

RED DE DISPOSITIVOS IOT EN PLANTAS INDUSTRIALES

Universidad Tecnológica Nacional - Facultad Regional Córdoba

Cátedra de Ingeniería y Calidad - Grupo 10 - 4k1



1 INTRODUCCIÓN

En las plantas industriales modernas, los dispositivos IoT son esenciales para monitorear variables críticas como temperatura y humedad.

El fabricante de estos sensores enfrenta el desafío de actualizar remotamente el firmware para corregir errores y mejorar algoritmos, en un entorno con conectividad restringida y alto riesgo de falla (possible brick del dispositivo).

Este trabajo presenta un plan de despliegue seguro, escalable y automatizado, orientado a minimizar el riesgo operativo, garantizar la integridad del software y mantener la continuidad del servicio.

2 MATERIALES Y MÉTODOS

Se propone desarrollar un Plan de Release aplicando un enfoque DevOps orientado a la automatización, trazabilidad y seguridad en el despliegue de firmware para dispositivos IoT.

Enfoque metodológico

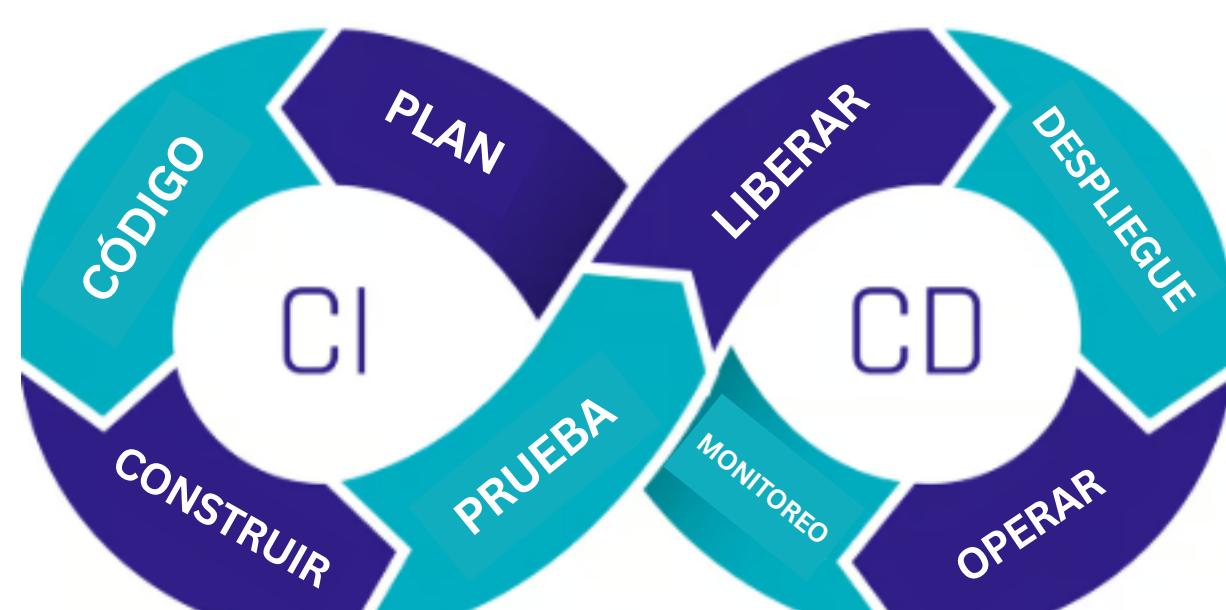
El proyecto plantea un proceso iterativo e incremental, que combine diseño, simulación y validación progresiva de los mecanismos de despliegue.

Cada componente del proceso será documentado y probado en entornos controlados antes de su aplicación real.

