# Explicación Detallada del Flujo CI/CD para Despliegue en WebLogic (Actualizado)

Este documento describe el flujo completo de Integración Continua (CI) y Despliegue Continuo (CD) que hemos diseñado, comenzando por la compilación y empaquetado de tu aplicación "Experto Contable" hasta su despliegue en un servidor WebLogic alojado en Azure. Esta versión refleja una estructura de 3 pipelines principales.

## Diagrama Simplificado del Flujo

1. Disparador (Push/Manual en el repo de la aplicación)

|

v

2. Pipeline Orquestador Principal (main\_ci\_cd\_orchestrator\_es\_v2.yml - en el repo de la app)

|

|--- Llama a --> 3. Pipeline de CI Reutilizable (deploy-container-app.yml - el de tu colega)

| | - Compila código Java

| | - Obtiene versión del pom.xml

| | - Crea un archivo .zip (ej. PR-123.zip) conteniendo el .ear

| | - Sube el .zip a JFrog Artifactory

| | - (Importante) Genera Salidas: nombre del .zip, ruta en JFrog, nombre original del .ear

|

|--- (El Orquestador mapea salidas de CI y entorno para CD)

|

|--- Llama a --> 4. Pipeline de CD Reutilizable (generic\_jfrog\_to\_weblogic\_cd\_es\_v1.yml)

| - Recibe información del artefacto en JFrog y detalles del entorno/WebLogic

| - Descarga el .zip desde JFrog

| - Extrae el archivo .ear del .zip

| - Deriva el nombre de la aplicación para WebLogic (ej. ExpertoContable\_v3.4)

| - Copia el .ear al servidor WebLogic (Azure VM) vía SCP

| - Se conecta al servidor WebLogic (Azure VM) vía SSH

| - Ejecuta `weblogic.Deployer -undeploy` (si aplica)

| - Ejecuta `weblogic.Deployer -deploy` con el .ear

| - Reporta el estado

## 1. Pipeline de CI Reutilizable (Integración Continua - deploy-container-app.yml)

Este es el pipeline reutilizable desarrollado (JuanC.). Su principal responsabilidad es tomar el código fuente, compilarlo y empaquetarlo, dejando un artefacto listo para el despliegue en JFrog.

* **Propósito Principal**:
  1. **Compilar el código Java**.
  2. **Gestionar Dependencias** (opcionalmente desde JFrog).
  3. **Obtener Versión y Nombre del Artefacto** del pom.xml.
  4. **Empaquetar para JFrog**: Comprime el artefacto compilado (ej. .ear) dentro de un archivo .zip.
  5. **Subir a JFrog**: Sube este archivo .zip a JFrog Artifactory.
* **Entradas Clave que Recibe**: Parámetros como entorno de CI, comando de compilación, ruta al pom.xml, etc.
* **Salidas (Outputs) Cruciales para el Orquestador**:
  1. ci\_original\_artifact\_filename\_output: El nombre del artefacto original (ej. ExpertoContable-3.4.ear).
  2. ci\_jfrog\_zip\_name\_output: El nombre del archivo .zip subido a JFrog (ej. PR-123.zip).
  3. ci\_jfrog\_repo\_path\_output: La ruta del repositorio en JFrog donde se guardó el .zip (ej. ExpertoContable-int).
  4. *Nota: Los nombres exactos de estas salidas deben coincidir entre lo que el CI genera y lo que el Orquestador espera.*

## 2. Pipeline Orquestador Principal (main\_ci\_cd\_orchestrator\_es\_v2.yml)

Este es el pipeline principal que reside en el repositorio de la aplicación ("Experto Contable") y que inicia todo el proceso.

