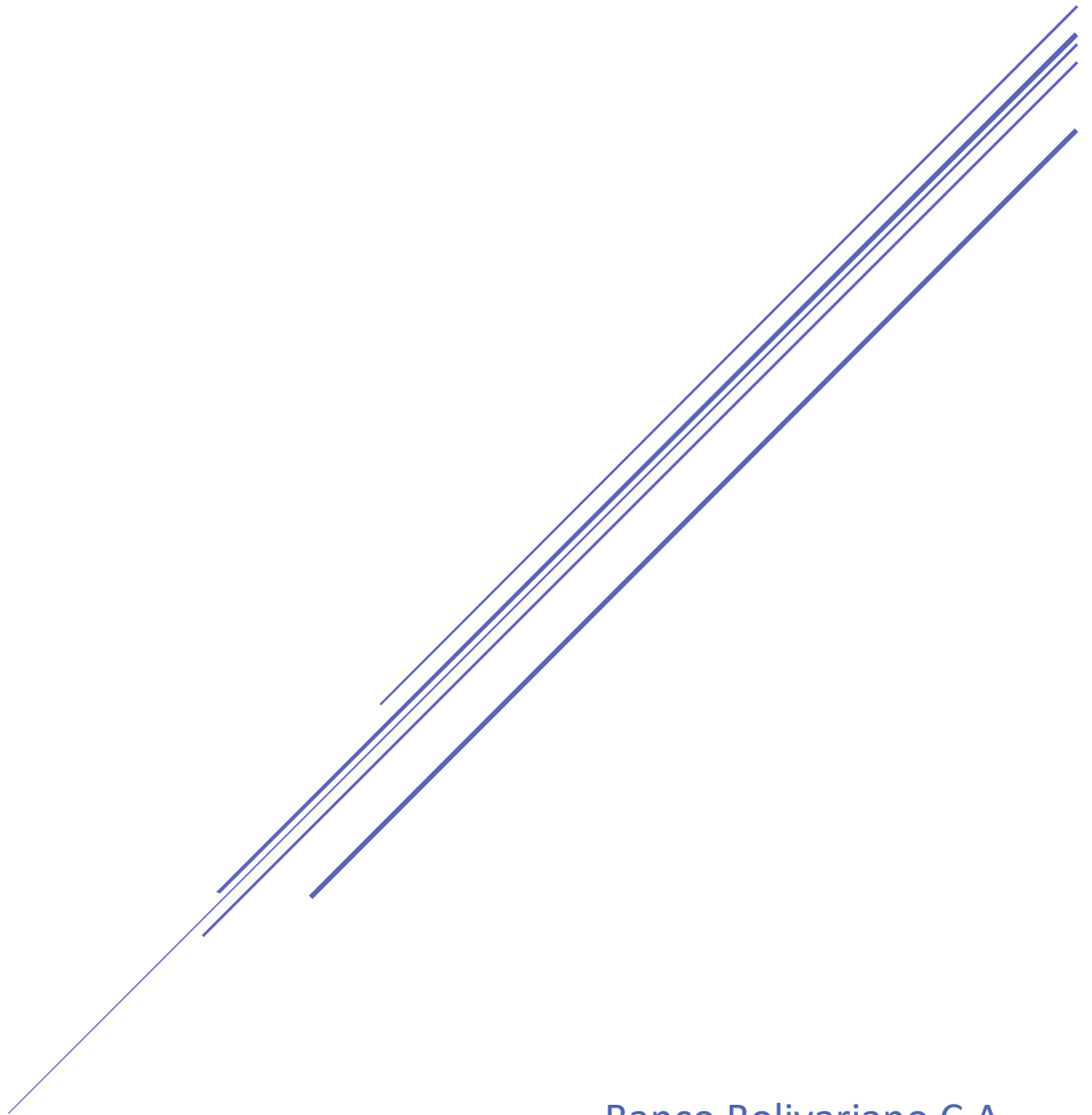


# GUÍA DE USO – INTEGRACIÓN CONTINUA (MICROSERVICIOS)

Pipelines de Integración Continua - Microservicios



Banco Bolivariano C.A.  
Integración Continua



## Tabla de Contenidos

Objetivos .....	3
Alcance .....	3
Generalidades .....	3
Flujo continuo: .....	4
Flujo de liberación de tareas:.....	5
Uso de la herramienta .....	6
Pre-Requisitos .....	6
Modo de operación.....	8
Preguntas Comunes .....	12
Ilustraciones .....	13



Revisión	Autor	Fecha
Emisión Inicial	Desarrollo	2022-04-01

## Objetivos

Describir los pasos necesarios para el uso de los pipelines de integración continua desde el punto de vista del usuario desarrollador.

## Alcance

Este documento describe las actividades básicas de los pipelines de integración continua para microservicios java con el sistema de construcción automático maven desde el punto de vista del usuario desarrollador.

## Generalidades

El sistema de integración continua para microservicios de Banco Bolivariano comprende de los siguientes componentes expuestos en red local de banco:

- Control de Versiones: Git + GitLab
- Repositorio de Artefactos: Nexus
- Análisis estático de código: Sonarqube
- Automatización: Jenkins

Servicio	URL
GitLab	<a href="https://gitlabsrv.bolivariano.fin.ec">https://gitlabsrv.bolivariano.fin.ec</a>
Nexus	<a href="https://nexusdes.bolivariano.fin.ec">https://nexusdes.bolivariano.fin.ec</a>
Sonarqube	<a href="https://sonarqdes.bolivariano.fin.ec">https://sonarqdes.bolivariano.fin.ec</a>
Jenkins	<a href="https://jenkinsdes.bolivariano.fin.ec">https://jenkinsdes.bolivariano.fin.ec</a>

El objetivo de estos servicios es automatizar el flujo de Desarrollo y liberación a pruebas para permitir la disminución de actividades manuales y optimizar los tiempos del desarrollador.