Herramientas y entorno de trabajo

Para implementar la propuesta se prevé el uso de:

- Git como sistema de control de versiones.
- Pipelines CI/CD (GitLab CI o GitHub Actions) para automatizar compilación, empaquetado y pruebas.
- Infraestructura como código (Terraform, Ansible) para reproducir entornos de manera estandarizada.
- Sistemas de monitoreo que permitan registrar métricas, telemetría y estado operativo de los sensores IoT

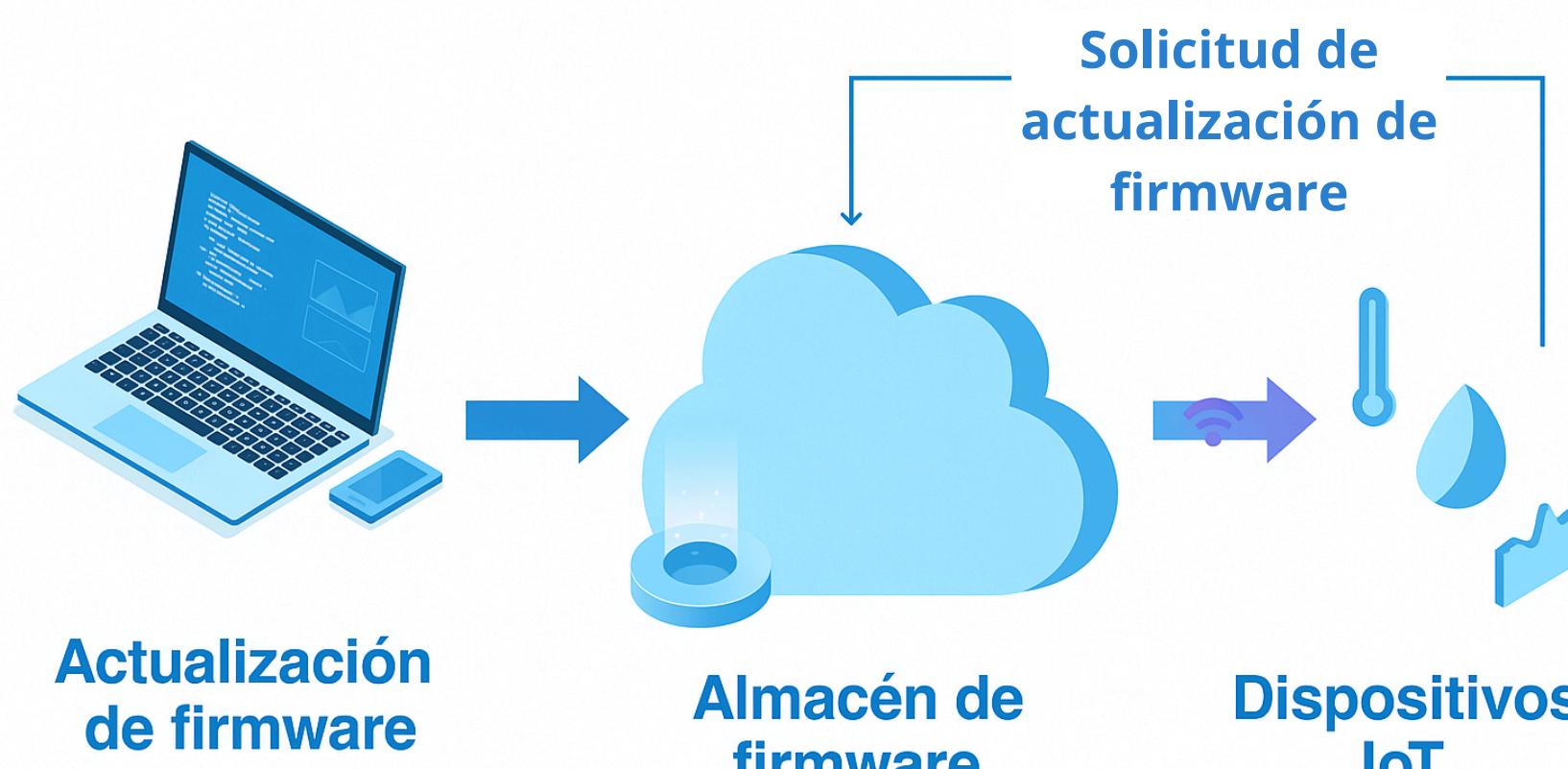


Relaciones importantes

El SCM gestiona la trazabilidad de versiones, asociando cada firmware a un hash, fecha y commit.

