API ADDICTS DAYS 23

EL MAYOR EVENTO DE EXPERTOS EN APIS

Un viaje para conformar un entorno automatizado de integración y entrega continua de APIs.

Julio Cejas
julio.cejas@integrita.es
Integrita

ÍNDICE

- **Contexto**
  - Una directiva poderosa
  - El dilema entre la calidad y la velocidad
  - Opciones para eliminar el dilema
  - El secreto del alto rendimiento

- **Iniciando el viaje**
  - CICD y la automatización
  - Estrategia y Diseño
  - Plataforma de Activos
  - Plataforma de Delivery
  - Protocolo
  - Observabilidad y Mejora Continua

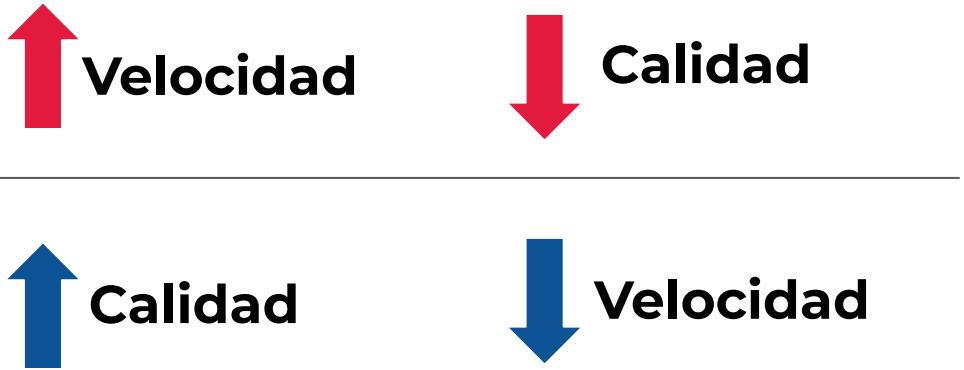
- **Demostración**
- **Lecciones Aprendidas**

Una directiva poderosa

Entregar software de calidad y más rápido, para ayudar a personas y organizaciones a prosperar.

Una directiva poderosa | El Dilema

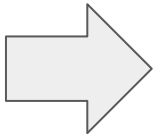
Cómo hacerlo mejor y más rápido?



El dilema es originado en la Ingeniería del **Detalle**

Una directiva poderosa | Como eliminar el dilema

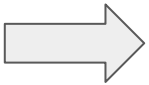
Como eliminar el dilema entre la calidad y velocidad ?



Las organizaciones de **alto rendimiento** entregan apis de forma rápida y confiable. Su secreto radica en la **automatización sólida de procesos** para la creación, pruebas, empaquetado y entrega de apis.

Acciones Más Generalizadas

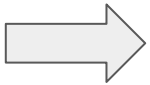
Por dónde comenzar?



1. Cultura colaborativa entre Dev y Ops (“DevOps”)
2. Herramientas de integración y despliegue continuo “**CI/CD”**
3. Equipo DevOps dedicado
4. Gestión y seguimiento
5. **Procesos y Gobierno**

Acciones con un ángulo diferente

Por dónde comenzar?



1. Evitar acumulación de deuda técnica.
2. Evitar acumulación de trabajo operativo.
3. Evitar acumulación de funciones.
4. Usar arquetipos o moldes.
5. Usar infraestructura como código.
6. Enfoque por procesos.

El secreto del Alto Rendimiento

Evitan la acumulación
Buscar la Simplicidad

Establecen un Marco de Principios y prácticas

Establecen un Marco de Directrices

Usan arquetipos o moldes

Modelan y automatizan procesos y pipelines

Adoptan Infraestructura como código

Iniciando el viaje....

CICD, su propósito

CI

Continuous Integration
Integración Continua

Automatizar y garantizar coherencia en la creación, pruebas y empaquetado de activos; facilitando la reducción del trabajo de implementación propenso a errores y su visibilidad temprana.

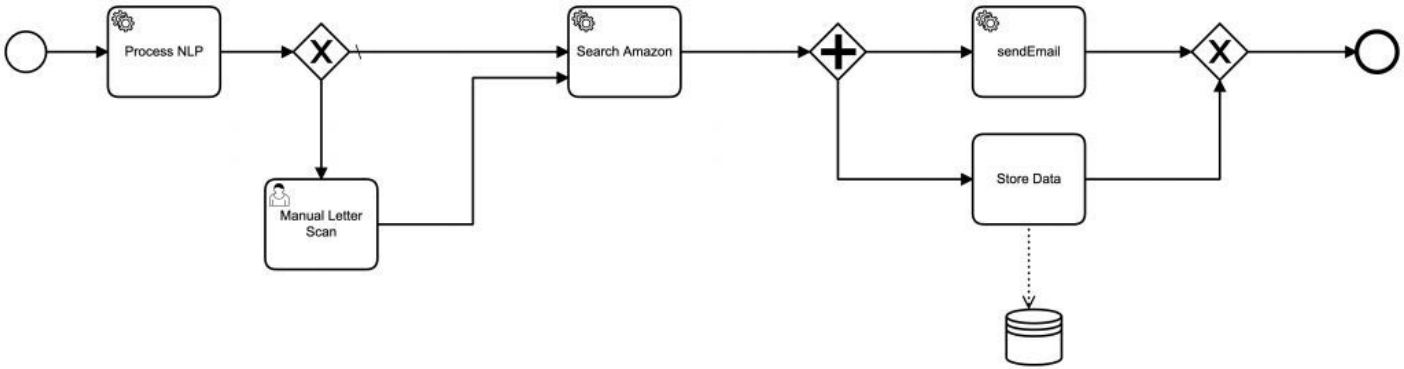
CD

Continuous Delivery
Despliegue Continuo

Automatizar la entrega de activos y cambios de código, a repositorios de artefactos y entrega a múltiples entornos de infraestructura, como pruebas y desarrollo.

