO DE INSTALACIÓN

El primer paso es instalar Docker Desktop. Usaremos el siguiente link para descargar el instalador necesario.

<https://www.docker.com/products/docker-desktop/>

Es importante descargar el instalador apropiado para nuestra máquina, el más común será AMD x86 para máquinas Windows.



Una vez hemos descargado el indicado, continuaremos con el proceso de instalación hasta que finalice.

Al ejecutar por primera vez nos pedirá que aceptemos los términos.

### Docker Subscription Service Agreement

By selecting **accept**, you agree to the [Subscription Service Agreement](#), the [Docker Data Processing Agreement](#), the [Data Privacy Policy](#) and the [Docker AI Supplemental Terms](#).

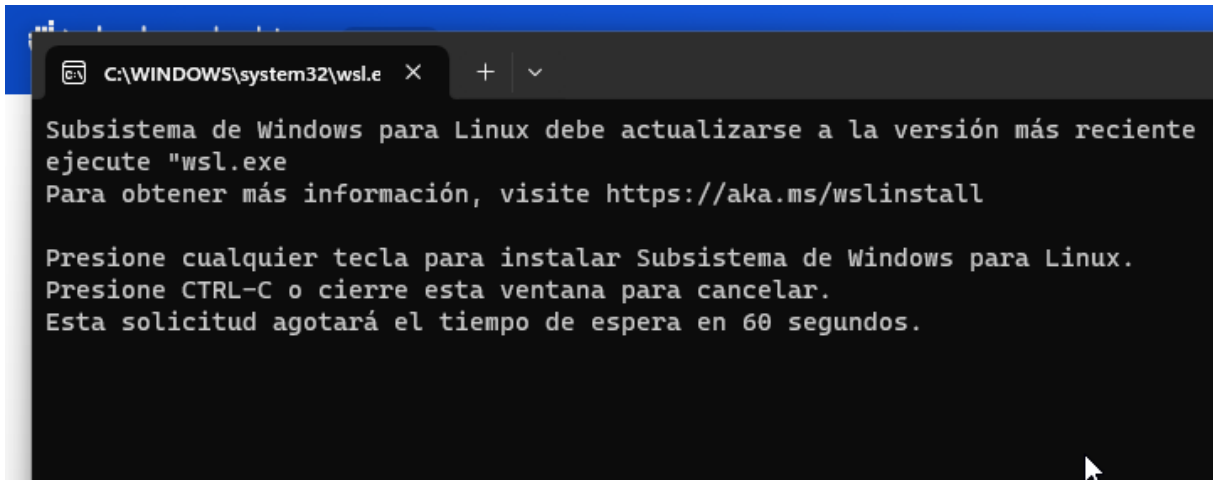
Commercial use of Docker Desktop at a company of more than 250 employees OR more than \$10 million in annual revenue requires a paid subscription (Pro, Team, or Business). [See subscription details](#)

[View Full Terms](#)

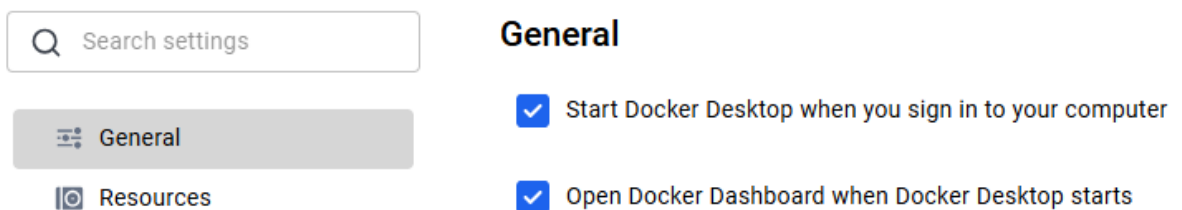
[Accept](#)

[Close](#)

Una vez aceptados nos pedirá que instalemos el Subsistema de Windows para Linux, presiona cualquier tecla para que se instale.



Una vez hayamos completado la instalación. Abriremos Docker desktop. Nos aseguraremos de que este arranque cuando iniciemos el terminal.



