

Consideraciones al migrar un sistema tradicional hacia Cloud Native

Víctor Orozco

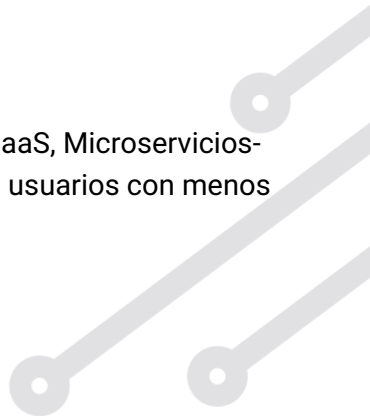
3 de marzo de 2022

Nabenik



NABENIK

Al migrar hacia nuevas tecnologías cloud -e.g. Serverless, BaaS, Microservicios- automáticamente van a conseguir escala -i.e. soportar más usuarios con menos dinero- y lanzar más rápido.



Enfoque para la construcción de sistemas informáticos modernos sobre entornos dinámicos como nubes privadas y públicas.

- Sistemas reactivos
- 12 factores cloud native
- Design patterns
- Domain Driven Design
- Microservices chassis y/o service mesh
- Orquestación de contenedores, serverless, BaaS
- Nube



- (Nos gustaría tener) Sistemas reactivos
- (Es posible con la metodología de) 12 factores Cloud Native
- (Usamos soluciones probadas mediante) design patterns
- (Fragmentamos el sistema mediante) Domain Driven Design
- (Implementamos los servicios con) Microservices chassis y/o service mesh
- (Hacemos despliegue) Mediante orquestación de containers, funciones serverless, BaaS
- (Contratamos) Nube

Un proceso de migración hacia Cloud Native es en si mismo un macro proyecto.



El proyecto de migración



Cloud native - Experiencias previas

- Inicio
 - Diagnostico de arquitectura y equipo de trabajo
- Planificación
 - Elaboración de hoja de ruta
- Ejecución
 - Implementación
 - Adquisición
 - Capacitación
 - Programación
- Monitoreamiento y control
 - Tecnológico: Evaluaciones de entregables, métricas de calidad
 - Proyecto: Presupuesto, plazos, viabilidad real de los entregables
- Cierre
 - Documentación viva
 - Mejoras futuras

No más de dos reuniones exploratorias. En promedio 2 horas por reunión
Stakeholders que no pueden faltar:

- Arquitecto de software (Tech Lead, Dev. Sr.)
- Jefe de infraestructura (Sysadmin principal, SRE)
- Punto de contacto (generalmente el arquitecto de software o soluciones)


Preguntas clave:

- Motivación principal
- Personal actual
- Stack(s) tecnológicos

Revisión de arquitectura actual

- Datos importantes
- Problemas
- Propuesta macro de mejoras
- Listado de acciones puntuales
- Hoja de ruta con opciones (contratación, consultoría, tercerización, entrenamiento)
- Contrato(s) basado(s) en entregables





CATEGORIES

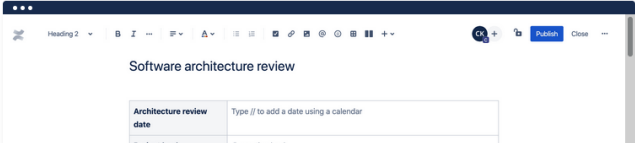
- [Featured](#)
- [Partners](#)
- [Business Strategy](#)
- [Design](#)
- [Docs & Reports](#)
- [Human Resources](#)
- [Marketing & Sales](#)
- [Personal](#)
- [Product Management](#)
- [Productivity](#)

Software architecture review template

BY ATlassian

Map out your software architecture and review your software quality

Use template



Siguientes pasos

Los siguientes pasos describen proyectos que pueden utilizarse como base para la implementación del sistema completo DevOps.

Proyecto	Descripción	Tiempo estimado	Oportunidades de tercerización
Implementación de automation testing en front-end y back-end	<p>Para la implementación de testing se necesita:</p> <ul style="list-style-type: none">• Capacitar a los desarrolladores en fundamentos de testing para Java (Spring Testing), JavaScript(Karma y Protractor) y herramientas complementarias (Docker, TestContainers)• Definir un porcentaje de cobertura de testing como criterio de aceptación en el servidor de integración continua• Automatizar la ejecución de pruebas mediante las herramientas de construcción	<p>1 mes de capacitación</p> <p>El tiempo de testing sobre los nuevos servicios es proporcional a la cantidad de servicios a desarrollar</p>	<ul style="list-style-type: none">• Capacitación del personal existente• Contratación de un QA Automation Developer

1-Punto de contacto, 2-Medio de comunicación inmediato, 3- Medio de comunicación no repudiable.