CICD, su propósito

CICD = Automatización de procesos operacionales



etapa 01

Evaluación Actual	Verticales Estratégicas	Premisas de Onboarding	Procesos Operacionales
- Capacidades Actuales	- Bloques de Trabajo - Marco de Referencia	- Casos de uso - Directrices iniciales - Modelo de Procesos	- Compilación - Empaquetado - Deploy - otros...

etapa 01

Evaluación Actual

Verticales Estratégicas

Premisas de Onboarding

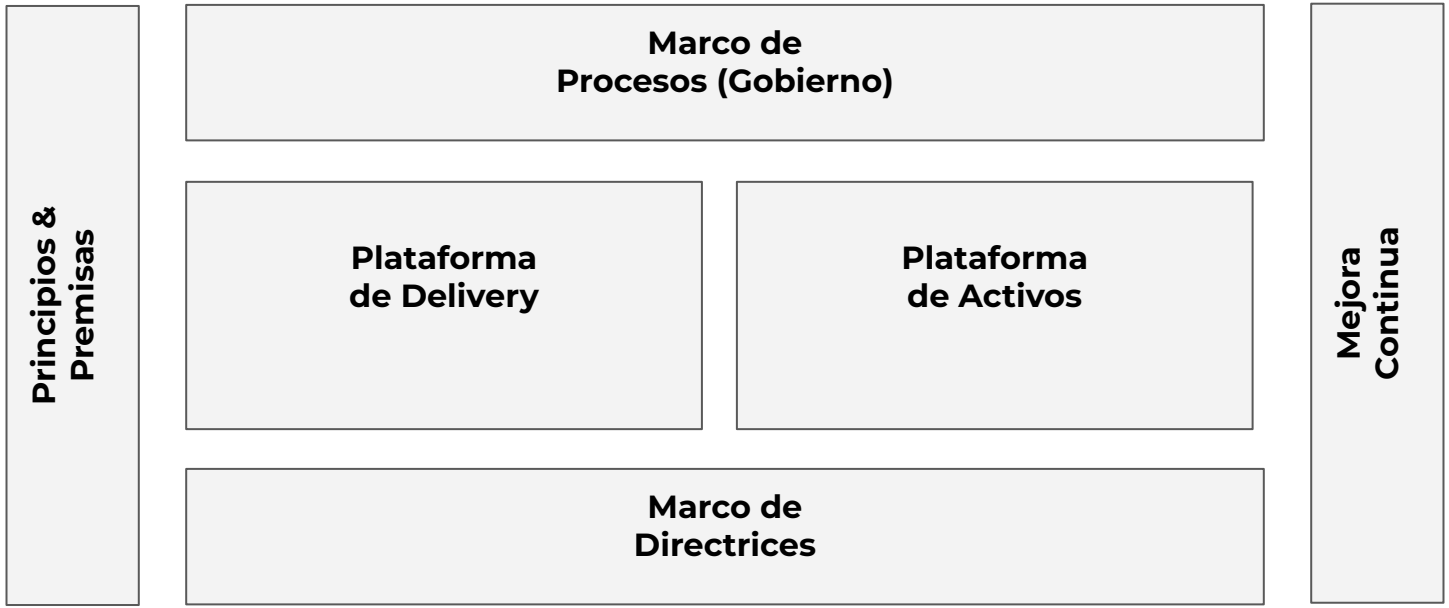
Diseñar Procesos Operacionales

Diseño

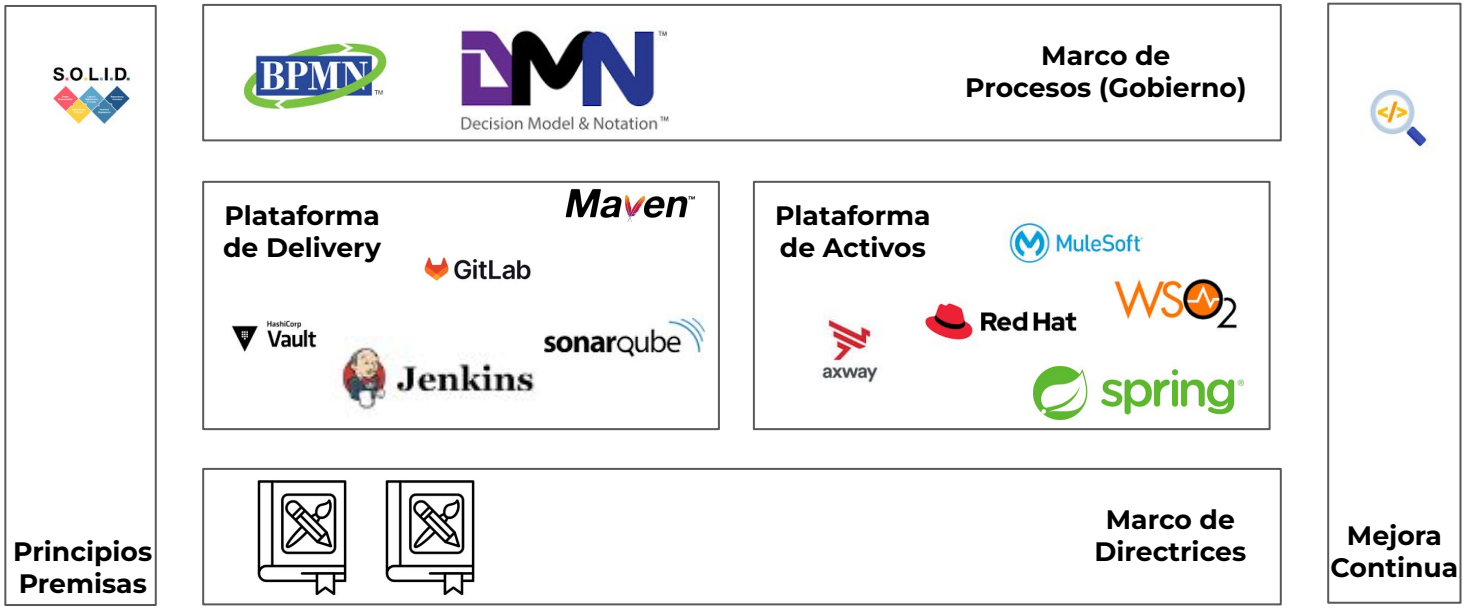
Evaluación Actual | Capacidades Actuales

1. No hay cultura DevOps presente dentro de la organización o se encuentra en una etapa muy temprana.
2. No hay un equipo de DevOps dedicado, lo que dificulta la propiedad y las operaciones CI/CD.
3. No hay infraestructura ni herramientas de CI/CD instaladas.
4. Los equipos no pueden asumir la responsabilidad adicional de mantener una plataforma CI/CD.
5. Los equipos de desarrollo se enfrentan continuamente a soluciones de CI/CD ad-hoc para cada proyecto.