## Servicios de IC Banco Bolivariano

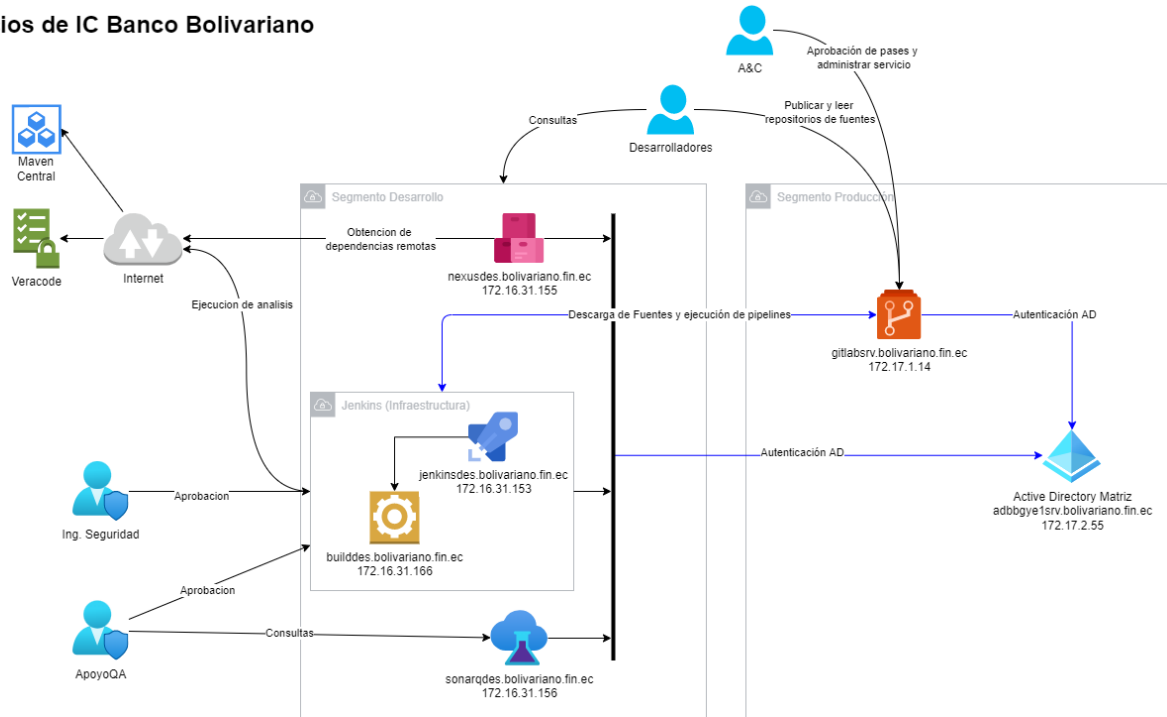


Ilustración 1 - Interacción de usuarios y servidores en el esquema de IC para microservicios

Existen dos flujos básicos definidos para microservicios

Flujo continuo:

- Ejecutado de forma automática cada vez que el developer publica código en el ramal desarrollo en Gitlab o bajo demanda.
- Se realizan verificaciones rápidas de los fuentes, el desarrollador no requiere estar pendiente de este pipeline salvo cuando le lleguen correos de alerta indicando que se ha roto el build.
- **Objetivo:** Que todo el código que el desarrollador suba a desarrollo tenga niveles mínimos de calidad y que no genere algún error de build que afecte la estabilidad de la rama.

El flujo continuo que se cubre con los pipelines desarrollados es el siguiente:

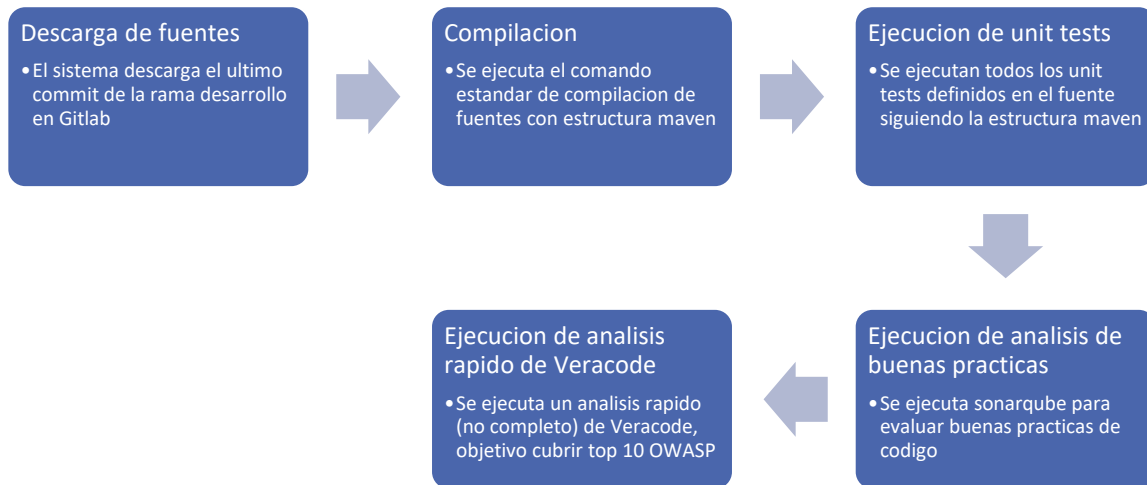


Ilustración 2 - Flujo Continuo

Flujo de liberación de tareas:

- Ejecutado solo bajo demanda por el desarrollador, debe ingresar a Jenkins y seleccionar el pipeline deseado, ingresar el código de tarea JIRA y el objetivo del pase.
- Se ejecutan los mismos pasos que el flujo continuo, pero adicionalmente se ejecutan pruebas completas de seguridad, se generan matrices unitarias y se solicita automáticamente la aprobación de ApoyoQA e Ing. Seguridad (Vulnerabilidades Veracode)
- **Objetivo:** Liberar un desarrollo para que se genere un ejecutable y se pueda enviar a producción un pase.

