

# Guía de Despliegue KuantumLibraryAPI

## 1. Prerrequisitos:

Asegúrate de tener el siguiente software instalado:

- **SDK de .NET 8:** Necesario para compilar la aplicación. Descargar desde <https://dotnet.microsoft.com/download/dotnet/8.0>.
- **Runtime de .NET 8:** Necesario para ejecutar la aplicación en el servidor de despliegue (ASP.NET Core Runtime).
- **Servidor MySQL/MariaDB:** Versión 8.0.23 o compatible.
- **Git (Opcional):** Para clonar el repositorio si el despliegue se hace desde el control de fuentes.

## 2. Configuración de la Base de Datos

### 2.1. Creación de la Base de Datos y Tablas

Ejecuta el script ScriptDB.sql para crear la base de datos library\_db y las tablas necesarias. El script se encuentra en: ...\\KuantumLibraryAPI\\DOCUMENTACION\\ScriptDB.sql.

```
-- Ejemplo de conexión a MySQL/MariaDB y ejecución del script (ajusta según tu cliente MySQL/MariaDB): -- mysql -u tu_usuario_db -p -- mysql> SOURCE ...\\KuantumLibraryAPI\\DOCUMENTACION\\ScriptDB.sql;
```

### 2.2. Configuración del String de Conexión

La API necesita saber cómo conectarse a la base de datos. Esto se configura en el archivo appsettings.json (o appsettings.Production.json para entornos de producción) dentro del directorio de la aplicación publicada. Modifica la sección ConnectionStrings:

```
{ "ConnectionStrings": { "DefaultConnection": "Server=tu_servidor_mysql;Port=3306;Database=library_db;Uid=tu_usuario_db;Pwd=tu_contraseña_db;" } // ... otras configuraciones ... }
```

**Nota:** Reemplaza tu\_servidor\_mysql, tu\_usuario\_db, y tu\_contraseña\_db con tus credenciales.

## 3. Instalación de Dependencias

Antes de compilar, asegúrate de restaurar todas las dependencias del proyecto. Navega al directorio raíz del proyecto (...\\KuantumLibraryAPI\\) en una terminal y ejecuta:

```
dotnet restore
```

## 4. Compilación y Publicación de la Aplicación

Para preparar la aplicación para el despliegue, compíllala y publícala. Desde el directorio raíz del proyecto:

### 4.1. Compilar (Opcional, ya que publish lo incluye):

```
dotnet clean (para este caso en particular, se usó este comando)
```

```
dotnet restore (para este caso en particular, se usó este comando)
```

```
dotnet build (para este caso en particular, se usó este comando)
```

```
dotnet build -c Release
```

### 4.2. Publicar la aplicación:

```
dotnet publish -c Release -o ./publish_output
```

Esto compilará la aplicación en modo Release y colocará los archivos necesarios en la carpeta publish\_output. Copia el contenido de esta carpeta al servidor donde se ejecutará la API.

## 5. Acceso a la API

Una vez que los archivos publicados estén en el servidor y la aplicación se haya iniciado (por ejemplo, ejecutando dotnet KuantumLibraryApi.dll desde la carpeta de publicación), puedes acceder a la API. Para explorar los endpoints y probar la API, utiliza la interfaz de Swagger UI (si está habilitada para el entorno de desarrollo o según tu configuración):

```
http://localhost:5000/swagger
```

Asegúrate de que el puerto (5000 en el ejemplo) coincida con el puerto en el que Kestrel está escuchando. Los endpoints de la API estarán disponibles bajo http://localhost:5000/api/documents (para este caso)

## 6. Verificación Post-Despliegue

Después de desplegar y ejecutar la API:

- **Revisa los logs de la aplicación:** Busca errores de inicio o de conexión a la base de datos. Los logs se mostrarán en la consola si ejecutas dotnet KuantumLibraryApi.dll directamente.
- **Accede a Swagger UI:** Abre http://localhost:5000/swagger en tu navegador para verificar que la interfaz carga y muestra los endpoints.
- **Prueba los endpoints:** Realiza algunas llamadas a la API a través de Swagger UI o una herramienta como Postman para asegurar que las operaciones CRUD funcionen correctamente con la base de datos.