Verticales Estratégicas



Verticales Estratégicas



Premisas de Onboarding

Definir Casos de Uso

1. Restful Data Service Provider
2. Soap Data Service Provider
3. API Proxy Passthrough
4. API Proxy Orchestration

Implementar Casos de Uso sobre staging platform

Construir Directivas

1. Establecer ambientes
2. Establecer estrategia de ramas/ branching
3. Establecer estrategia de versionamiento
4. Establecer convenciones de nombrado
5. Establecer estrategia de promoción
6. Establecer estrategia de automatización
7. Establecer estrategia de cumplimiento
8. Establecer estrategia de configuración
9. Establecer estrategia de secrets
10. otros...

Diseñar y Modelar Procesos

Procesos Operacionales

PRC-001

Compromiso y Especificación

PRC-002

Repositorio de Código & Archetype

PRC-003

Construcción de Activo (Api) - Local

PRC-004

Publicación Activo & Notificación de Revisión

PRC-005

Validación de cumplimientos

PRC-006

Compilación & Empaquetado

PRC-007

Despliegue Remoto

PRC-008

Pruebas Unitarias

. . .

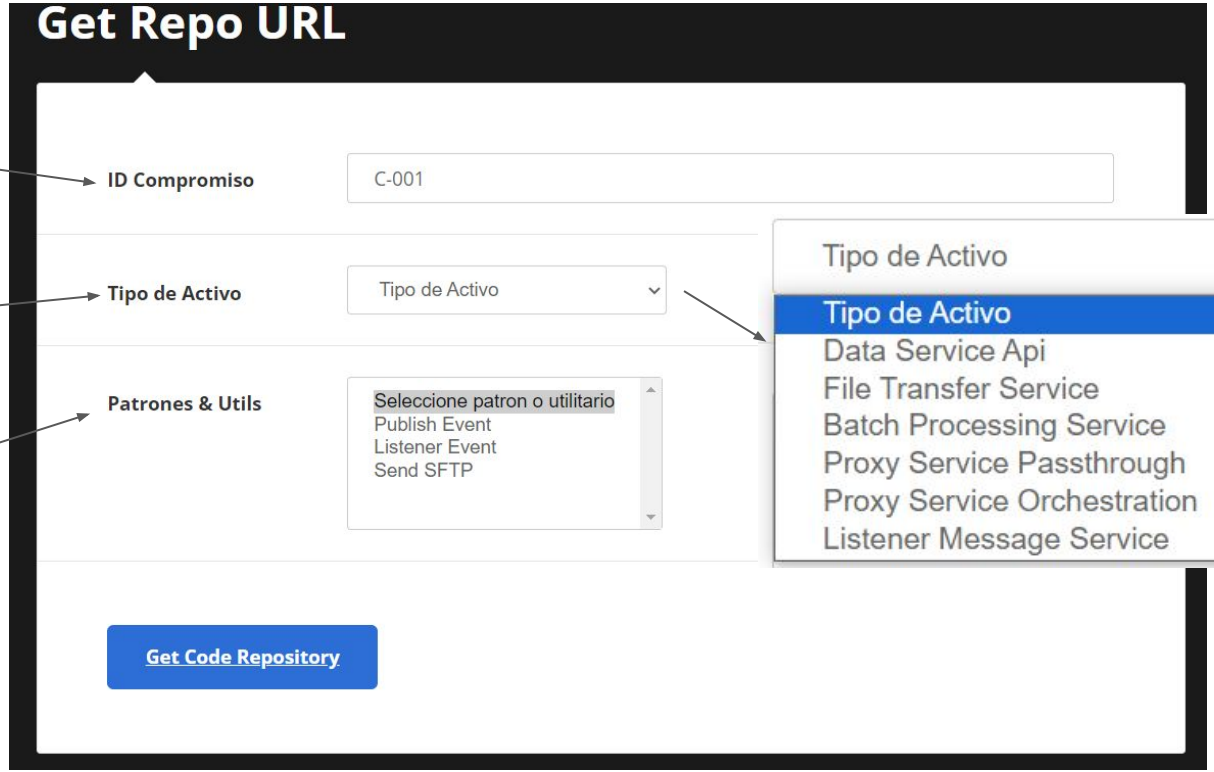
Procesos Operacionales | Un ejemplo

Repositorios de Compromisos

Tipos de Activos (“Apis”)