El flujo de liberación de tareas que se cubre con los pipelines desarrollados es el siguiente:

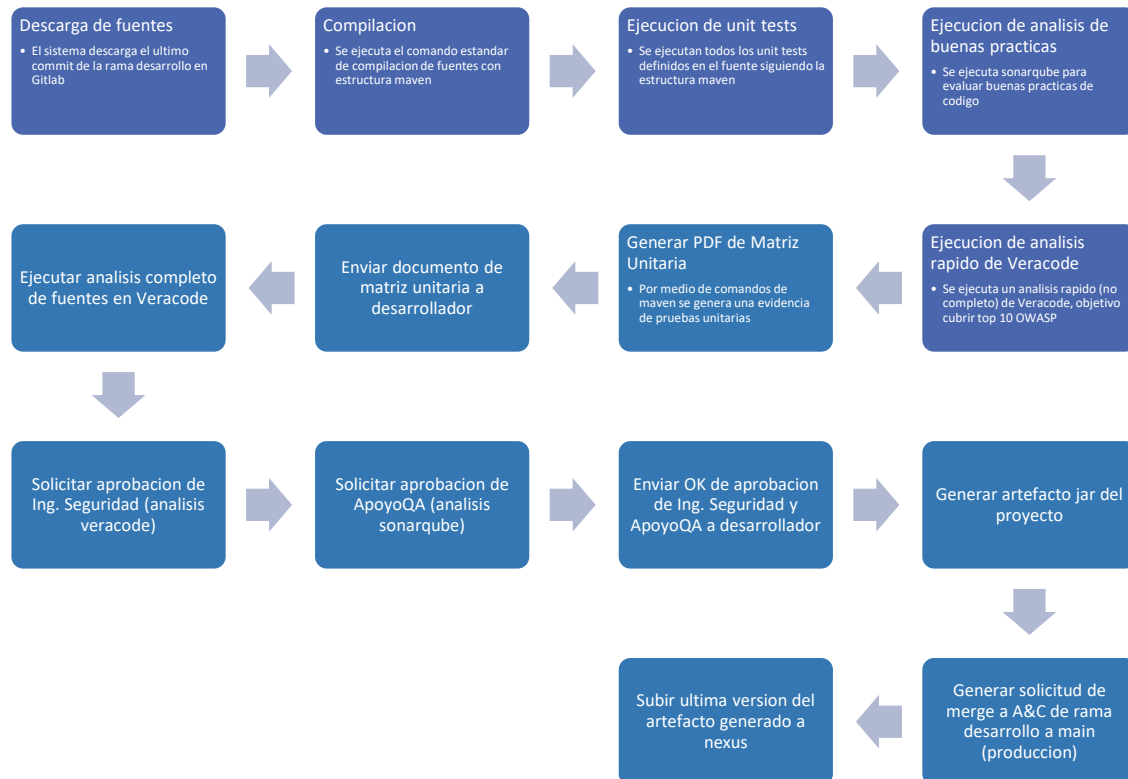


Ilustración 3 - Flujo de liberación de tareas

## Uso de la herramienta

### Pre-Requisitos

#### Acceso al sistema de versionamiento GitLab de Producción

Para los usuarios a los cuales ya se les solicito el anexo I el trámite ya se encuentra realizado y tiene acceso a <https://gitlabsrv.bolivariano.fin.ec>.

Si es un usuario nuevo, debe solicitar la aprobación a su jefatura inmediata de la actualización de su anexo I agregando lo siguiente:

Usuario debe tener acceso al versionador de fuentes gitlabsrv.bolivariano.fin.ec -> 172.17.1.14 TCP 443

Agregar al usuario <<su usuario de matriz>> al grupo de AD <<devops\_users>>

Rol: <<para lideres de grupo o segundos al mando "maintainer", para desarrolladores "developer">>

Grupo: <<su grupo de trabajo (ejem.- OnBoarding)>>

Cabe indicar que Seg. Informática y AyC valorarán el permiso que este solicitando y tendrán la potestad de aprobar o rechazar su petición.

### Acceso a servicios de IC Bolivariano

Debe ingresar vía navegador a los siguientes sitios con su usuario y clave de matriz:

Servicio	URL
GitLab	<a href="https://gitlabsrv.bolivariano.fin.ec">https://gitlabsrv.bolivariano.fin.ec</a>
Nexus	<a href="https://nexusdes.bolivariano.fin.ec">https://nexusdes.bolivariano.fin.ec</a>
Sonarqube	<a href="https://sonarqdes.bolivariano.fin.ec">https://sonarqdes.bolivariano.fin.ec</a>
Jenkins	<a href="https://jenkinsdes.bolivariano.fin.ec">https://jenkinsdes.bolivariano.fin.ec</a>