* **Propósito Principal**:
  1. Ser el punto de entrada del flujo completo (activado por push o manualmente).
  2. Llamar al Pipeline de CI Reutilizable, pasándole los parámetros necesarios para la compilación específica de la aplicación.
  3. Recibir las salidas del CI (información del artefacto en JFrog).
  4. Mapear/transformar estas salidas y el entorno de CI a las entradas que espera el Pipeline de CD Reutilizable. Esto incluye determinar las URLs de WebLogic, IPs de VMs, y contraseñas correctas según el entorno.
  5. Llamar al Pipeline de CD Reutilizable, pasándole toda la información procesada.
* **Disparadores (on)**: push a ramas específicas, workflow\_dispatch para ejecución manual.
* **Jobs (Trabajos)**:
  1. **call\_ci\_build**:
     + Invoca al pipeline de CI reutilizable (deploy-container-app.yml).
     + Espera a que el CI complete la compilación y subida a JFrog.
  2. **call\_cd\_reusable**:
     + Depende de que call\_ci\_build haya terminado exitosamente (needs: call\_ci\_build).
     + Invoca al pipeline de CD reutilizable (generic\_jfrog\_to\_weblogic\_cd\_es\_v1.yml).
     + Le pasa como entradas:
       - La información del artefacto en JFrog (obtenida de las salidas del CI).
       - El nombre del entorno de CD (derivado del entorno de CI).
       - Las configuraciones específicas del entorno para WebLogic (URLs, targets, host SSH).
     + Pasa los secretos necesarios (credenciales de JFrog, SSH, WebLogic) al CD reutilizable.

### 3. Pipeline de CD Reutilizable (generic\_jfrog\_to\_weblogic\_cd\_es\_v1.yml)

Este es el pipeline que encapsula toda la lógica de Despliegue Continuo, desde la obtención del artefacto de JFrog hasta su instalación en WebLogic. Está diseñado para ser genérico y poder ser utilizado por diferentes aplicaciones.

* **Propósito Principal**:
  1. Ser llamado por un pipeline orquestador (workflow\_call).
  2. Descargar el archivo .zip (que contiene el .ear) desde JFrog Artifactory, usando la información provista por el orquestador.
  3. Extraer el archivo desplegable (ej. .ear) del .zip.
  4. Derivar el nombre de la aplicación en el formato que WebLogic espera (ej. de ExpertoContable-3.4.ear a ExpertoContable\_v3.4).
  5. Copiar el archivo .ear extraído al servidor WebLogic (alojado en una VM de Azure) utilizando SCP (Secure Copy Protocol).
  6. Conectarse al servidor WebLogic vía SSH.
  7. Ejecutar el comando weblogic.Deployer -undeploy para remover cualquier versión anterior de la aplicación.
  8. Ejecutar el comando weblogic.Deployer -deploy para instalar la nueva versión del .ear.
* **Entradas Clave que Recibe (del orquestador)**:
  1. jfrog\_zip\_artifact\_name, jfrog\_zip\_repository\_path, jfrog\_base\_url, original\_artifact\_filename\_inside\_zip: Para localizar y descargar el artefacto de JFrog.
  2. cd\_environment\_name: Nombre del entorno de CD.
  3. weblogic\_target\_ssh\_host, weblogic\_admin\_url, weblogic\_targets\_cluster: Detalles específicos del servidor WebLogic para el entorno destino.
* **Secretos que Utiliza**:
  1. Credenciales de JFrog para la descarga (JFROG\_USER, JFROG\_PASSWORD).
  2. Credenciales SSH para el acceso a la VM de Azure (AZURE\_VM\_SSH\_USER, AZURE\_VM\_SSH\_PRIVATE\_KEY).
  3. Contraseña del usuario deployer de WebLogic (WEBLOGIC\_DEPLOYER\_PASSWORD).
* **Salidas (Outputs)**: Puede generar salidas como el nombre de la aplicación desplegada.

**Flujo de Datos y Entornos**

* El **entorno de CI** (ej. integration) se define en el orquestador y se pasa al pipeline de CI.
* El pipeline de CI usa este entorno para, por ejemplo, nombrar el repositorio en JFrog (ej. ExpertoContable-int).
* El **orquestador** recibe las salidas del CI y traduce el entorno de CI a un **entorno de CD** (ej. integration de CI se convierte en desarrollo para CD).
* El orquestador pasa este entorno de CD y la información del artefacto al pipeline de CD reutilizable.
* El pipeline de CD reutilizable utiliza el entorno de CD para seleccionar las configuraciones correctas (IPs de VMs, URLs de WebLogic, targets de clúster) y las credenciales apropiadas para realizar el despliegue.

Esta estructura de 3 pipelines (Orquestador en el repo de la app, CI reutilizable, y CD reutilizable) ofrece un buen balance entre simplicidad, separación de responsabilidades y alta reutilización.