Repositorios de Patrones de integración

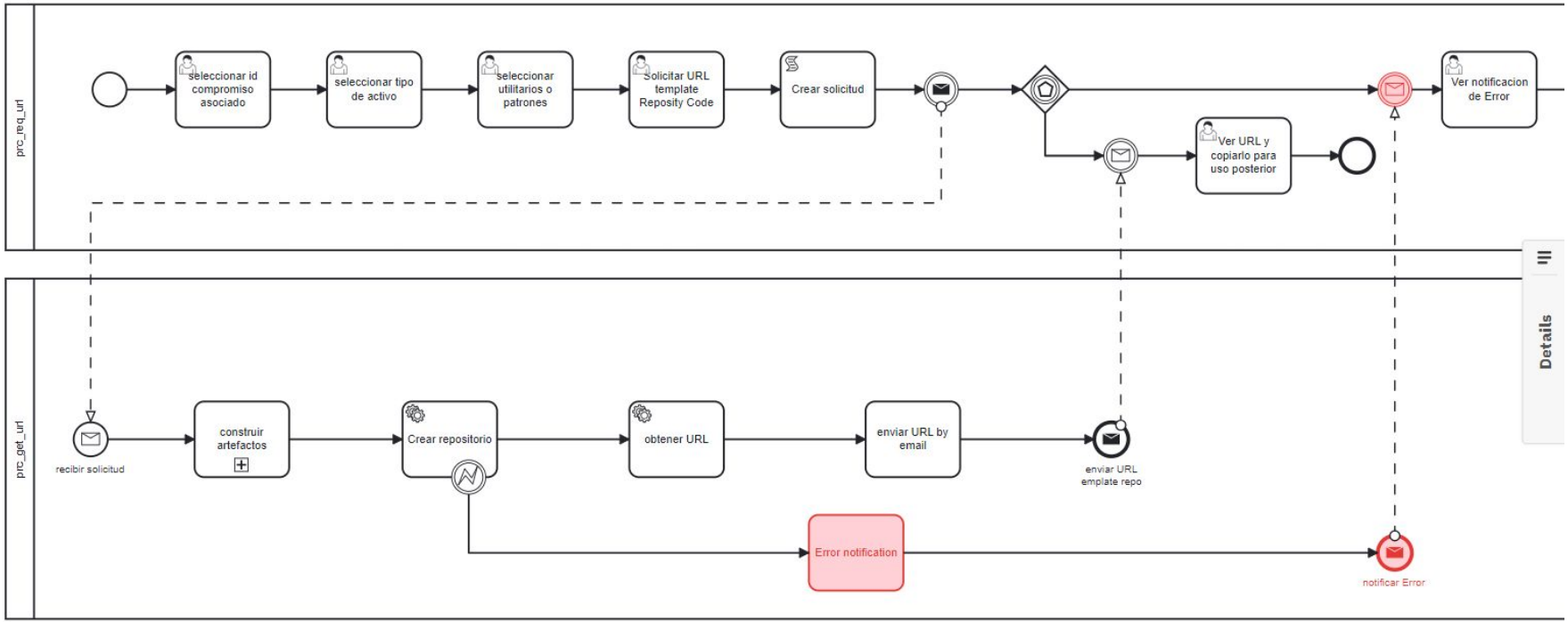
Repositorios de Arquetipos



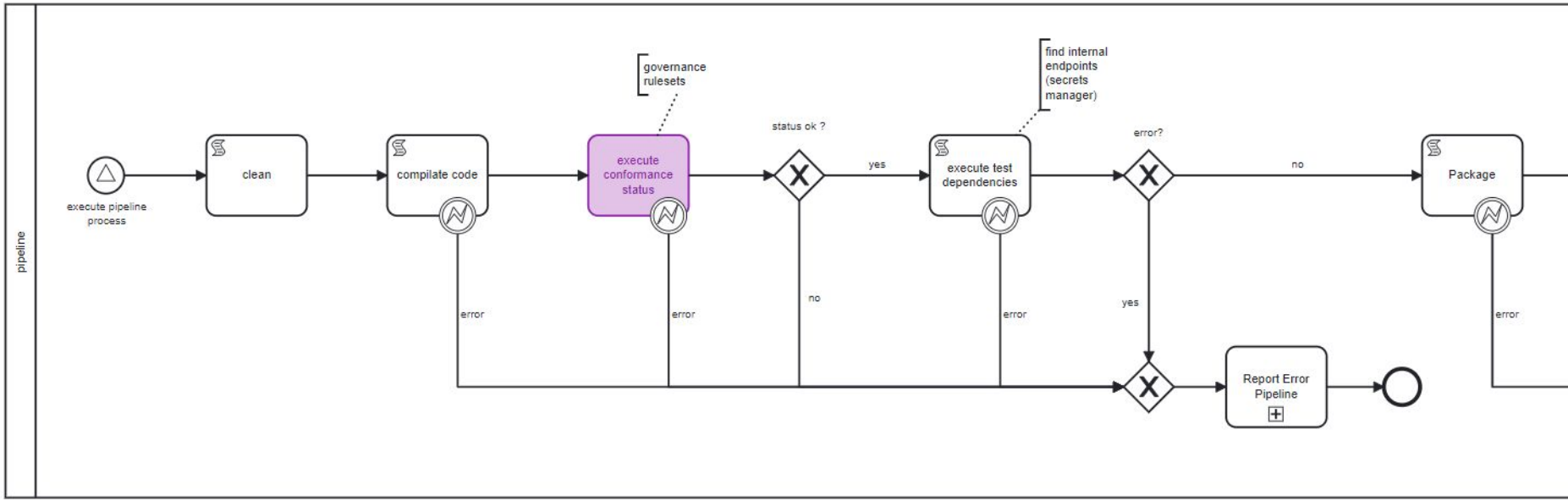
Procesos Operacionales | Un ejemplo

Repositorio de trabajo basado en Arquetipo y patrones de integración

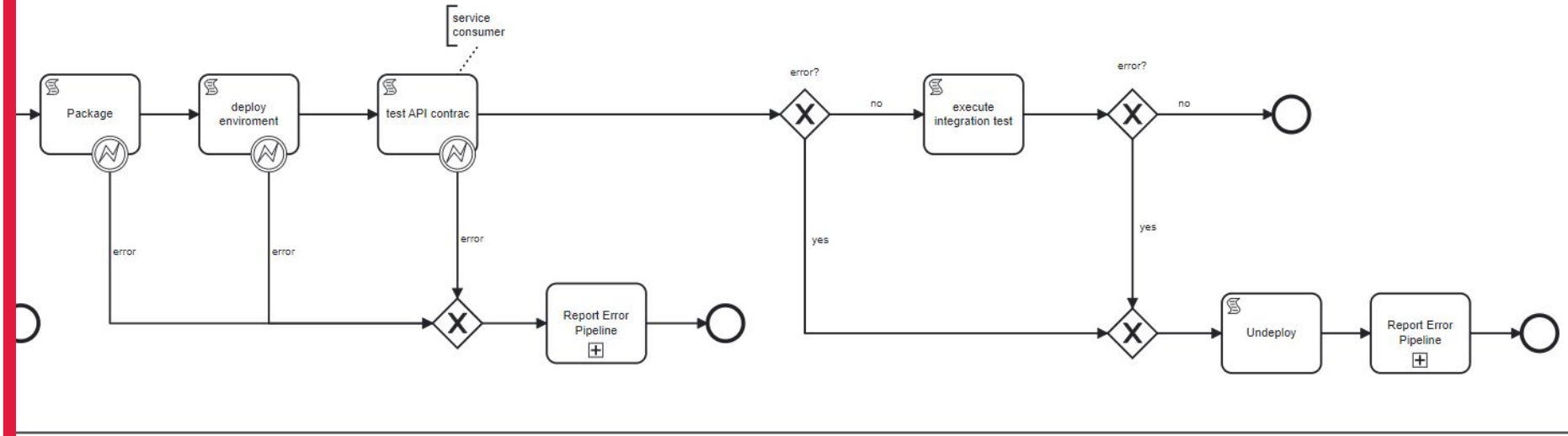
Procesos Operacionales | Un ejemplo



Procesos Operacionales = Pipeline



Procesos Operacionales = Pipeline



etapa 02

Arquitectura Plataforma de Activos	Aprovisionar Plataforma de Activos	Implementar Casos de Uso	Marco de Directrices y Procesos
- API Gateway - API Control - Integration Hubs - MicroServices - Cache Services - otros.	- MuleSoft Anypoint - WSO2 Lean Enterprise - Spring Cloud - RedHat - Otros	- Generar directrices - Deploy Manual - Modelo de procesos	