### Configuración de cliente maven

Puede apuntar al servidor nexus modificando su archivo settings.xml de la ruta de su binario de Maven:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.2.0 http://maven.apache.org/xsd/settings-1.2.0.xsd">
  <pluginGroups>
  </pluginGroups>
  <proxies>
  </proxies>
  <servers>
    <server>
      <id>NexusRepo</id>
      <username>SU_USUARIO_MATRIZ</username>
      <password>SU_CLAVE_MATRIZ</password>
    </server>
  </servers>
  <mirrors>
    <mirror>
      <id xmlns="">central-mirror</id>
      <name xmlns="">Mirror Repository of Maven Central</name>
      <url xmlns="">http://nexusdes.bolivariano.fin.ec/repository/maven-public/</url>
      <mirrorOf xmlns="">central</mirrorOf>
    </mirror>
  </mirrors>
  <profiles>
    <profile>
      <id>SinSeguridad</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <properties>
        <maven.wagon.http.ssl.insecure>true</maven.wagon.http.ssl.insecure>
        <maven.wagon.http.ssl.allowall>true</maven.wagon.http.ssl.allowall>
      </properties>
      <maven.wagon.http.ssl.ignore.validity.dates>true</maven.wagon.http.ssl.ignore.validity.dates>
    </profile>
  </profiles>
  <activeProfiles>
    <activeProfile>SinSeguridad</activeProfile>
  </activeProfiles>
</settings>
```



Recordar exceptuar el servidor `nexusdes.bolivariano.fin.ec` del proxy de banco

#### *Agregar unit tests y la generación del reporte de pruebas en su proyecto*

Para más información sobre como agregar unit tests y generar el reporte de pruebas mediante configuración de Maven, usar las guías anexas”

- Documentación → GuiaConfiguracionPluginReportes
- Training → (Capacitación\_Entrega Configuración Unit Tests MS y Mock Services - DevOps Fase I Banco Bolivariano-20220218\_140511-Meeting Recording.mp4 y Capacitación Pruebas Automáticas-20211019\_140809-Meeting Recording.mp4)

#### *Modo de operación*

Si su microservicio ya tiene un pipeline asociado, básicamente ya esta listo para trabajar. Puede verificar su existencia ingresando a <https://jenkinsdes.bolivariano.fin.ec> y escogiendo su grupo aplicativo, tipo de fuente (microservicio) y pipeline con el nombre de su servicio.