El enfoque DevOps permite automatizar compilación, pruebas, empaquetado y despliegue, asegurando consistencia y reducción de errores manuales.

Las prácticas continuas incluyen integración, entrega y monitoreo permanente, junto con testing automatizado en simuladores y hardware real.



AGRADECIMIENTOS

Se agradece la colaboración de la Universidad Tecnológica Nacional – Facultad Regional Córdoba, Cátedra de Ingeniería y Calidad de Software, y a todos los integrantes del Grupo 10 por su aporte en la investigación y desarrollo del plan de despliegue.

3 RESULTADOS

El principal resultado del trabajo es la definición estructurada del **Plan de Release** para el despliegue de firmware IoT.

1. Entorno de destino: sensores IoT distribuidos en plantas industriales y backend OTA alojado en la nube, con entornos Blue-Green para garantizar disponibilidad continua.

2. Tipo de despliegue: Over-The-Air (OTA) con validación Canary Release (5–10% de los dispositivos) y rollback automático A/B ante fallas.

3. Infraestructura requerida: servidor OTA, pipeline CI/CD, infraestructura como código, repositorio de firmware firmado y dashboards de monitoreo.

4. Artefacto a desplegar: firmware de los sensores IoT embebido versión vX.Y.Z, con posibilidad de revertir a vX.Y.(Z-1) mediante rollback. Identificado por hash y commit Git.

5. Procedimiento de despliegue

- Compilación y pruebas automáticas (CI).
- Validación en entorno staging (Green).
- Despliegue Canary y monitoreo durante 24–48 h.
- Despliegue masivo si no se detectan fallas.
- Rollback automático si se produce error.

6. Dependencias: conectividad mínima, certificados X.509 válidos y pipeline operativo.

7. Pre-requisitos: validación QA, firma digital del binario y aprobación del Release Manager.

8. Horario de despliegue: entre las 22:00 y 06:00 h, minimizando el impacto en la operación industrial.

9. Forma de despliegue (técnica): actualización OTA cifrada por TLS y backend mediante Blue-Green Deployment.

10. Estrategia de reducción de riesgos: canary Release, rollback automático, entornos paralelos y monitoreo continuo.

11. Monitoreo: dashboards centralizados con métricas de estado, logs y telemetría.

12. Pruebas post-despliegue: validación de lectura de sensores, conectividad y estabilidad operativa.

13. Comunicación y aprobaciones: coordinación entre QA, DevOps y Release Manager antes y después del despliegue.

14. Plan de contingencia (rollback): rollback automático, bloqueo de release fallido, análisis de incidentes y restauración del entorno estable.

15. Roles involucrados: Desarrollador Firmware, Ingeniero DevOps, Ingeniero QA

4 CONCLUSIONES

El Plan de Release diseñado permite ejecutar actualizaciones remotas de firmware de forma segura, automatizada y trazable, integrando prácticas de DevOps, CI/CD, QA y SCM.

Las estrategias combinadas de OTA, Canary Release, rollback automático y Blue-Green Deployment aseguran la continuidad operativa, reducen riesgos y facilitan la recuperación ante fallas.

Este enfoque promueve una mejora continua del ciclo de despliegue, optimiza recursos y establece un modelo reproducible para futuras implementaciones de IoT industrial.

REFERENCIAS

- Mender.io – OTA Updates Framework
Eclipse Hawkbit Project
AWS IoT Device Management
The DevOps Handbook – Kim et al., 2021