

Flujos de trabajo en los repositorios DGTIC

Contenido:

- [Cumplimiento normativo](#)
- [Histórico de cambios](#)
- [1. Introducción](#)
 - [1.1 Alcance](#)
- [2. Listado de pautas](#)
- [3. Referencias](#)

Resumen:



- **Versión:** v01r58
- **Fecha última actualización:** 17 jun 2024
- **Entrada en vigor desde:** 17 jun 2024

Cumplimiento normativo



Las normas expuestas son de obligado cumplimiento. La DGSIC podrá estudiar los casos excepcionales los cuales serán gestionados a través de los responsables del proyecto correspondiente y autorizados por el Área de Gobernanza de la DGSIC. Asimismo cualquier aspecto no recogido en estas normas deberá regirse en primera instancia por las guías técnicas correspondientes al esquema nacional de seguridad y esquema nacional de interoperabilidad según correspondencia y en su defecto a los marcos normativos y de desarrollo software establecidos por la Junta de Andalucía, debiendo ser puesto de manifiesto ante la DGSIC.

La DGSIC se reserva el derecho a la modificación de la norma sin previo aviso, tras lo cual, notificará del cambio a los actores implicados para su adopción inmediata según la planificación de cada proyecto.

En el caso de que algún actor considere conveniente y/o necesario el incumplimiento de alguna de las normas y /o recomendaciones, deberá aportar previamente la correspondiente justificación fehaciente documentada de la solución alternativa propuesta, así como toda aquella documentación que le sea requerida por la DGSIC para proceder a su validación técnica.

Contacto Arquitectura: l-arquitectura.stic@juntadeandalucia.es

Histórico de cambios

Los cambios en la normativa vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución. Ordenándose de mas recientes a menos recientes, prestando especial cuidado a las cabeceras de la tablas dónde se indican las fechas de entrada en vigor y versión.

Versión	v01r58	Fecha publicación	17 jun 2024	Fecha entrada en vigor	17 jun 2024
Alcance					
Se incluyen las siguientes puntualizaciones en la Pauta 14, sobre el manejo de la rama de soporte:					
a. La rama de soporte se crea desde la rama master. Concretamente desde el tag del que se desea mantener una versión en paralelo.					
b. La evolución del producto que vaya a mantenerse viva, estará contenida en las ramas develop y master. La de soporte se utilizará para aquel mantenimiento y/o evolución que tenga una fecha de finalización.					
c. La rama de soporte realizará funciones de "develop" y "master" para la evolución funcional de la aplicación.					
d. La rama support se crea siguiendo la nomenclatura support/version, teniendo versión como <u>máximo</u> 2 dígitos W.X. , donde W: Identifica el dígito mayor de la versión. X: Identifica el dígito medio de la versión. Será el jefe de proyecto quien decida el nombre de la rama y los dígitos de la misma.					
e. Los tags en la rama support continuarán siendo de 4 dígitos. Sonarqube trabaja con tags, por lo que el análisis de código no se verá afectado.					
f. Cuando se finalice la evolución tecnológica, si es necesario, habrá que "volcar" en develop, la rama soporte. Para hacerlo, se crearán previamente una rama de mejora en develop y una rama de mejora en support, haciendo el merge de dichas ramas y resolviendo los conflictos u adaptaciones que sean necesarios.					
g. Las ramas de soporte, nunca se cerrarán directamente sobre la rama develop.					

Versión	v01r57	Fecha publicación	16 jun 2021	Fecha entrada en vigor	3 may 2021
Alcance					
- Definición de Pautas y validaciones. - Se realizan las siguientes puntualizaciones sobre algunas pautas anteriormente definidas: - Se aclara la imposibilidad de release y hotfix en ramas de soporte. - Se añaden las siguientes pautas respecto al documento anterior cómo aclaraciones fruto de la mejora continua: - Está permitido entregar ramas internas sobre hotfix con la nomenclatura definida en el P5 para mejoras e historias de usuario, sólo en casos excepcionales cuando el desarrollo se considera urgente. Mas info - Las ramas de soporte, tagearán sobre la misma rama de soporte la versión prevista, sin hacer uso de ramas release.					

Versión	v01r56	Fecha publicación	4 oct 2017	Fecha entrada en vigor	4 oct 2017
Alcance					

- Versión inicial sobre normativa de git

1. Introducción

El objeto de esta normativa, es la definición de las pautas necesarias a seguir para la entrega de código fuente y la correcta ejecución de los procesos de integración continua. Basándose en las mejores prácticas actualmente aplicadas en la industria, se definen pautas de obligado cumplimiento con el objeto de garantizar la homogeneidad, trazabilidad y correcto funcionamiento de todos los procesos previos a la liberación del software en entornos previos y productos.