Al iniciar su sprint, clone el proyecto con los fuentes vigentes desde <https://gitlabshr.bolivariano.fin.ec> y muévase al Branch “desarrollo” para descargar la última versión del mismo.

Una vez en el Branch “desarrollo” podrá crear un nuevo Branch temporal de tipo “feature” con el código de tarea JIRA asociado a su cambio. Haga cuantos cambios usted requiera en ese Branch temporal y cuando tenga código funcional proceda a realizar merge de ese Branch al Branch desarrollo. Recuerde que encontrara documentación de la práctica de versionamiento que estamos incorporando (GitFlow) en el archivo “Gitlab Banco Bolivariano.pdf”.

La acción de realizar push al ramal desarrollo automáticamente lanza el pipeline de integración continua correspondiente a su microservicio en modo “flujo continuo”. Puede revisar el avance del pipeline en cualquier momento, aunque no es necesario salvo que reciba un correo de notificación informando que hubiere fallado el build, en cuyo caso deberá inmediatamente revisar su contribución a fin de corregir el error. La notificación de build fallido llegara constantemente a todo su grupo hasta que nuevamente el build funcione.

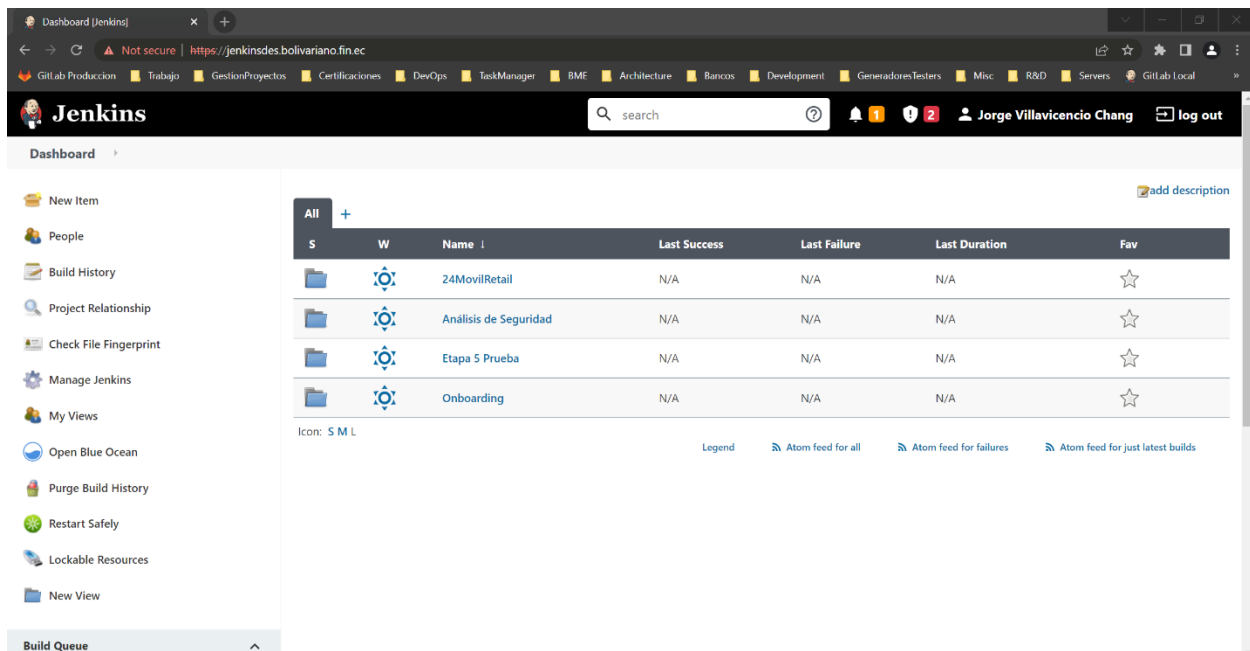


Ilustración 4 - Pantalla principal de Jenkins, vista de grupos aplicativos (actualmente 24MovilRetail y Onboarding)

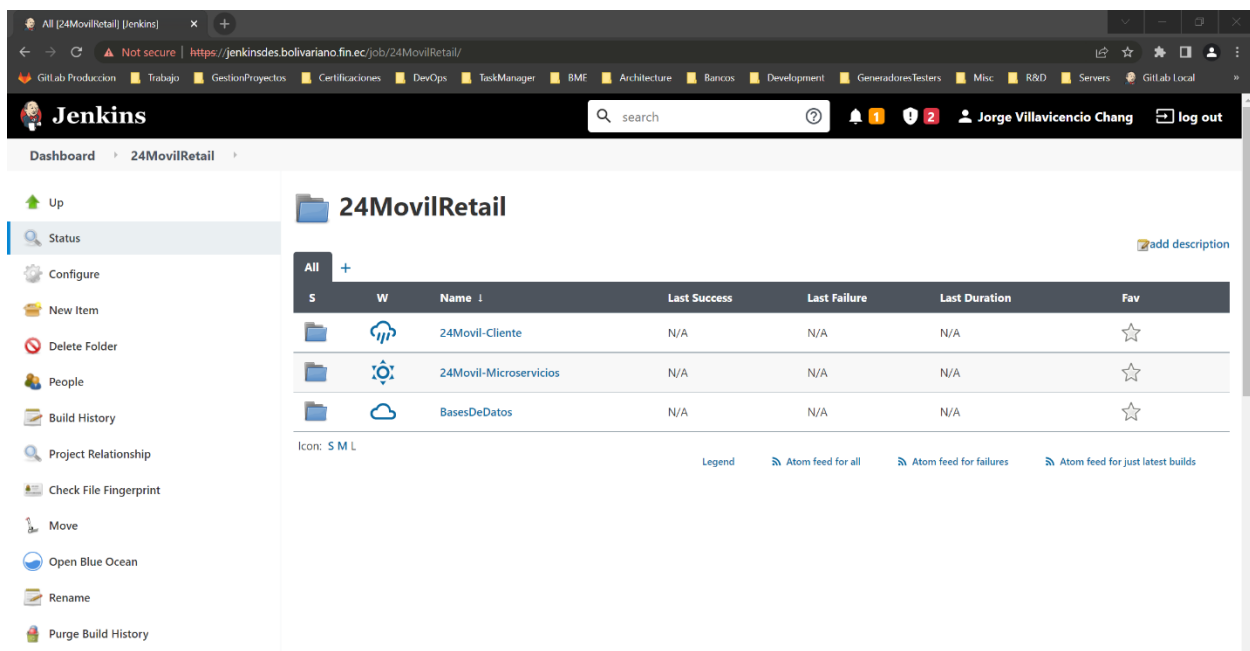
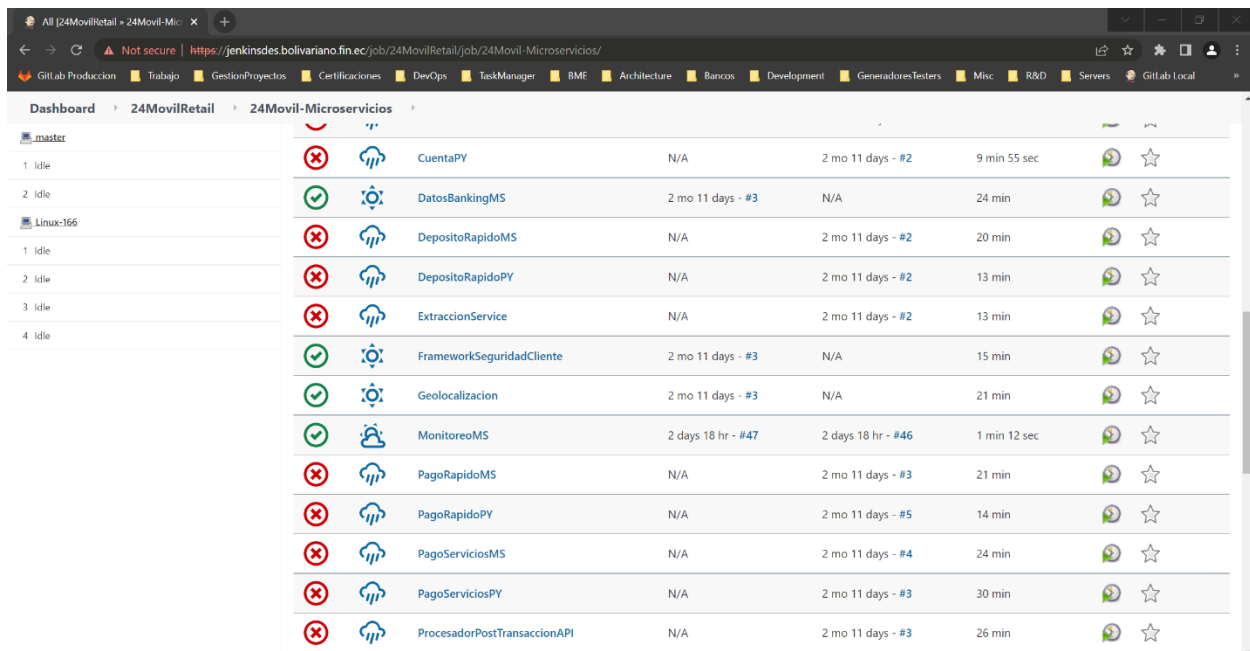


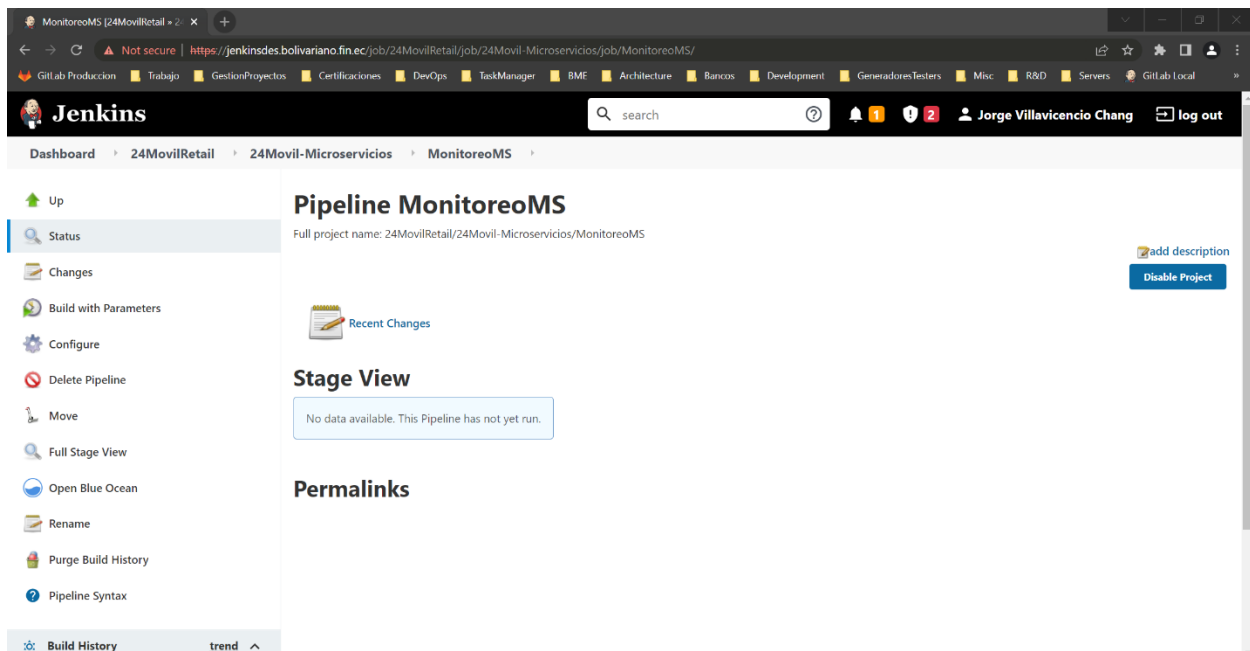
Ilustración 5 - Vista de contenido de Grupo Aplicativo, escoger de la lista microservicios



Build	Source	Build Name	Build Status	Build Number	Build Duration	Build Time
1	master	CuentaPY	Failed	N/A	2 mo 11 days - #2	9 min 55 sec
2	Linux-166	DatosBankingMS	Success	2 mo 11 days - #3	N/A	24 min
1	Linux-166	DepositoRapidoMS	Failed	N/A	2 mo 11 days - #2	20 min
2	Linux-166	DepositoRapidoPY	Failed	N/A	2 mo 11 days - #2	13 min
3	Linux-166	ExtraccionService	Failed	N/A	2 mo 11 days - #2	13 min
4	Linux-166	FrameworkSeguridadCliente	Success	2 mo 11 days - #3	N/A	15 min
		Geolocalizacion	Success	2 mo 11 days - #3	N/A	21 min
		MonitoreoMS	Success	2 days 18 hr - #47	2 days 18 hr - #46	1 min 12 sec
		PagoRapidoMS	Failed	N/A	2 mo 11 days - #3	21 min
		PagoRapidoPY	Failed	N/A	2 mo 11 days - #5	14 min
