

# Procedimiento de Implementación del Software

Este documento proporciona una guía técnica y detallada para el despliegue del software en el entorno de producción. Se describen los requisitos, la configuración del entorno, los pasos de instalación y despliegue tanto para el back-end como para el front-end, los scripts de automatización, las validaciones posteriores y la documentación final. La guía está diseñada para ser seguida paso a paso por el equipo técnico.

## 1. Descripción General

El propósito de este documento es asegurar que la implementación del software se realice de forma segura, eficiente y de acuerdo a las mejores prácticas. Se incluyen:

- Configuraciones y requisitos previos.
- Instrucciones de instalación y despliegue.
- Ejemplos de scripts para automatización.
- Validaciones post-despliegue.
- Ejemplos de pruebas (HTML y Postman).

## 2. Requisitos Previos y Configuración del Entorno

### 2.1 Requisitos de Hardware y Software

- **Sistema Operativo:** Linux (preferido) o Windows Server.
- **Servidor de Aplicaciones:** Apache Tomcat 9 o superior.
- **Base de Datos:** MySQL 8.0 o superior.
- **JDK:** Java Development Kit 17 o superior.
- **Recursos de Red:** Asegúrese de tener acceso a los puertos necesarios (por ejemplo, 8080 para Tomcat y 3306 para MySQL) y que el firewall esté configurado correctamente.

### 2.2 Permisos y Accesos

- **Acceso Administrativo:** Permisos de administrador en el servidor para instalar software y modificar configuraciones.

- **Acceso a la Base de Datos:** Credenciales válidas para conectar y modificar la base de datos MySQL.

## 2.3 Configuración Inicial del Entorno

- **Actualización del Sistema:** Aplique todos los parches y actualizaciones de seguridad.
- **Instalación de Dependencias:**
  - Instale JDK 17.
  - Configure Maven o Gradle según corresponda.
  - Instale Apache Tomcat y MySQL, si no se encuentran instalados.
- **Variables de Entorno:**
  - Establezca JAVA\_HOME, MAVEN\_HOME y CATALINA\_HOME (en el caso de Tomcat) según la estructura de su sistema.

## 3. Instalación y Despliegue del Software

El proceso de despliegue se realiza en dos partes: Back-end y Front-end.

### 3.1 Despliegue del Back-end

#### 3.1.1 Ubicación y Compilación

- **Código Fuente:** C:\Users\juand\Desktop\GOAPPV2
- **Acciones:**
  - Abra una terminal o línea de comandos.
  - Navegue hasta el directorio del proyecto.
  - Ejecute el comando de compilación y empaquetado (por ejemplo, mvn clean package o gradle build).
  - Verifique que el archivo .war se haya generado en la carpeta target o build/libs.

#### 3.1.2 Configuración y Despliegue en Tomcat

- **Pasos:**
  - Copie el archivo .war generado al directorio webapps de Apache Tomcat.
  - Edite el archivo application.properties o application.yml para configurar la conexión a la base de datos MySQL (host, puerto, usuario, contraseña y nombre de la base de datos).
  - Reinicie Tomcat para desplegar la nueva aplicación:
    - **En Linux:** Utilice los scripts shutdown.sh y startup.sh.
    - **En Windows:** Utilice shutdown.bat y startup.bat.
  - Verifique el despliegue revisando los logs (catalina.out o equivalente).

## 3.2 Despliegue del Front-end

### 3.2.1 Ubicación y Preparación

- **Código Fuente:** C:\Users\juand\Desktop\GOAPPV2-Front
- **Acciones:**
  - Navegue hasta el directorio del front-end.
  - Compile o procese los archivos HTML, CSS y JavaScript utilizando herramientas de bundling (por ejemplo, Webpack, Gulp o similar).
  - Genere los archivos estáticos que serán desplegados.

### 3.2.2 Publicación

- **Opciones de Despliegue:**
  - Despliegue los archivos estáticos en un servidor web dedicado (Nginx, Apache HTTPD) o en el mismo servidor donde se aloja el back-end.
  - Configure las rutas y las políticas CORS para asegurar la comunicación entre el front-end y el back-end.

## 4.2 Script para Entornos Windows (Batch)

## 4. Automatización del Despliegue: Scripts

### 4.1 Script para Entornos Unix (Bash)

```
#!/bin/bash
echo "Iniciando el despliegue del Back-end..."
export TOMCAT_HOME=/opt/tomcat
APP_NAME=mi_aplicacion

# Copiar el archivo WAR
cp /ruta/del/proyecto/GOAPPV2/target/$APP_NAME.war $TOMCAT_HOME/webapps/
echo "Archivo .war copiado a $TOMCAT_HOME/webapps/"

# Reiniciar Tomcat
```

```
$TOMCAT_HOME/bin/shutdown.sh  
sleep 5  
$TOMCAT_HOME/bin/startup.sh  
echo "Tomcat reiniciado exitosamente."  
  
# Mostrar los últimos 100 registros del log  
tail -n 100 $TOMCAT_HOME/logs/catalina.out
```

## 5. Validaciones Posteriores al Despliegue

### 5.1 Conexión a la Base de Datos

- **Acciones:**
  - Ejecutar consultas de prueba desde la aplicación.
  - Revisar logs de MySQL para confirmar la ejecución correcta de scripts y migraciones.

