# Setup Guide for StackFlowServer

Este documento describe los pasos iniciales para configurar el proyecto en **NetBeans**, la base de datos en **MariaDB** y la estructura de documentación recomendada.

## 1. Configuración en NetBeans

1. **Importar el proyecto**
   * Abre NetBeans y selecciona **File > Open Project**.
   * Navega hasta la carpeta raíz stackflow-server y haz clic en **Open Project**.
2. **Configurar el servidor WildFly**
   * Ve a **Services > Servers**.
   * Haz clic derecho y selecciona **Add Server**.
   * Elige **WildFly 36** (o la versión que uses) y apunta al directorio de instalación.
   * Configura el puerto de administración (default 9990) y el puerto HTTP (8090).
3. **Configurar el DataSource**
   * Despliega tu servidor en **Services**.
   * Haz clic derecho sobre WildFly > **View Domain** (o **Open Admin Console**).
   * En la consola de administración, ve a **Configuration > Subsystems > Datasources**.
   * Asegúrate de que el datasource StackflowDS esté presente y **Enabled**.
4. **Deploy Automático**
   * En el proyecto, haz clic derecho en la raíz > **Properties > Run**.
   * Marca **Deploy on Save** para que cada cambio se despliegue automáticamente.

## 2. Configuración de la Base de Datos (MariaDB)

1. **Crear la base de datos**

* CREATE DATABASE stackflowdb CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;  
  USE stackflowdb;

1. **Ejecutar script de esquema**
   * Abre tu cliente SQL (DBeaver, MySQL Workbench, consola).
   * Ejecuta el contenido de db/db\_schema.sql (incluye todos los CREATE TABLE).
2. **Verificar tablas**

* SHOW TABLES;
* Deberías ver las 27 tablas definidas (users, signals, pipelines, pipeline\_versions, etc.).

1. **Datasource en WildFly**
   * Asegúrate de que en standalone-full.xml:

* <datasource jndi-name="java:/jdbc/StackflowDS" pool-name="StackflowDS" ...>  
   <connection-url>jdbc:mariadb://localhost:3307/stackflowdb</connection-url>  
   <driver>mariadb</driver>  
   <security user-name="root" password="root"/>  
  </datasource>

## 3. Estructura de Documentación

Coloca los archivos de documentación en la carpeta docs/:

* **README.md**: Visión general del proyecto y cómo empezar.
* **docs/SETUP\_GUIDE.md**: Esta guía de configuración.
* **docs/DATABASE\_SCHEMA.md**: Detalle de todas las tablas y relaciones.
* **docs/NETBEANS\_SETUP.md** (opcional si se desea separado): Detallado de NetBeans.
* **docs/DB\_SETUP.md** (opcional si se desea separado): Detallado de la base de datos.
* **docs/CONTRIBUTING.md**: Normas de contribución y estilo de código.
* **docs/DEPLOYMENT.md**: Guía de deploy en producción.