		PagoServiciosMS	Failed	N/A	2 mo 11 days - #4	24 min
		PagoServiciosPY	Failed	N/A	2 mo 11 days - #3	30 min
		ProcesadorPostTransaccionAPI	Failed	N/A	2 mo 11 days - #3	26 min

Ilustración 6 - Vista de pipelines disponibles para el producto y tipo de fuente (microservicios)

En caso de requerir ejecutar manualmente el pipeline de su microservicio, puede ingresar al mismo en Jenkins y dar clic a la opción “Build with Parameters” sin ingresar más información. Aparecerá un check desmarcado que indicará “desplegar” y dos campos de ingreso de datos. No marque ni llene ninguno de esos campos, esto correrá el pipeline en “modo continuo”.



**Pipeline Monitor**

Full project name: 24MovilRetail/24Movil-Microservicios/MonitoreoMS

**Recent Changes**

**Stage View**

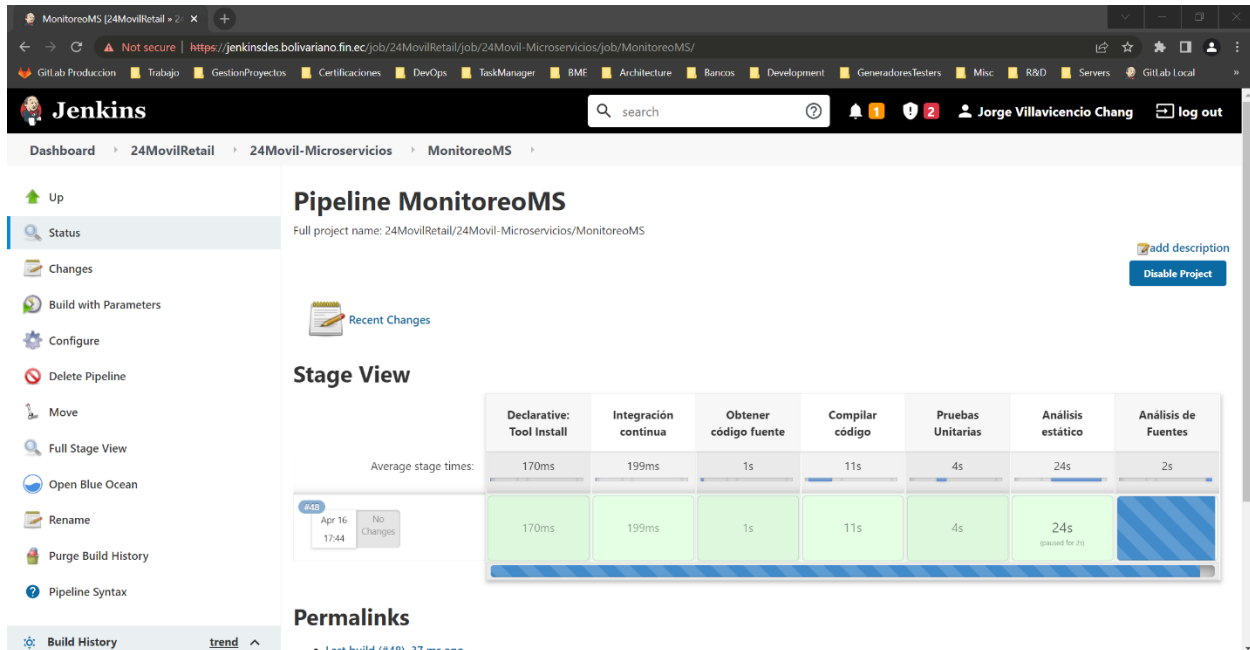
No data available. This Pipeline has not yet run.

**Permalinks**

**Build History**

Ilustración 7 - Vista general del pipeline

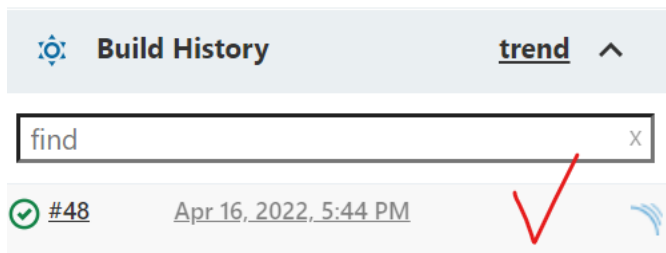
Se mostrará el avance del pipeline y el éxito o fracaso de este:



The screenshot shows the Jenkins web interface for the 'Pipeline MonitoreoMS' project. The left sidebar contains navigation options like 'Up', 'Status', 'Changes', 'Build with Parameters', 'Configure', 'Delete Pipeline', 'Move', 'Full Stage View', 'Open Blue Ocean', 'Rename', 'Purge Build History', and 'Pipeline Syntax'. The main area displays the 'Pipeline MonitoreoMS' stage view, showing a table of stages and their durations. Below the table, there is a 'Permalinks' section with a link to the last build.

Stage	Declarative: Tool Install	Integración continua	Obtener código fuente	Compilar código	Pruebas Unitarias	Análisis estático	Análisis de Fuentes
Average stage times:	170ms	199ms	1s	11s	4s	24s	2s
#48 (Apr 16 17:44)	170ms	199ms	1s	11s	4s	24s (paused for 2s)	

Ilustración 8 - Vista de corrida de pipeline



The screenshot shows the 'Build History' section of the Jenkins interface. It features a search bar with the text 'find' and a 'trend' button. Below the search bar, there is a list of builds. The first build, #48, is highlighted with a green checkmark and a red checkmark, indicating a successful build. The build date and time are 'Apr 16, 2022, 5:44 PM'.

