**Estrategia para Etiquetado de repositorios de código Fuente**

**V.1**

**Centro de Excelencia DevOps**

**AVAL 2020.**

**Control de Cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Estado Proceso** | En Revisión / Aprobado |  |  |
| **Autor Documento** | [@Gomez Castillo Diana (Coordinacion de Calidad)](mailto:POR09250@porvenir.com.co) | **Fecha de Creación** |  |
| **Revisado por** |  | **Fecha de Revisión** |  |
| **Capitulo** | Estándares y Prácticas | **Área** |  |

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Comentarios** |
| 1.0 | 11 Noviembre 2020 | Prototipo en desarrollo |

**Tabla de contenido**

[Objetivo documento 4](#_Toc57200175)

[Conceptos básicos 5](#_Toc57200176)

[Repositorio de Código Fuente 5](#_Toc57200177)

[Etiquetado de Repositorio de Código Fuente 5](#_Toc57200178)

[Estrategia de etiquetado 6](#_Toc57200179)

[Nomenclatura 6](#_Toc57200180)

[Flujo básico para el etiquetado automático 7](#_Toc57200181)

[Etiquetado en Git 8](#_Toc57200182)

[Etiquetado en Subversión 12](#_Toc57200183)

[Etiquetado en Team Foundation Server 13](#_Toc57200184)

[Referencias 13](#_Toc57200185)

## Objetivo documento

Este documento describe la estrategia para el etiquetado de repositorios de código fuente que las entidades de Grupo AVAL pueden emplear como guía de implementación en sus proyectos de desarrollo de Software.

# Conceptos básicos

## Repositorio de Código Fuente

Mantiene un historial del desarrollo de código, y ayudan a resolver conflictos a la hora de combinar las contribuciones de diferentes orígenes. El control de código fuente es un componente esencial del proceso de desarrollo. Los sistemas de administración de código fuente permiten supervisar los cambios en el código, consultar el historial de revisiones del código y volver a versiones anteriores de un proyecto cuando sea necesario.

Con un repositorio de código fuente, puede trabajar en equipo en el código, aislar el trabajo hasta que esté listo e identificar quién realizó los cambios y qué cambios se realizaron para poder solucionar rápidamente los problemas. Los sistemas de administración de código de fuente ayudan a optimizar el proceso de desarrollo y proporcionan un origen centralizado para todo su código.

## Etiquetado de Repositorio de Código Fuente

Las etiquetas o tags son una forma de etiquetar estados del repositorio, que se usa comúnmente para indicar las versiones o releases de un proyecto mantenido con Git, especificando puntos o commits específicos dentro de los repositorios.

Git tiene la posibilidad de marcar estados importantes en el ciclo de vida de un repositorio, algo que se suele usar habitualmente para el manejo de las releases de un proyecto. A través del comando "git tag" es posible crear etiquetas, en una operación que se conoce comúnmente con el nombre de "tagging". Es una operativa que tiene muchas variantes y utilidades.

# Estrategia de etiquetado

Para desarrollar un proceso de etiquetado de repositorio de código fuente es necesario considerar como mínimo:

* Nomenclatura
* Flujo básico para el etiquetado automático
* Etiquetado en Git
* Etiquetado en Subversion
* Etiquetado en Team Foundation Server

## Nomenclatura

Utilizar nomenclatura Semantinc Versión

* Nomenclatura
  + Major
    - India la versión principal del desarrollo que conjunta funcionalidades concretas para la versión.
    - Incremente cuando se realizan cambios que afectan la compatibilidad de las dependencias.
  + Minor
    - Indica funcionalidad concreta dentro de la versión de desarrollo entregado
    - Incrementa cuando se realizan cambios que no afectan la compatibilidad de las dependencias
  + Patch
    - Incrementa al solucionar un Bug
  + Variables
    - d: para la rama de desarrollo
    - q: para la rama que indica el ambiente de QA
    - u: para la rama que indica el ambiente de UAT
* Ejemplos
  + 1.0.1d
    - 1 major
    - 0 minor
    - 1 patch
    - d desarrollo
  + 2.1.3u
    - 2 major
    - 1 minor
    - 3 path
    - u UAT