Asegúrate de tener marcada esta opción en **Configuración -> general**

## INSTALACIÓN DE LA API, BASE DE DATOS Y phpMyAdmin

Una vez esté todo listo esta herramienta nos facilitará la instalación de la API puesto que nos generará un entorno con todo lo necesario para su funcionamiento

El siguiente paso será ejecutar el archivo instalar.bat. En el caso de estar usando Linux, ejecutaremos instalar.sh.

Docker nos generará una base de datos siguiendo el esquema que le hemos facilitado.

```
version: "3.8"

services:
  db:
    image: mysql:8.0
    container_name: open_z8_mysql
    profiles: ["localdb"]
    environment:
      MYSQL_ROOT_PASSWORD: Gram1231
      MYSQL_DATABASE: open_z8
      MYSQL_USER: apiuser
      MYSQL_PASSWORD: Gram1231
    volumes:
      - db_data:/var/lib/mysql
      - ./sql:/docker-entrypoint-initdb.d
    restart: unless-stopped
    ports:
      - "3306:3306"
  api:
    build: .
    container_name: open_z8_api
    env_file: .env
    depends_on:
      - db
    ports:
      - "8080:8080"
    restart: unless-stopped
  phpmyadmin:
    image: phpmyadmin/phpmyadmin:latest
    container_name: open_z8_phpmyadmin
    profiles: ["localdb"]
    depends_on:
      - db
    environment:
      PMA_HOST: db
      PMA_USER: root
      PMA_PASSWORD: Gram1231
      UPLOAD_LIMIT: 64M
    ports:
      - "8081:80"
    restart: unless-stopped

volumes:
  db_data:
```

Hay que tener en cuenta que en el archivo docker-compose.yml podemos definir tanto el puerto de la base de datos, las credenciales de esta, el puerto en el que estará operativa la API y el puerto para conectarnos a phpMyAdmin.

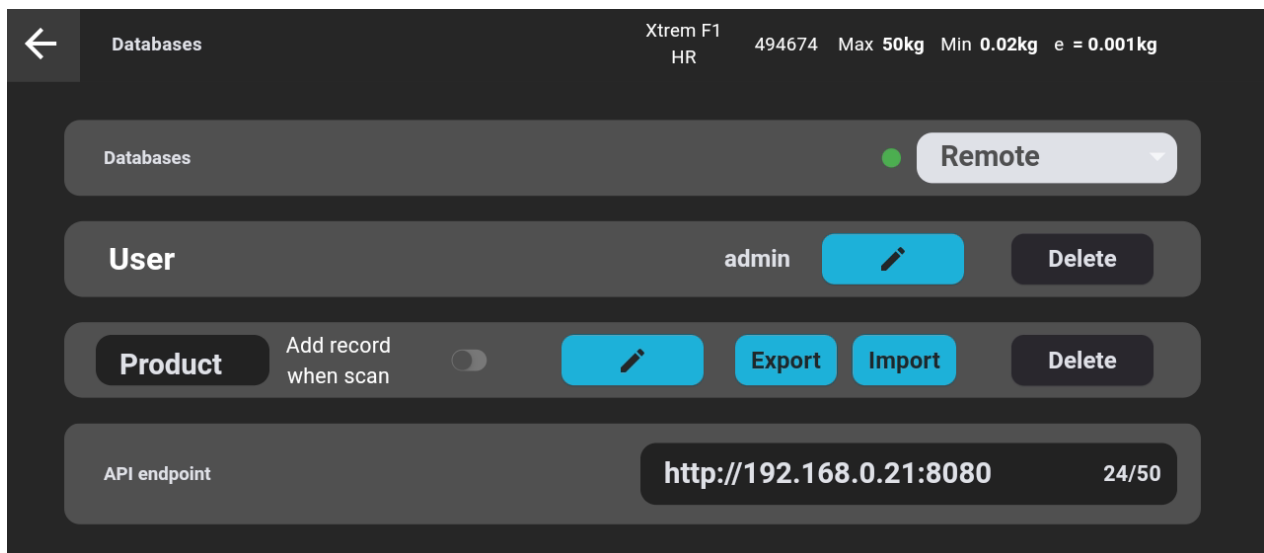
En el servicio de API vemos que la información que necesita la API para saber a qué base de datos debe conectar se la entrega un archivo **.env**.

```
# =====  
# Configuración de la base de datos  
# =====  
  
SPRING_DATASOURCE_URL=jdbc:mysql://db:3306/open_z8  
SPRING_DATASOURCE_USERNAME=apiuser  
SPRING_DATASOURCE_PASSWORD=Gram1231
```