Ilustración 9 - Estado de corrida, en este caso exitosa

Cuando ya tenga un alto nivel de certeza de su cambio y quiera liberar su tarea, combine todos sus cambios al Branch “desarrollo” y haga el push directo a Gitlab en el ramal “desarrollo” remoto. Recuerde eliminar su brach feature temporal localmente y en el servidor de Gitlab.

Ahora deberá ingresar a Jenkins (<https://jenkinsdes.bolivariano.fin.ec>) con su usuario y clave de matriz, ubicarse en su pipeline y dar clic en “Build with Parameters”. Esta vez marque el check “Liberar” e ingrese tanto el código de tarea JIRA y la descripción de la misma y proceda a correr. Esta acción corre el pipeline en modo de “liberación”, la cual realiza todas las acciones descritas en la sección de Generalidades de este documento.

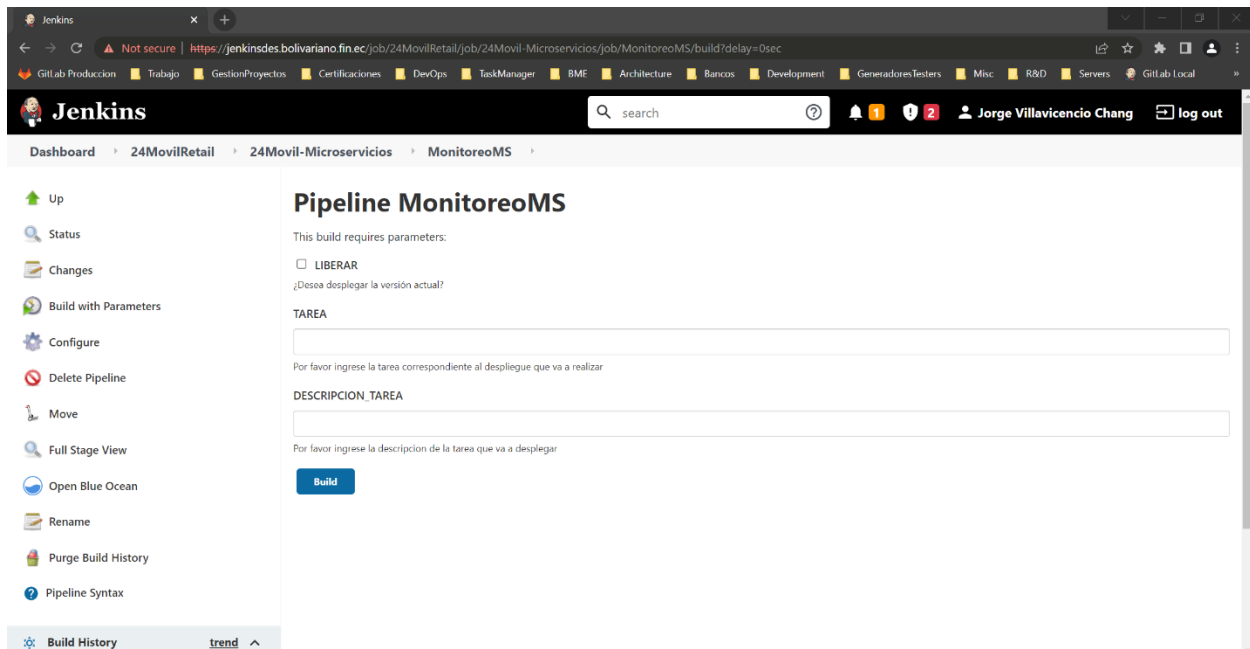


Ilustración 10 - Pantalla de corrida de pipeline y seleccion de accion

En este punto usted deberá recibir tres correos (asumiendo que se han pasado todas las pruebas y ha recibido la aprobación automática de las áreas de control), uno con el documento de matriz unitaria, el OK de Ing. Seguridad (Veracode) y el OK de ApoyoQA (Sonarqube). Estos documentos deberá luego subirlos manualmente como evidencia en Confluence siguiendo los flujos regulares de aprobación de pases con A&C.

Al final de la corrida recibirá adicionalmente un correo resumen indicando la finalización correcta del proceso.

## Preguntas Comunes

*¿Porque estamos adoptando esta práctica de integración continua?*

Puede encontrar una explicación completa en esta dirección

<https://smartbear.com/learn/automated-testing/the-continuous-development-pipeline/>

En términos generales la practica nos permite eliminar de nuestros procesos de construcción pasos manuales que pudieran introducir error durante el tiempo de desarrollo y a la vez incorporar nuevos controles de calidad de forma automática a la vez que se reducen los tiempos de actividades no relacionadas a la codificación.

*¿Por qué solo microservicios?*

Pensamos cubrir la mayor cantidad de tipos de desarrollos en la medida de lo posible, incluyendo fuentes de base de datos, GAW, SOA, frontends web, etc.