etapa 02

Arquitectura de plataforma de activos

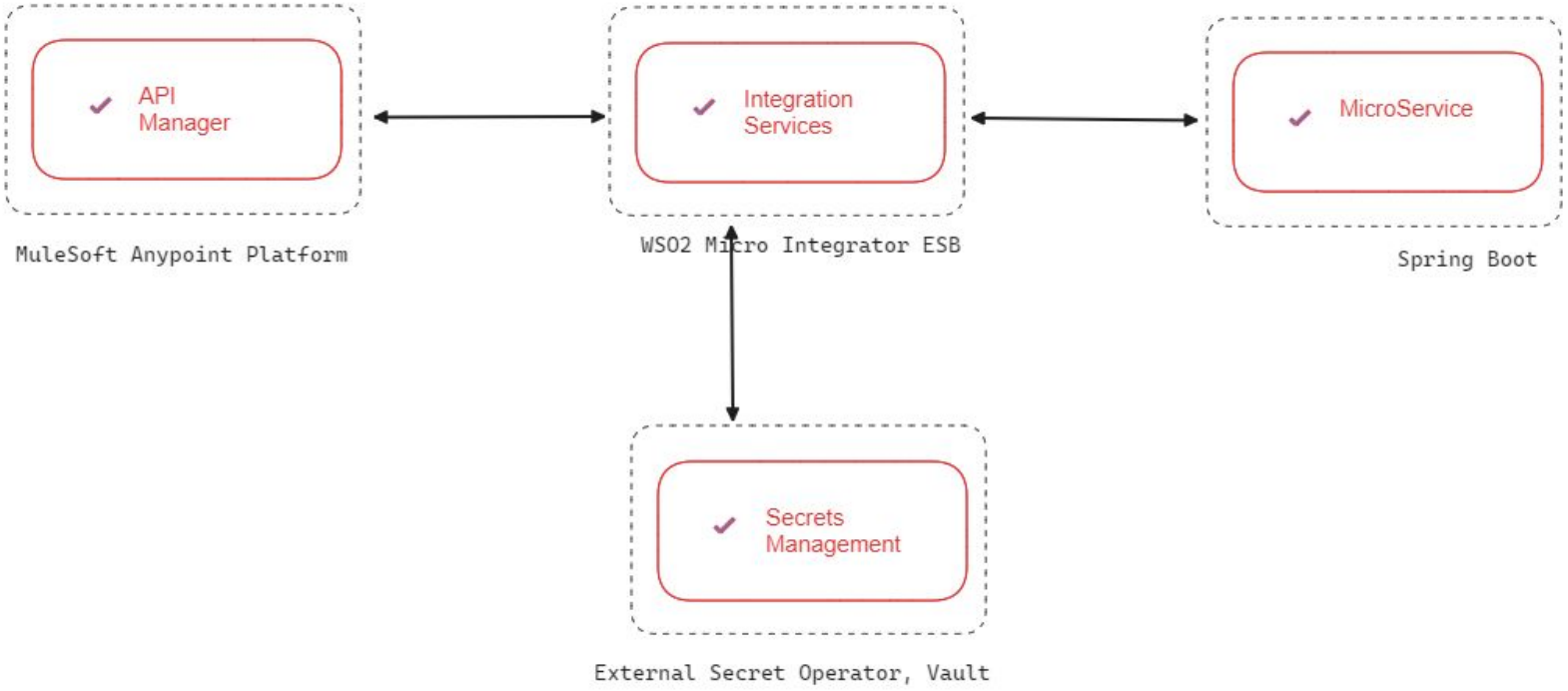
Aprovisionamiento

Implementación de Casos de uso

Marco Directrices & Procesos

Implementación

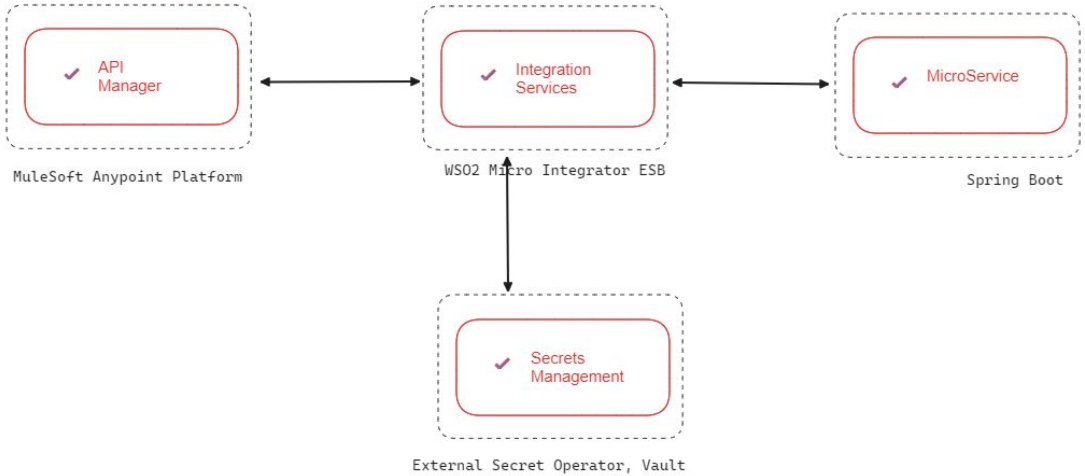
Arquitectura Plataforma de Activos



Aprovisionamiento Plataforma de Activos

- Ansible + Docker/K8
- Terraform + Docker/K8

Evitar el bloqueo de proveedores cloud sobre una infraestructura específica.