## 4. Lista de tablas, funcionalidad y relaciones

1. **users**
   * **Función:** Usuarios del sistema.
   * **PK:** id
   * **Campos clave:** username, email, password\_hash
   * **Relaciones:**
     + 1:N → user\_roles
     + 1:N → user\_settings
     + 1:N → projects (como owner\_id)
     + 1:N → project\_users
     + 1:N → signals (como user\_id)
     + 1:N → models (como user\_id)
     + 1:N → pipelines (como user\_id)
     + 1:N → history
     + 1:N → memory\_slots
     + 1:N → db\_connections
     + 1:N → job\_queue
     + 1:N → schedules
2. **roles**
   * **Función:** Roles predefinidos (e.g. ADMIN, USER).
   * **PK:** id;
   * **Relaciones:**
     + N:M → user\_roles
3. **user\_roles**
   * **Función:** Asigna roles a usuarios.
   * **PK Compuesta:** (user\_id, role\_id)
   * **FK:** user\_id → users, role\_id → roles
4. **user\_settings**
   * **Función:** Preferencias de cada usuario.
   * **PK:** id
   * **FK:** user\_id → users
5. **projects**
   * **Función:** Proyectos de trabajo.
   * **PK:** id
   * **FK:** owner\_id → users
   * **Relaciones:** 1:N → project\_users
6. **project\_users**
   * **Función:** Usuarios asignados a proyectos.
   * **PK Compuesta:** (project\_id, user\_id)
   * **FK:** project\_id → projects, user\_id → users
7. **datasets**
   * **Función:** Datasets externos y sus metadatos.
   * **PK:** id
8. **signals**
   * **Función:** Señales procesadas o cargadas.
   * **PK:** id
   * **FK:** project\_id → projects, user\_id → users
   * **Relaciones:** 1:N → signal\_data, N:M → signal\_group\_signals
9. **signal\_data**
   * **Función:** Datos crudos o serializados de señales.
   * **PK:** id
   * **FK:** signal\_id → signals
10. **signal\_groups**
    * **Función:** Grupos/colecciones de señales.
    * **PK:** id
    * **FK:** user\_id → users
    * **Relaciones:** N:M → signal\_group\_signals
11. **signal\_group\_signals**
    * **Función:** Asocia señales a grupos.
    * **PK Compuesta:** (signal\_group\_id, signal\_id)
    * **FK:** signal\_group\_id → signal\_groups, signal\_id → signals
12. **models**
    * **Función:** Modelos de ML/NN.
    * **PK:** id
    * **FK:** user\_id → users
    * **Relaciones:** 1:N → model\_files
13. **model\_files**
    * **Función:** Archivos binarios de cada modelo.
    * **PK:** id
    * **FK:** model\_id → models
14. **pipelines**
    * **Función:** Definición de workflows de procesamiento.
    * **PK:** id
    * **FK:** user\_id → users
    * **Relaciones:** 1:N → pipeline\_steps, 1:1 → pipeline\_current\_version, 1:N → pipeline\_versions
15. **pipeline\_steps**
    * **Función:** Pasos individuales dentro de un pipeline.
    * **PK:** id
    * **FK:** pipeline\_id → pipelines
16. **pipeline\_versions**
    * **Función:** Versionado limitado de las últimas N operaciones de cada pipeline.
    * **PK:** id
    * **FK:** pipeline\_id → pipelines
17. **pipeline\_current\_version**
    * **Función:** Indica la versión activa de cada pipeline.
    * **PK y FK:** pipeline\_id → pipelines
18. **history**
    * **Función:** Bitácora de cambios genéricos.
    * **PK:** id
    * **FK:** user\_id → users
19. **memory\_slots**
    * **Función:** Almacenamiento de “memoria” persistente (embeddings, caches).
    * **PK:** id
    * **FK:** user\_id → users
20. **config**
    * **Función:** Configuración global de la plataforma (clave/valor).
    * **PK:** id
21. **scenarios**
    * **Función:** Definición de snapshots de estado (señales, modelos, pipelines, datasets, config).
    * **PK:** id
    * **FK:** creator\_id → users
    * **Relaciones:** 1:N → scenario\_entities, 1:N → scenario\_runs
22. **scenario\_entities**
    * **Función:** Asocia entidades genéricas a un escenario.
    * **PK:** id
    * **FK:** scenario\_id → scenarios
23. **scenario\_runs**
    * **Función:** Historial de ejecuciones de cada escenario.
    * **PK:** id
    * **FK:** scenario\_id → scenarios, user\_id → users
24. **job\_queue**
    * **Función:** Cola de tareas batch o post-process.
    * **PK:** id
    * **FK:** user\_id → users
    * **Relaciones:** 1:N → post\_process\_tasks, 1:N → schedules
25. **post\_process\_tasks**
    * **Función:** Tareas de post-procesamiento vinculadas a jobs.
    * **PK:** id
    * **FK:** job\_id → job\_queue
26. **db\_connections**
    * **Función:** Conexiones a bases de datos externas.
    * **PK:** id
    * **FK:** user\_id → users
27. **schedules**
    * **Función:** Programación (cron) de jobs.
    * **PK:** id
    * **FK:** job\_id → job\_queue, user\_id → users