Como punto de partida estamos iniciando con microservicios en las líneas de 24Movil y OnBoarding, sin embargo, iremos expandiendo los tipos de fuentes según el plan de adopción incremental definido por el Área de Tecnología.

*¿Por qué no se cubren los despliegues de la aplicación en OpenShift?*

La etapa de despliegue corresponde a las practicas de entrega y despliegue continuo, por lo pronto estamos probando nuestros fuentes y generando los artefactos para prepararlos para fases posteriores que cubren despliegue en ambientes.

*¿Quién administra, mantiene y crea los pipelines de integración continua de banco?*

El esquema esta pensado para que nuestros desarrolladores tengan el control de los pipelines de sus respectivos grupos, de esta forma cada grupo podrá implementar sus automatizaciones como conocedores del alcance de sus desarrollos. Por ejemplo, para un fuente de frontend de react, se deben correr tests unitarios de una forma diferente a lo que aplicaría para un fuente java con gradle o Maven.

*¿Por qué no hacer las cosas como están actualmente definidas en el banco?*

Parte del plan estratégico de Tecnología incluye adoptar prácticas y modelos que nos permitan lograr una ventaja competitiva en el mercado mediante la optimización continua de procesos y reducción del Time to Market. Es nuestra obligación como equipo de desarrollo buscar mecanismos que nos permitan obtener resultados de calidad y un alto nivel de satisfacción de nuestros clientes.

## Ilustraciones

Ilustración 1 - Interacción de usuarios y servidores en el esquema de IC para microservicios .....	4
Ilustración 2 - Flujo Continuo.....	5
Ilustración 3 - Flujo de liberación de tareas.....	6
Ilustración 4 - Pantalla principal de Jenkins, vista de grupos aplicativos (actualmente 24MovilRetail y OnBoarding) .....	9
Ilustración 5 - Vista de contenido de Grupo Aplicativo, escoger de la lista microservicios .....	9
Ilustración 6 - Vista de pipelines disponibles para el producto y tipo de fuente (microservicios) .....	10
Ilustración 7 - Vista general del pipeline.....	10
Ilustración 8 - Vista de corrida de pipeline .....	11



Ilustración 9 - Estado de corrida, en este caso exitosa.....	11
Ilustración 10 - Pantalla de corrida de pipeline y seleccion de accion .....	12