## Flujo básico para el etiquetado automático

Utilizando el disparador de compilación de Poll SCM en Jenkins y acceder a la sección SCM en el panel de configuración de trabajos de canalización para ingresar las siguientes sentencias que permite evidenciar nuevos flujos de trabajo en Jenkins a partir de las etiquetas:

* Refspec: +refs/tags/\*:refs/remotes/origin/tags/\*
* Branch: specifier \*\*/tags/\*\*

***Utilización de tags en el pipeline***

Al habilitar dentro del stage la opción de validar el tag reléase-\* se condiciona el despliegue a tener una etiqueta en él, como es el caso de:

pipeline {

agent any

stages {

stage('Build') {

steps {

sh 'make package'

}

}

stage('Test') {

steps {

sh 'make check'

}

}

stage('Deploy') {

when { tag "release-\*" }

steps {

echo 'Deploying only because this commit is tagged...'

sh 'make deploy'

}

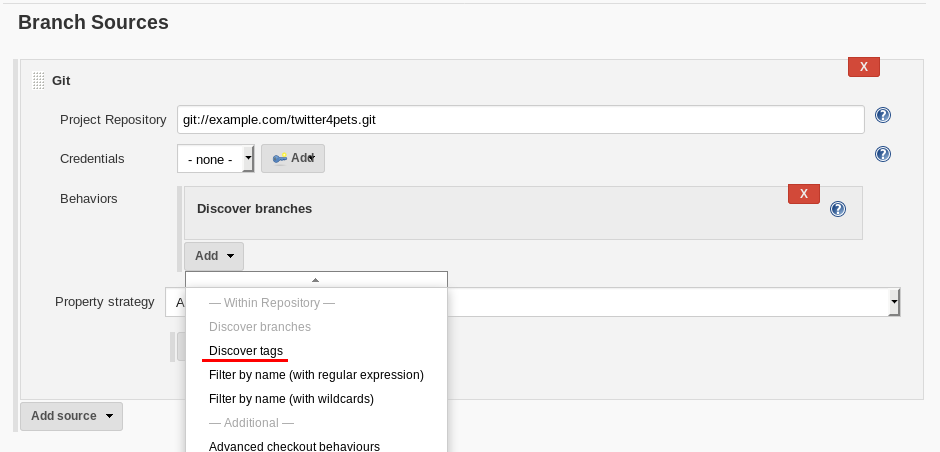
}

}

}

***Configuración.***

Desde la pantalla de configuración de Multibranch de pipeline, discover tags puede ser activado agregando el “comportamiento” apropiado a la configuración:



Con estos cambios en el Jenkinsfile en el etiquetado de las versiones del repositorio ahora puede impulsar un comportamiento distinto de implementación.

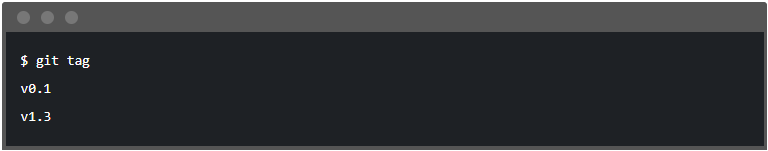
## Etiquetado en Git

Git nos permite etiquetar (tag) puntos específicos en la historia como importantes. Generalmente la gente usa esta funcionalidad para marcar puntos donde se ha lanzado alguna versión (v1.0, y así sucesivamente).

* **Listar etiquetas**

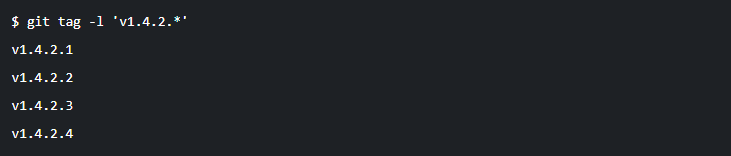
Con el comando git tag listamos las etiquetas que tenemos disponibles

$ git tag



Adicionando el parámetro –l podemos buscar etiquetas de acuerdo a un patrón en particular. Por ejemplo, si un repositorio contiene una considerable cantidad de etiquetas y solo estamos interesado en la serie 1.4.2, podemos buscar un patrón

$ git tag -l 'v1.4.2\*'



* **Creando etiquetas**

Git usa dos tipos principales de etiquetas: ligeras y anotadas.

