Manual de Instalación Completo - Confitería Santana Web

# i. Requerimientos previos a la instalación

Para el correcto funcionamiento del sistema Confitería Santana Web, es necesario contar con una serie de requisitos previos, tanto a nivel de servidor como de servicios adicionales.

## 1. Servidor web y configuración necesaria

- Sistema operativo recomendado: Linux (Debian, Ubuntu o similar)  
- Servidor de aplicaciones: Open Liberty v5 o superior  
- Java: JDK 17 o superior  
- Angular: No es necesario instalar Angular en el servidor si se compila previamente; basta con servir los archivos estáticos.  
- Servidor web estático (opcional): Nginx o Apache2, para servir el frontend compilado

## 2. Servidor de base de datos y configuración necesaria

- MySQL: Versión 8 o superior  
- Usuario y base de datos:  
 - Crear una base de datos con nombre por ejemplo confiteria\_db  
 - Crear un usuario con permisos de lectura/escritura  
 - Configurar el url, user y password en el archivo de configuración del backend (persistence.xml o propiedades externas)

## 3. Servicios o librerías necesarios

- Librerías backend (ya incluidas mediante Maven o Gradle en el proyecto):  
 - Jakarta EE / MicroProfile APIs  
 - Hibernate ORM  
 - JAX-RS  
 - JSON-P o JSON-B  
- Librerías frontend (ya incluidas mediante package.json):  
 - Angular  
 - RxJS  
 - Tailwind CSS + DaisyUI

## 4. Otros

- Acceso SSH al servidor Linux para subir archivos y configurar servicios  
- Acceso a puertos 8080 (backend) y 80 o 4200 (frontend, si se lanza directamente)  
- Editor de texto o consola para modificar configuraciones (ej. Vim, nano o VS Code vía SFTP)  
- Compilador de Angular (Node.js + npm) si se desea compilar directamente en el servidor (opcional)

# ii. Instalación

## 1. Subida de la aplicación al servidor

Conéctate al servidor mediante SSH:

ssh usuario@IP\_DEL\_SERVIDOR

Crea la estructura de carpetas:

mkdir -p ~/confiteria/backend  
mkdir -p ~/confiteria/frontend  
mkdir -p ~/confiteria/db

Sube el archivo backend.war desde tu equipo local:

scp ruta/al/backend.war usuario@IP\_DEL\_SERVIDOR:~/confiteria/backend/

Sube el frontend compilado (carpeta dist):

scp -r ruta/a/dist/confiteria-santana/ usuario@IP\_DEL\_SERVIDOR:~/confiteria/frontend/

## 2. Configuración del servidor

Instala y configura MySQL:

sudo apt update  
sudo apt install mysql-server  
sudo mysql\_secure\_installation

Accede a MySQL:

sudo mysql -u root -p

Dentro de MySQL, crea la base de datos y el usuario:

CREATE DATABASE confiteria\_db;  
CREATE USER 'confiteria\_user'@'localhost' IDENTIFIED BY 'claveSegura123';  
GRANT ALL PRIVILEGES ON confiteria\_db.\* TO 'confiteria\_user'@'localhost';  
FLUSH PRIVILEGES;  
EXIT;

Descarga e instala Open Liberty:

wget https://public.dhe.ibm.com/ibmdl/export/pub/software/websphere/wasdev/downloads/wlp/22.0.0.12/wlp-kernel-22.0.0.12.zip  
unzip wlp-kernel-22.0.0.12.zip -d ~/openliberty  
cd ~/openliberty/wlp/bin  
./server create confiteriaServer

Copia el archivo backend.war a la carpeta dropins:

cp ~/confiteria/backend/backend.war ~/openliberty/wlp/usr/servers/confiteriaServer/dropins/

Edita el archivo server.xml:

nano ~/openliberty/wlp/usr/servers/confiteriaServer/server.xml

Ejemplo de configuración mínima en server.xml:

<server>  
 <featureManager>  
 <feature>jaxrs-2.1</feature>  
 <feature>jsonb-1.0</feature>  
 <feature>cdi-2.0</feature>  
 <feature>jpa-2.2</feature>  
 </featureManager>  
  
 <httpEndpoint id="defaultHttpEndpoint" host="\*" httpPort="8080" />  
  
 <dataSource id="MySQLDataSource" jndiName="jdbc/confiteriaDS">  
 <jdbcDriver libraryRef="MySQLLib"/>  
 <properties databaseName="confiteria\_db" serverName="localhost" portNumber="3306"  
 user="confiteria\_user" password="claveSegura123"/>  
 </dataSource>  
  
 <library id="MySQLLib">  
 <fileset dir="/home/usuario/mysql-connector-j-8.0.33" includes="mysql-connector-java-8.0.33.jar"/>  
 </library>  
  
 <applicationMonitor updateTrigger="mbean"/>  
 <application type="war" location="backend.war" name="confiteriaApp"/>  
</server>

Inicia el servidor:

./server start confiteriaServer

## 3. Servir el frontend con Nginx

Instala Nginx:

sudo apt install nginx

Elimina la página por defecto y copia los archivos del frontend:

sudo rm /var/www/html/index.nginx-debian.html  
sudo cp -r ~/confiteria/frontend/\* /var/www/html/  
sudo systemctl restart nginx

## 4. Conexión entre backend y base de datos

Edita el archivo persistence.xml y configura la conexión:

<property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/confiteria\_db"/>  
<property name="javax.persistence.jdbc.user" value="confiteria\_user"/>  
<property name="javax.persistence.jdbc.password" value="claveSegura123"/>

Reinicia el servidor para aplicar cambios:

./server restart confiteriaServer

## 5. Configuración de firewall y dominio

Abre los puertos necesarios en el firewall:

sudo ufw allow OpenSSH  
sudo ufw allow 80  
sudo ufw allow 8080  
sudo ufw enable

Si tienes dominio, apunta los registros DNS A al servidor y edita el archivo de configuración de Nginx:

sudo nano /etc/nginx/sites-available/default

Ejemplo de configuración para un dominio:

server {  
 listen 80;  
 server\_name www.confiteriasantana.com;  
  
 root /var/www/html;  
 index index.html;  
  
 location / {  
 try\_files $uri $uri/ =404;  
 }  
}

Reinicia Nginx:

sudo systemctl restart nginx

## 6. Activar HTTPS con Let's Encrypt (opcional)

Instala Certbot y activa HTTPS:

sudo apt install certbot python3-certbot-nginx  
sudo certbot --nginx