Casos de Uso en Plataforma de Activos

Casos de Uso
Construcción de Activos

1. Restful Data Service Provider
2. Soap Data Service Provider
3. API Proxy Passthrough
4. API Proxy Orchestration

Implementar Casos de Uso sobre Staging Platform

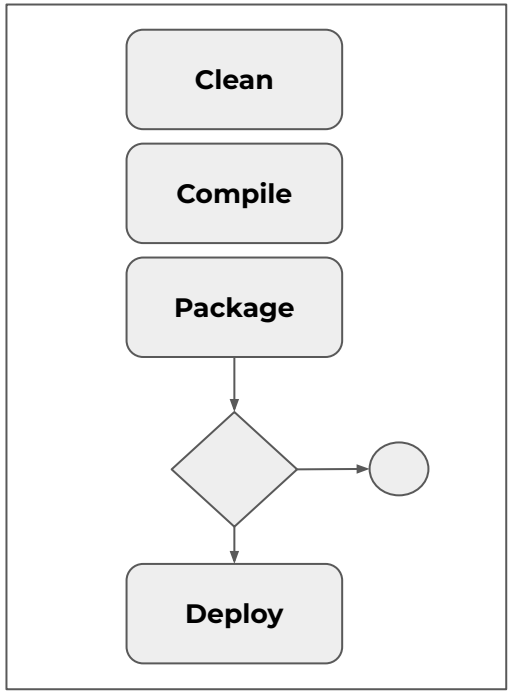
Directivas Generadas

1. Establecer ambientes
2. Establecer estrategia de ramas/ branching
3. Establecer estrategia de versionamiento
4. Establecer convenciones de nombrado
5. Establecer estrategia de promoción
6. Establecer estrategia de automatización
7. Establecer estrategia de cumplimiento
8. Establecer estrategia de configuración
9. Establecer estrategia de secrets
10. otros...

Marco de Procesos Generado

Procesos Operacionales

Pipeline



Jenkins

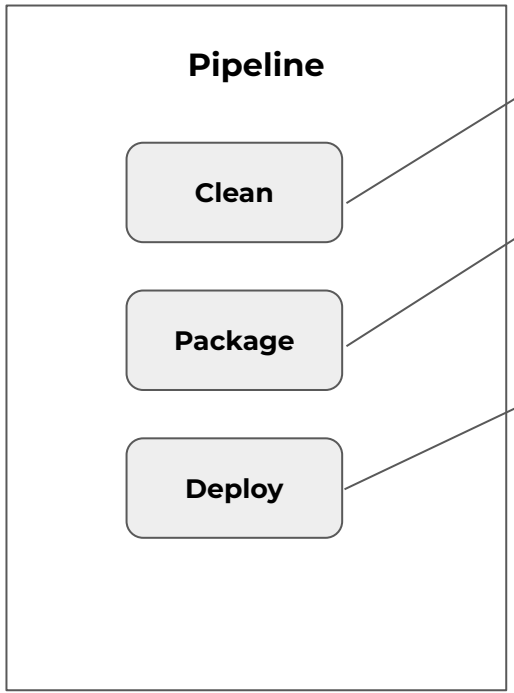
Maven

<APACHE ANT>

Groovy

Postman - Newman
Framework to run multiple Postman collections in single run

Procesos Operacionales

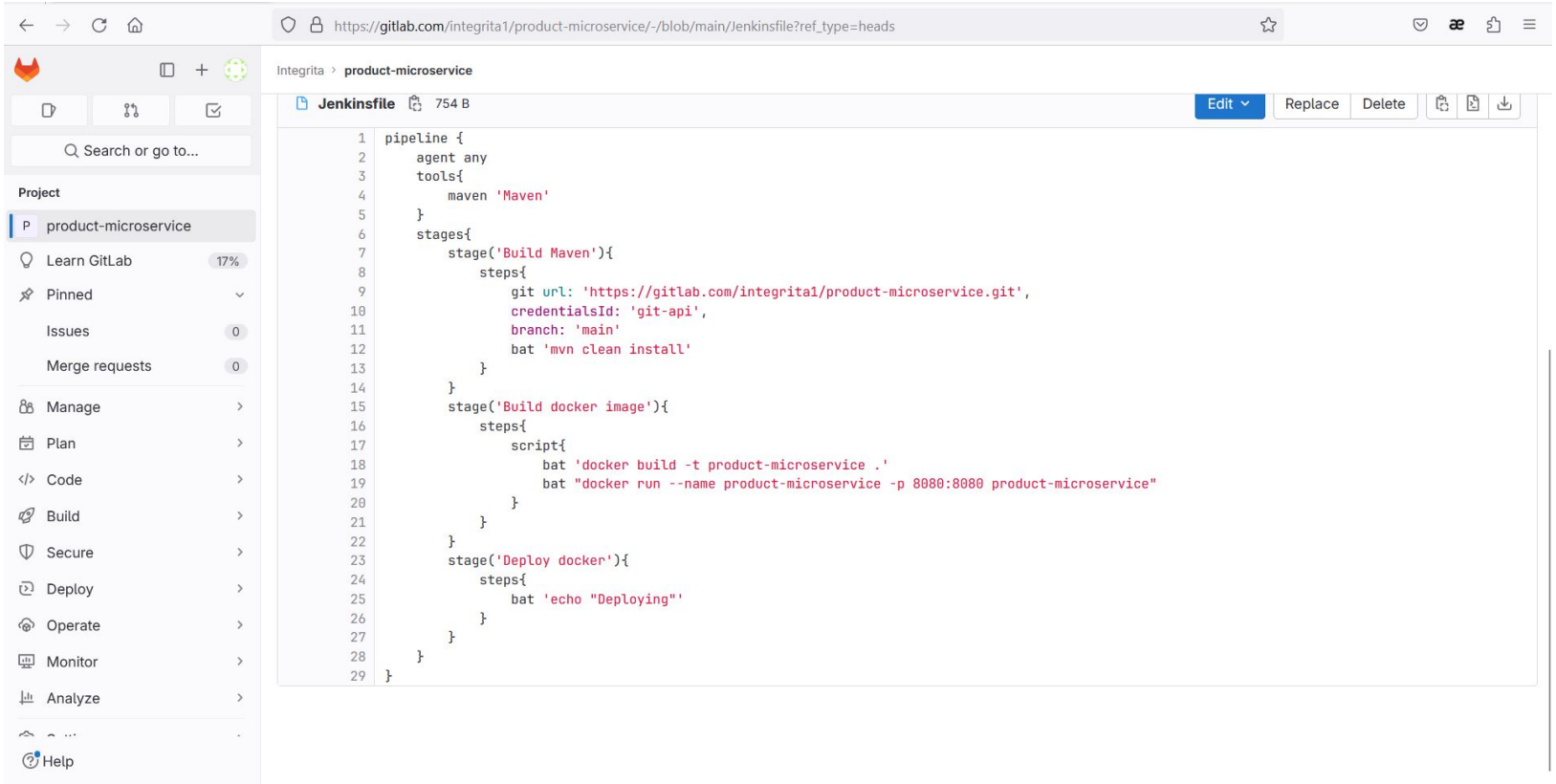


mvn clean

mvn -B -U -e -V clean -gs %M2SETTINGS% -DskipTests package

mvn -s $SETTINGS -f ./*CompositeExporter/pom.xml clean deploy
-Dmaven.deploy.skip=true
-Dmaven.car.deploy.skip=false
-Dtruststore.path=$WSO2_TRUSTSTORE_PATH
-Dtruststore.password=$WSO2_TRUSTSTORE_PASSWORD
-Dserver.url=$WSO2_DEV_SERVER_URL
-Dserver.username=$CARBON_SERVER_USERNAME
-Dserver.password=$CARBON_SERVER_PASSWORD