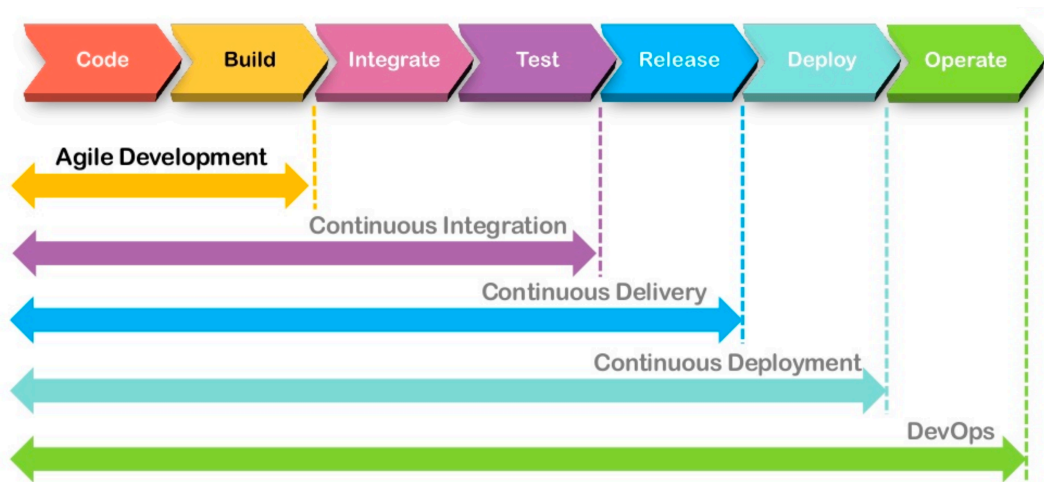
- Cultura
 - DevOps
 - Test Driven Development
- Infraestructura
 - Accesos remotos
 - VCS, CI/CD
 - Plataforma Cloud Native -e.g. Openshift, Kubernetes, Amazon EKS, Oracle Kubernetes Engine-
 - Observabilidad -e.g. Linkerd, Prometheus, Grafana, ELK, alarmas-

Idealmente realizar una presentación/transferencia de tecnología por entregable.

Todas estas etapas producirán "**documentación viva**".

- Entrenamiento y desarrollo
 - Arquetipo / Proyecto 0
 - SCM -e.g. Maven, NPM -
 - TDD, DDD, Microservice Chassis seleccionado, infraestructura como código
 - Migración de sistemas actuales mediante pipelines
 - Ecosistema alrededor del monolito
 - Nuevos proyectos nacen con 12 factors cloud native


Ejecución - DevOps



Ejecución - CI/CD

 **failed** Pipeline #147 triggered 2 weeks ago by  **Victor Orozco**

Reading key from p12

 4 jobs for **develop** in 3 minutes and 11 seconds (queued for 8 seconds)

 **latest**

 **faa7fa5f** 

 No related merge requests found.

Pipeline Needs Jobs **4** Failed Jobs **1** Tests **4**

Compile


 build



 sonarqube-c...




Package

 image for cer...

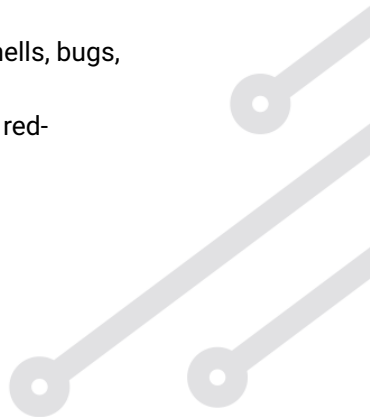


Deploy

 deploy for cer...



- Tecnológico
 - Métricas de calidad del código -e.g. cobertura, code smells, bugs, vulnerabilidades-
 - Métricas de desempeño -e.g. caídas e intermitencia de red-
 - Instrumentación de la infraestructura
 - Monolitos existentes migrados satisfactoriamente
- Proyecto
 - Presupuesto
 - Entregables estimados vs plazo
 - Resultados y percepción de los desarrolladores



Control - Calidad

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

?

Q Search for projects...

A

My FavoritesAll

Filters

Quality Gate

Passed3

Failed0

Reliability (🐛 Bugs)

A2

B0

C1

D0

E0

Security (🔒 Vulnerabilities)

A2

B0

C0

D1

E0

Security Review (🔒 Security Hotspots)

Q Search by project name or key

Create Project

3 projects

Perspective: Overall Status

Sort by: Name

☆ Passed Last analysis: 16 days ago

🐛 Bugs

🔒 Vulnerabilities

🔒 Hotspots Reviewed

🔒 Code Smells

Coverage

Duplications

Lines

A

A

E

A

0.0%

0.0%

787 XS Java, XML...

☆ Passed Last analysis: 14 days ago

🐛 Bugs

🔒 Vulnerabilities

🔒 Hotspots Reviewed

🔒 Code Smells

Coverage

Duplications

Lines

C

A

E

A

42.6%

0.0%

9.9k S CSS, Typ...