Hay dos tipos de etiquetas: ligeras (lightweight) y anotadas (annotated). Una etiqueta ligera almacena únicamente el commit sobre el que se creó la etiqueta, mientras que una etiqueta anotada almacena también la información de la persona que la creó, la fecha y un mensaje. La recomendación es siempre crear **etiquetas anotadas**.

* + **Etiquetas anotadas**

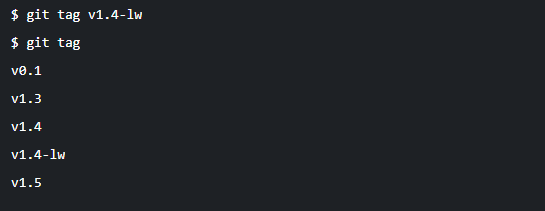
Para crear una etiqueta una etiqueta anotada utilizamos el parámetro –a

git tag -a v1.4.2 -m 'Version 1.4.2'

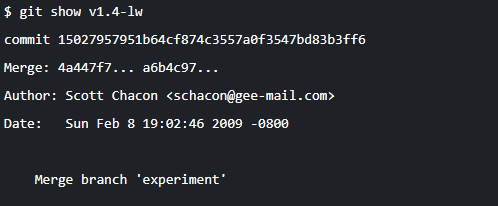
El nombre va después de la opción -a, en este caso v1.4.2. La opción -m define el mensaje que le vamos a adjuntar a la etiqueta, si se omite se abre el editor de texto para ingresarlo.

* + **Etiquetas ligeras**

Otra forma de etiquetar confirmaciones es con una etiqueta ligera. Esto es básicamente la suma de comprobación de la confirmación almacenada en un archivo — ninguna otra información es guardada —. Para crear una etiqueta ligera no añadas las opciones -a, -s o –m

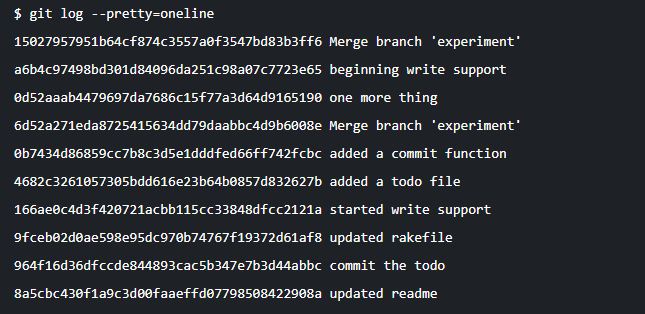


Esta vez, si ejecutas el comando git show en la etiqueta, no se vera ninguna información extra. El comando simplemente muestra la confirmación.



* **Etiquetado tardío**

Git ofrece la posibilidad de etiquetar **commits** mucho tiempo después de haberlos hecho. Supongamos que tu historial luce como el siguiente:

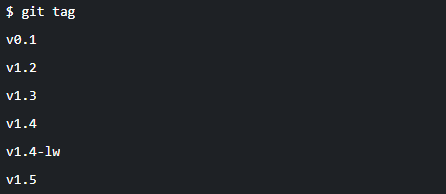


En caso que se olvidara etiquetar el proyecto en v1.2, que estaba en la confirmación "updated rakefile". Para etiquetar esa confirmación especifica se asigna con el siguiente comando añadiendo sobre el final el identificador del commit o una parte de el