Procesos Operacionales



etapa 03

Arquitectura Plataforma de Delivery	Aprovisionar Plataforma de Delivery	Implementar Procesos & Pipelines
- Hacer proceso de forma manual para comprension. - Que stack tecnológico usar?. - Como se realice el despliegue ? - Automatizar procesos manuales	- MuleSoft Anypoint - WSO2 Lean Enterprise - Spring Cloud - RedHat - Otro	- Generar directrices - Deploy Manual - Modelo de procesos

etapa 03

Arquitectura de plataforma de activos

Aprovisionamiento

Implementación de pipelines

Implementación

Arquitectura Plataforma de Delivery

Code Review
Gerrit, **GitLab**, GitHub

Build
Go, Apache Gradle, **Apache Maven**, NPM

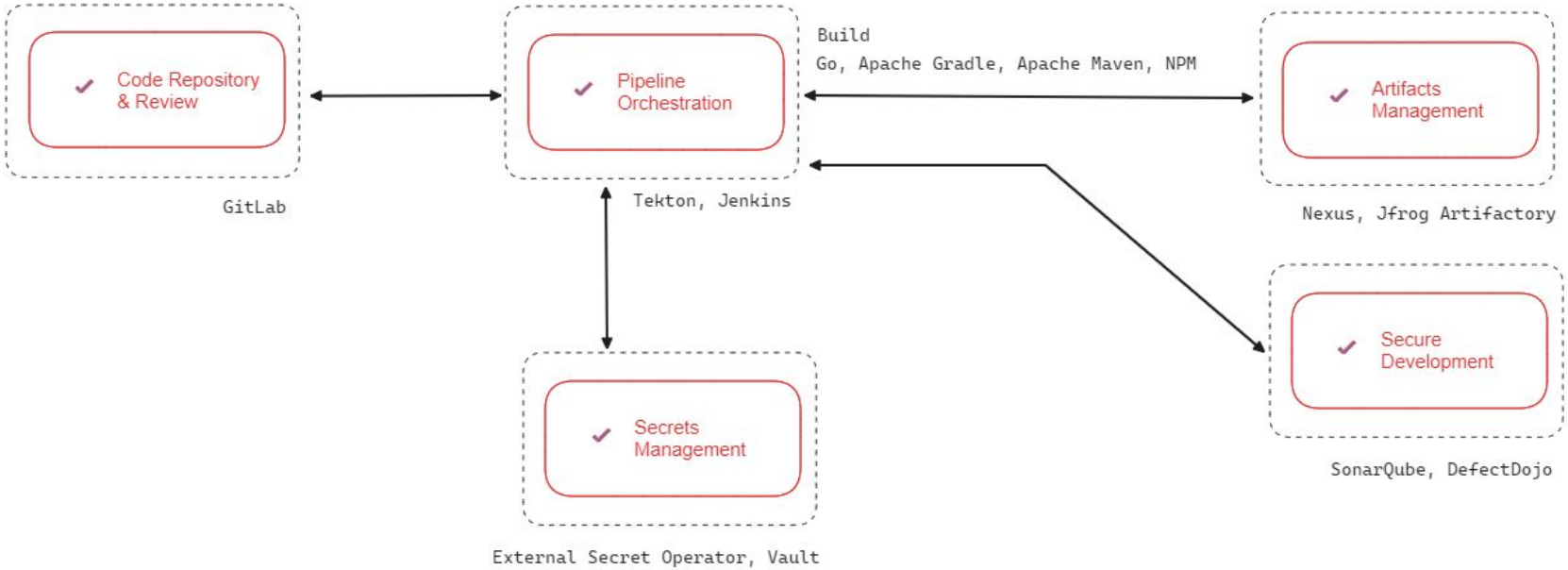
Artifacts Management
Nexus Repository, Jfrog Artifactory

Secrets Management
External Secret Operator, **Vault**

Secure Development
SonarQube, DefectDojo

Pipeline Orchestration
Tekton, **Jenkins**

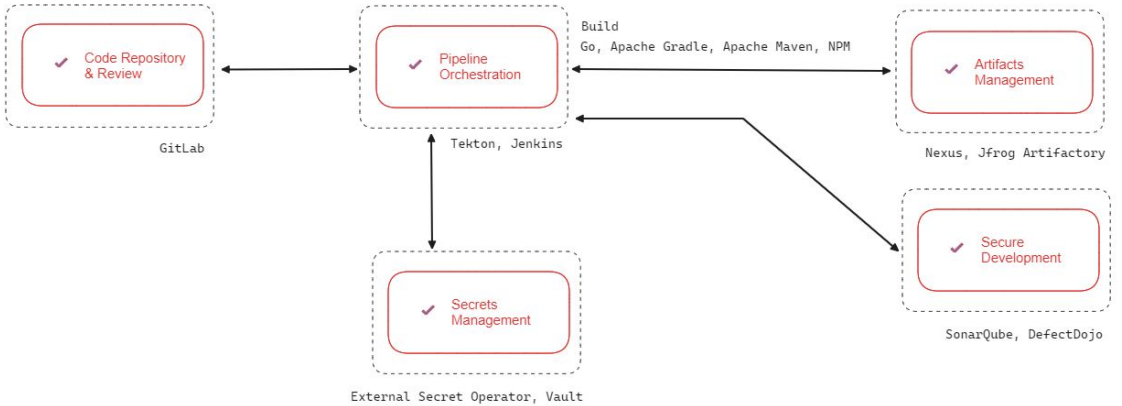
Aprovisionamiento Plataforma de Delivery