En caso de que queramos actualizar el app.jar con la API, basta con cambiar el archivo y volver a ejecutar el archivo instalar.bat, instalar.sh en caso de usar Linux.

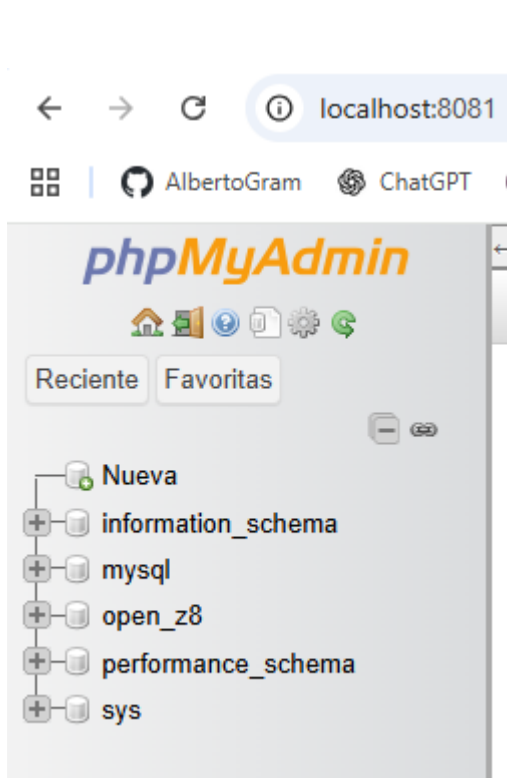
Finalmente, para hacer peticiones, las efectuaremos en la url:

**http://{dirección de nuestro server}:{puerto}**



Para conectarnos a phpMyAdmin, con la intención de interactuar con la base de datos del contenedor de MySQL, debemos hacerlo abriendo una ventana del navegador y dirigiéndonos al siguiente endpoint:

<http://localhost:8081>



## ERRORES

Si al iniciar Docker desktop se muestra el siguiente mensaje, es debido a que la virtualización no está activada en su terminal. Para solventar el error debe activarla.



### Virtualization support not detected

Docker Desktop failed to start because virtualisation support wasn't detected. Contact your IT admin to enable virtualization or check system requirements.

[View system requirements](#) ↗

Si se está usando la base de datos de Docker hay que recordar que el endpoint apunta a db no a localhost:

```
# =====  
# Configuración de la base de datos  
# =====  
  
SPRING_DATASOURCE_URL=jdbc:mysql://db:3306/open_z8  
SPRING_DATASOURCE_USERNAME=apiuser  
SPRING_DATASOURCE_PASSWORD=Gram1231
```

De lo contrario, si quisiéramos usar una base de datos que se encuentre en el mismo equipo en el que se encuentra Docker, debemos usar host.docker.internal, ya que localhost (127.0.0.1) lo entenderá como el contenedor no como nuestro host.

```
# =====  
# Configuración de la base de datos  
# =====  
  
SPRING_DATASOURCE_URL=jdbc:mysql://host.docker.internal:3306/open_z8  
SPRING_DATASOURCE_USERNAME=root  
# SPRING DATASOURCE PASSWORD=Gram1231
```

También es importante destacar que, si un servicio como WAMP u otro está activo y éste usa el puerto 3306, nosotros no podremos usar este para nuestra base de datos local con Docker, sería necesario acabar con el servicio o usar otro puerto editando el archivo docker-compose.



**Gram Precision S.L.**

Travesía Industrial, 11 · 08907 Hospitalet de Llobregat · Barcelona (Spain)

Tel. +34 93 300 33 32

[www.gram-group.com](http://www.gram-group.com)