☆ Passed Last analysis: 14 days ago

🐛 Bugs

🔒 Vulnerabilities

🔒 Hotspots Reviewed

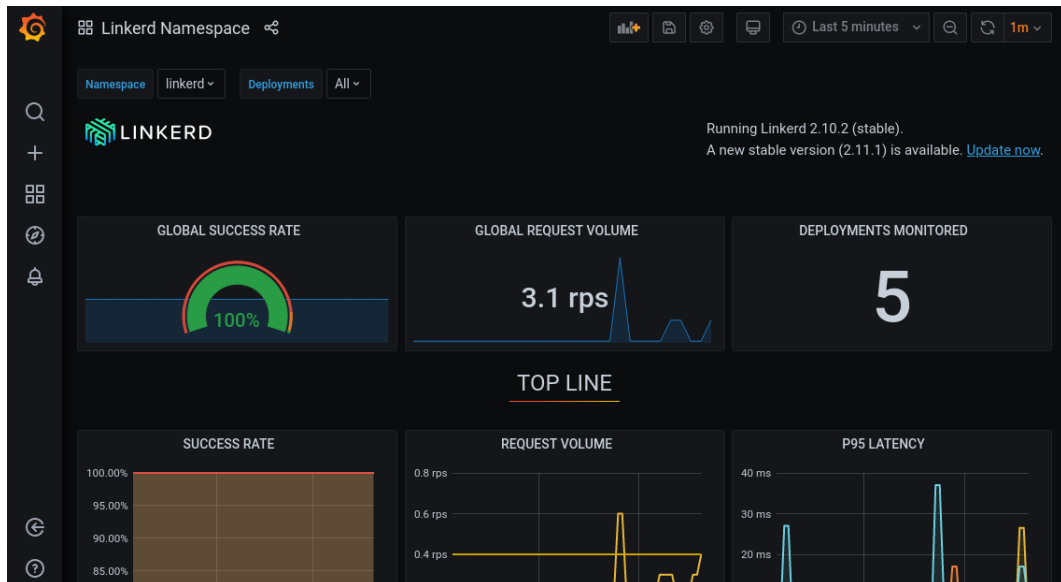
🔒 Code Smells

Coverage

Duplications

Lines

Control - Instrumentación



- Transición de implementación hacia soporte
- Documentación viva
- Propuesta de mejoras





The screenshot shows the GitLab web interface. The top navigation bar is dark blue with the GitLab logo, a menu icon, and a search bar. The left sidebar is light gray and contains a list of project navigation items: Project information, Repository, Issues (0), Merge requests (0), CI/CD, Security & Compliance, Deployments, Monitor, Infrastructure, Packages & Registries, and a collapsed 'Lib. Análisis' section. The main content area has a dark blue header with the text 'To use this project you need the following requirements' and a list of requirements: Docker Desktop \$\$\$ (Windows/MacOS), Docker Engine or Podman. Below this, the section 'Factor 3: Configuraciones' is displayed. It includes a 'Descripción' paragraph, an 'Estado anterior' paragraph, and an 'Acciones ejecutadas' list. The list item mentions externalizing configuration for a backend test, specifically using application*.properties files, and mentions a dependency on the environment (e.g., JVM variable in Kubernetes).

GitLab Menu

Search GitLab

M

- Project information
- Repository
- Issues 0
- Merge requests 0
- CI/CD
- Security & Compliance
- Deployments
- Monitor
- Infrastructure
- Packages & Registries
- Lib. Análisis

« Collapse sidebar

To use this project you need the following requirements

- Docker Desktop \$\$\$ (Windows/MacOS), Docker Engine or Podman

Factor 3: Configuraciones

Descripción: Debe ser posible alterar el comportamiento de una aplicación dependiendo de su entorno e idealmente sin necesidad de re-programar la aplicación. Estas configuraciones deben ser externalizadas mediante mecanismos propios de la aplicación y variables de entorno o archivos de configuración

Estado anterior: Ninguna de las aplicaciones analizadas contaban con externalización de la configuración y existe una dependencia fuerte del entorno

Acciones ejecutadas:

- Se externalizó la configuración de las conexiones hacia base de datos en el caso de la prueba de concepto backend específicamente mediante los archivos descriptores `application*.properties`, propio de Spring Boot. Ahora es obligatorio que las credenciales y string de conexión sean proporcionados mediante el entorno -e.g. variable de entorno, bandera de JVM, variable en Kubernetes-



- vorozco@nabenik.com
- @tuxtor
- <http://voroeco.com>
- <http://tuxtor.shekalug.org>



This work is licensed under
Creative Commons Attribution-
NonCommercial-ShareAlike 3.0
Guatemala (CC BY-NC-SA 3.0 GT).