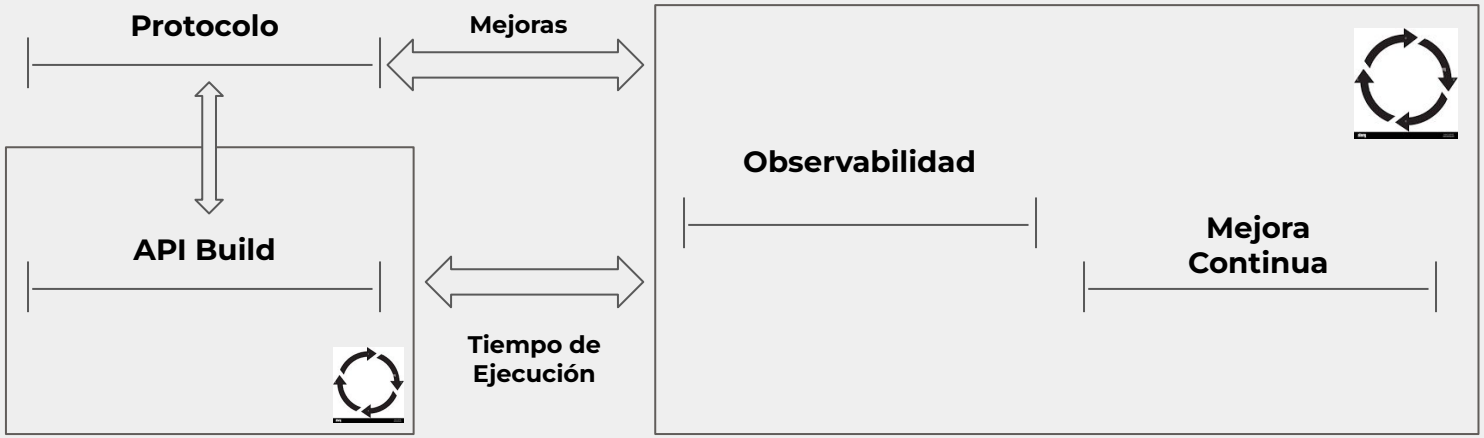
Aprovisionamiento Plataforma de Delivery

- Ansible + Docker/K8
- Terraform + Docker/K8

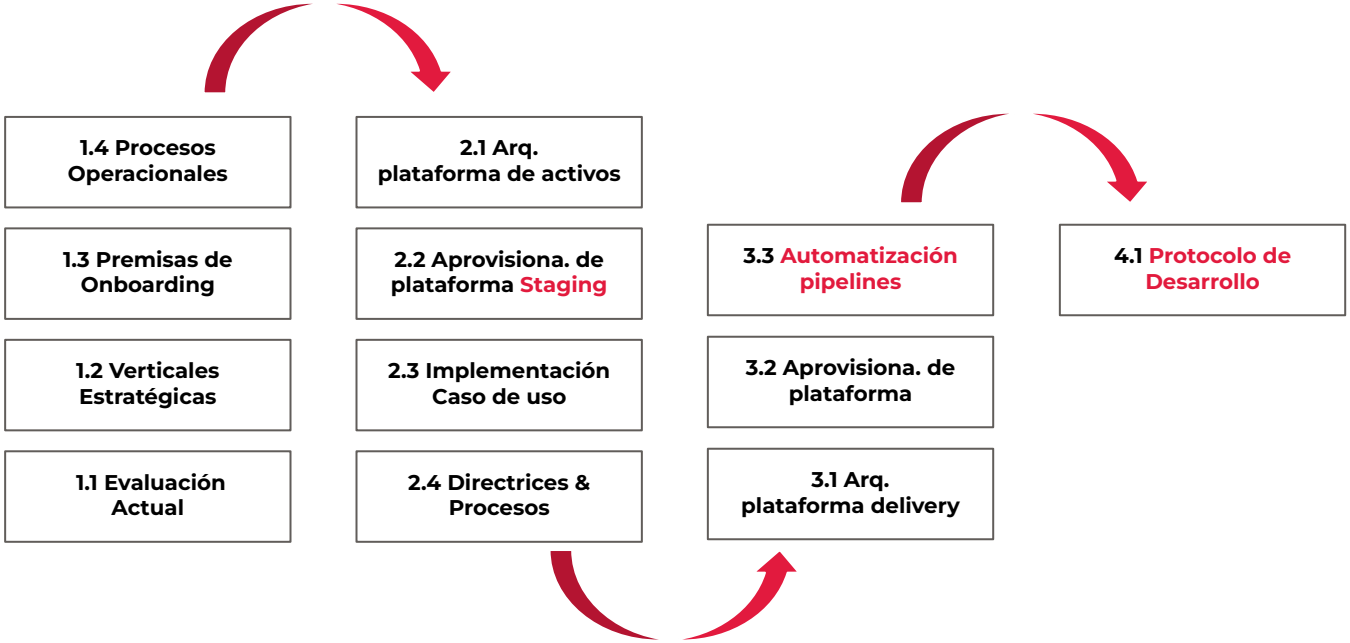
Evitar el bloqueo de proveedores cloud sobre una infraestructura específica.



etapa04


Protocolo
Mejoras
Observabilidad
API Build
Mejora
Continua
Tiempo de
Ejecución

Mapa de Ruta



1. Entregar software de manera más rápida y confiable.
2. Entregar características de software más rápido.
3. Construir sistemas distribuidos seguros, resistentes y de rápida evolución y escala.

Demostración

Delivery

Recomendaciones

Establezca arquetipos para estandarizar el desarrollo de activos (APIs, otros).	Construya directrices basado en casos de uso sobre ambiente de staging.
Modele y automatice los procesos de entrega mediante pipelines y procesos.	Incorpore arquitectos para definir un marco de gobierno y procesos para delivery.
Evite la acumulacion de tareas y roles. Simplifique.	Simplifique su estrategia de ramificación mediante “Trunk”

Preguntas

Procesos y videos en :
https://github.com/jccejas/deliveryIQ

¡GRACIAS POR ASISTIR!

Contacto

+34 **91 764 79 82**

contacta@apiaddicts.org

www.apiaddicts.org

Facebook

ApiAddicts

Linkedin

API Addicts

Twitter

@APIAddicts

Meet Up

API Addicts

Youtube

API Addicts

{API Addicts}