La normativa se ha organizado en pautas que describen las reglas de verificación necesarias a cumplir en la entrega de código fuente.

Para más detalles sobre el modo de articular todas las pautas a continuación expuestas puede consultar la [guía de ayuda a las entregas en la STIC](#).

1.1 Alcance

Esta normativa afecta a aquellos desarrollos corporativos que estén bajo el marco de gestión centralizada de la Dirección General de Sistemas de Información y Comunicaciones del SAS.

2. Listado de pautas

Ámbito: Commits	
Pauta	Descripción
P1	No se permitirá por norma general hacer commits en master, develop o ramas support directamente, para ello se deberán de usar las demás ramas (release, feature, hu) dentro del flujo de Gitflow.
P2	Cada commit, incluido en una feature, debe incluir en su comentario el código Jira de la issue que corresponda, en caso de no disponer de dicha traza se referenciará el código de solicitud CGES que desencadena el cambio.
P3	Cada commit, incluido en una release, debe incluir la referencia a la issue de tipo versión especificada en Jira a la cual afecta.
C1	Cada commit, incluido, debe ser atómico y pequeño, incluyendo solo cambios que están relacionados entre sí.
C2	Cada commit, incluirá un mensaje claro y descriptivo conteniendo una primera línea con un máximo de 70 caracteres dónde se da un resumen claro y conciso del cambio. Posteriormente se agregará una descripción resumida con carácter imperativo dónde se aclara "qué" a cambiado y "porqué"

Ámbito: Trazabilidad	
Pauta	Descripción
P4	Se ha de seguir el flujo adaptado de Gitflow para la STIC, dónde cada rama de mejora conlleva la entrega de una rama de tipo feature (definida en git-flow) con nombre en el formato [feature/{Codigo_Jira_Mejora}_{Descripcion}]. Dónde: a. Feature: (Literal) Identifica el tipo de rama que está definido por el flujo adaptado de Gitflow. b. Código_Jira_Mejora: (Variable) Identifica el código de la issue que está registrada en Jira y que hace referencia a la Mejora que se entrega en la rama. i. Excepción I: En caso de no existir una mejora asociada en Jira por cualquier motivo puede hacer uso del código de CGES que pueda identificar dicha mejora. c. Descripción: (Variable) Identifica una muy breve descripción libre donde se resume la mejora.

P5	 Alcance No aplica en proyectos en metodología tradicional (no ágil). Se ha de seguir el flujo adaptado de Gitflow para la STIC, donde cada rama de historia de usuario, conlleva la entrega de una rama de tipo feature (definida en git-flow) con nombre en el formato [feature/{Codigo_Jira_Mejora}/{Codigo_Jira_HistoriaUsuario}_{Descripcion}]. Dónde: a. Feature: (Literal) Identifica el tipo de rama que está definido por el flujo adaptado de GitFlow. b. Código_Jira_Mejora: (Variable) Identifica el código de la issue que está registrada en Jira y que hace referencia a la Mejora que se entrega en la rama. i. Excepción I: En caso de no existir una mejora asociada en Jira por cualquier motivo puede hacer uso del código de CGES que pueda identificar dicha mejora. c. Código_Jira_HistoriaUsuario: Variable que identifica el código de la issue que está registrada en Jira y que hace referencia a una Historia de Usuario que se entrega en la rama. d. Descripción: Variable que identifica una muy breve descripción libre donde se resume la historia de usuario.
P6	Se ha de seguir el flujo adaptado de Gitflow para la STIC, dónde cada rama de Release cumplirá con la siguiente nomenclatura: [Release/{Version}]. Dónde: a. Versión: Corresponde a la versión expresada en la issue de Jira Versión, que identifica la versión que se va a liberar para entornos de preproducción y producción. Siguiendo el formato de 4 dígitos W.X.Y.Z
P7	Se ha de seguir el flujo adaptado de Gitflow para la STIC, donde cada rama de Hotfix cumplirá con la siguiente nomenclatura: [Hotfix/{Version}]. Dónde: a. Versión: Corresponde a la versión expresada en la issue de Jira Versión, que identifica la versión que se va a liberar para entornos de preproducción y producción conteniendo el hotfix. Siguiendo el formato de 4 dígitos W.X.Y.Z
P8	Se ha de seguir el flujo adaptado de Gitflow para la STIC, donde cada rama de soporte cumplirá con la siguiente nomenclatura: [Support/{Version}]. Dónde: a. Versión: Corresponde a la versión que se mantiene en soporte y que corresponderá con el valor de la versión que haya en el commit desde el que se abre la rama. b. Versión tiene como máximo 2 dígitos W.X. , donde W: Identifica el dígito mayor de la versión. X: Identifica el dígito medio de la versión. Será el jefe de proyecto quien decida el nombre de la rama y los dígitos de la misma. c. Los tags en la rama support continuarán siendo de 4 dígitos W.X.Y.Z. Sonarqube trabaja con tags, por lo que el análisis de código no se verá afectado.