### 5.2 Pruebas Funcionales

- **Verificaciones:**
  - Realizar pruebas de autenticación (login/logout).
  - Validar la generación y verificación de códigos diarios.
  - Simular el registro de jornadas y consultar reportes para asegurar la integridad del flujo de negocio.

### 5.3 Pruebas de Seguridad

- **Revisar:**
  - Configuración del firewall y permisos de red.
  - Gestión de sesiones y protección contra ataques (inyección SQL, XSS, CSRF).
  - Validación de certificados SSL si se usa HTTPS.

### 5.4 Pruebas de Rendimiento

- **Acciones:**
  - Ejecute pruebas de carga simulando múltiples usuarios concurrentes.
  - Monitorear el rendimiento del servidor (CPU, memoria, I/O) usando herramientas como JMeter o similares.

## 6. Documentación y Reporte Final

Al concluir el proceso de despliegue, se debe generar un informe final que incluya:

- **Resumen del Despliegue:** Descripción detallada de cada paso realizado.
- **Resultados de las Validaciones:** Datos y métricas obtenidas en las pruebas.
- **Registro de Incidencias:** Lista de errores identificados y las acciones correctivas aplicadas.
- **Recomendaciones:** Sugerencias para futuras mejoras y optimización del proceso.
- **Contacto de Soporte:** Información de contacto del equipo responsable del mantenimiento.

## 7. Ejemplos de URLs de Prueba y Configuraciones en HTML/Postman

### 7.1 Endpoints de la API Person (Controlador de Ejemplo)

Los siguientes endpoints se utilizan para gestionar las operaciones CRUD en la entidad *Person*:

- **Crear Persona (POST):**
  - **URL:** <http://localhost:8080/api/person>
  - **Método:** POST
  - **Headers:** Content-Type: application/json
  - **Cuerpo (ejemplo):**

```
{  
  "name": "Juan Perez",  
  "age": 30,  
  "email": "juan.perez@example.com"  
}
```

- **Actualizar Persona (PUT):**
  - **URL:** <http://localhost:8080/api/person/{id}>
  - **Método:** PUT
  - **Cuerpo:** Similar al de creación, con los campos actualizados.
- **Obtener Todas las Personas (GET):**
  - **URL:** <http://localhost:8080/api/person>
  - **Método:** GET
- **Obtener Persona por ID (GET):**

- **URL:** <http://localhost:8080/api/person/{id}>
- **Método:** GET
- **Eliminar Persona (DELETE):**
  - **URL:** <http://localhost:8080/api/person/{id}>
  - **Método:** DELETE

## 7.2 Prueba de API en HTML (con AJAX)

Un ejemplo de archivo HTML para probar el endpoint GET:

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <title>Prueba de API Person</title>

  <script>

    function obtenerPersonas() {

      fetch('http://localhost:8080/api/person')

        .then(response => response.json())

        .then(data => {

          console.log(data);

          document.getElementById('resultado').innerText = JSON.stringify(data, null, 2);

        })

        .catch(error => console.error('Error:', error));

    }

  </script>

</head>

<body>

  <h1>Prueba de API Person</h1>
```

```
<button onclick="obtenerPersonas()">Obtener Personas</button>

<pre id="resultado"></pre>

</body>

</html>
```

## 7.3 Configuración de Pruebas en Postman

Cree una colección en Postman con las siguientes solicitudes:

- **Crear Persona:**
  - **Método:** POST
  - **URL:** <http://localhost:8080/api/person>
  - **Body:**

```
{
  "name": "Juan Perez",
  "age": 30,
  "email": "juan.perez@example.com"
}
```
- **Actualizar Persona:**
  - **Método:** PUT
  - **URL:** <http://localhost:8080/api/person/1> (reemplazar 1 con el ID correspondiente)
  - **Body:** JSON con los datos actualizados.
- **Obtener Todas las Personas:**
  - **Método:** GET
  - **URL:** <http://localhost:8080/api/person>
- **Obtener Persona por ID:**
  - **Método:** GET
  - **URL:** <http://localhost:8080/api/person/1> (reemplazar 1 con el ID correspondiente)
- **Eliminar Persona:**
  - **Método:** DELETE
  - **URL:** <http://localhost:8080/api/person/1> (reemplazar 1 con el ID correspondiente)

Este documento estructurado y técnico se integra al proyecto como parte del procedimiento de implementación del software. Cada sección está diseñada para ser seguida paso a paso, facilitando la implementación, validación y mantenimiento del sistema en el entorno de producción.