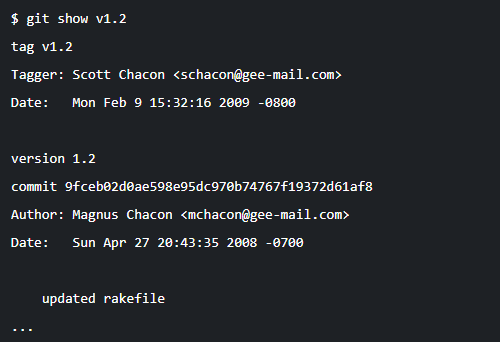
*git tag -a v1.2 9fceb02*

Para revisar que se etiqueto correctamente con el comando

$ git tag



$ git show v1.2



## Etiquetado en Subversión

Paso a paso para la generación de etiquetas en Subversión

Los Tags, permiten crear una captura (snapshot) de un proyecto en una determinada etapa, por lo que no son usados para desarrollo. Ya que son las ramas las usadas para trabajar en versiones nuevas.

Crear un Tag en Subversion funciona de la misma manera que crear una Rama, la única diferencia es que el Tag será ubicado dentro del directorio de tags, en vez del directorio de ramas. Por lo que no hay diferencia entre ambas, sino solamente que los tags nunca cambian y se guardan en un directorio diferente.

1. **Crear un directorio de tags, si no está creado:**

*svn mkdir tags*

Se debe crear el directorio de /tags, en caso de que no exista aún.

1. **Crear una copia del proyecto usando “svn copy”:**

*$ svn copy http://svn.example.com/repos/project/trunk \*

*http://svn.example.com/repos/project/tags/release-1.0 \*

*-m “Tagging the 1.0 release of the ‘project’ Project.”*

*Committed revisión 902*

Se debe crear una copia del proyecto trunk en el directorio de /tags, para crear el snapshot del proyecto.

## Etiquetado en Team Foundation Server

En **Source Control Explorer**, abrir el menú de acceso directo de la colección, proyecto, rama, carpeta o archivo al que desea aplicar una etiqueta y elija **Avanzado**, **Aplicar etiqueta**.

La **nueva etiqueta** aparece el cuadro de diálogo.

1. En el cuadro **Nombre**, escriba un nombre para la etiqueta.
2. (Opcional) Escriba sus comentarios en el cuadro **Comentario**.
3. En la lista **Versión**, la **última versión** está seleccionada de forma predeterminada. Esta opción es la que se usa con más frecuencia, pero si necesita la etiqueta para apuntar a una versión específica, seleccione:
   * **Conjunto de cambios**: especifique el número del conjunto de cambios en el cuadro **Conjunto de** cambios. Alternativamente, elija las elipses
   * **Fecha**: especifique una fecha en el cuadro **Fecha** o seleccione una fecha del calendario en el menú desplegable.
   * **Etiqueta**: especifique una etiqueta existente en la que basar la nueva etiqueta. Escriba el nombre de la **etiqueta** en el cuadro **Etiqueta** o elija los puntos suspensivos.
   * **Versión del espacio de trabajo**: la etiqueta se crea para la versión en su espacio de trabajo.
4. Cuando esté satisfecho con su configuración, realice uno de los siguientes pasos:
   * Para aplicar la etiqueta, elija **Crear**.
   * Para aplicar la etiqueta y luego modificarla después de que se haya aplicado, elija **Crear y editar**. Esto aplica la etiqueta y luego muestra la ventana **Etiqueta**.

## Referencias

AWS, ¿Qué es control de código fuente? Tomado de: <https://aws.amazon.com/es/devops/source-control/>

Miguel Angel Alvarez, Especificar versiones en Git con tag. Tomado de: <https://desarrolloweb.com/articulos/especificar-versiones-git-tag.html>

Andrea Navarro, Uso de tags en git. Tomado de: <https://juncotic.com/tag-en-git/>

Jenkins, When using tags in Jenkins Pipeline?. Tomado de:

<https://www.jenkins.io/blog/2018/05/16/pipelines-with-git-tags/>

Gkourtzounis. How to trigger Jenkins when new Tags are created? Tomado de: <https://comquent.de/en/triggering-jenkins-when-new-tags-are-created/>