Ámbito: Gestión de ramas en proyectos basados en metodologías ágiles

Pauta	Descripción
P9	Se ha de seguir el flujo adaptado de Gitflow para la STIC cuando genere una rama para una mejora. El formato del nombre está definido en P4 .
P10	Se ha de seguir el flujo adaptado de Gitflow para la STIC cuando genere una rama para una historia de usuario. El formato del nombre está definido en P5 .
P11	Las ramas de historia de usuario se generan y cierran sobre las ramas de mejora en la que están contenidas, con la aceptación de los siguientes criterios al cierre: a. Aprobación del DoD: Una vez el DoD está aprobado.
P12	Las ramas de Mejora permanecerán abiertas siempre, se cerrarán sobre la rama develop tras la aceptación de los siguientes criterios: a. Inclusión de la mejora en versión: Una vez la mejora se incluye para la instalación en una versión.
P13	Las ramas de soporte, permanecerán siempre abiertas, en paralelo a la rama develop, y se comportará como tal, permitiendo la apertura sobre ella de ramas de mejora, historia de usuario.-
P14	Manejo de la rama support en escenarios que requieren versiones en paralelo (por ejemplo, evolución de la aplicación y evolución tecnológica ejecutadas en paralelo) a. La rama de soporte se crea desde la rama master. Concretamente desde el tag del que se desea mantener una versión en paralelo. b. La evolución del producto que vaya a mantenerse viva, estará contenida en las ramas develop y master. La de soporte se utilizará para aquel mantenimiento y/o evolución que tenga una fecha de finalización. c. La rama de soporte realizará funciones de "develop" y "master" para la evolución funcional de la aplicación. d. La rama support se crea siguiendo la nomenclatura support/version, teniendo versión como <u>máximo</u> 2 dígitos W.X. , donde W: Identifica el dígito mayor de la versión. X: Identifica el dígito medio de la versión. Será el jefe de proyecto quien decida el nombre de la rama y los dígitos de la misma. e. Los tags en la rama support continuarán siendo de 4 dígitos. Sonarqube trabaja con tags, por lo que el análisis de código no se verá afectado. f. Cuando se finalice la evolución tecnológica, si es necesario, habrá que "volcar" en develop, la rama soporte. Para hacerlo, se crearán previamente una rama de mejora en develop y una rama de mejora en support, haciendo el merge de dichas ramas y resolviendo los conflictos u adaptaciones que sean necesarios. g. Las ramas de soporte, nunca se cerrarán directamente sobre la rama develop.
P15	Las ramas hotfix o release se registrarán por las nomenclaturas P8 , P7 y P6 respectivamente.
P16	Está permitido entregar ramas internas sobre hotfix con la nomenclatura definida en el P5 para mejoras e historias de usuario, sólo en casos excepcionales cuando el desarrollo se considera urgente. Más info

Ámbito: Gestión de ramas en proyectos basados en metodologías tradicionales (no ágil)

Pauta	Descripción
P17	Vease P9.
P18	Vease P12.

P19	Vease P13.
P20	Vease P14.
P21	Vease P15.

Ámbito: Gestión de tags

Pauta	Descripción
P22	Nunca se nombrarán tags con el sufijo "-Snapshot" o variantes en las ramas master, support.
P23	El formato permitido para un tag siempre será del tipo W.X.Y.Z, dónde: w. Identifica el dígito mayor de la versión. X. Identifica el dígito medio de la versión. Y. Identifica el dígito menor de la versión. Z. Identifica el dígito de entrega de la versión.
P24	Excepcionalmente y bajo acuerdo previo entre el área de gobernanza y el proyecto se permitirán versiones de sprint identificadas con el sufijo SP"n", dónde n identifica el número secuencial del sprint al que representa.
P25	Excepcionalmente y bajo acuerdo previo entre el área de gobernanza y el proyecto se permitirán versiones candidatas a release, identificadas por el sufijo RC"n", dónde "n" identifica un secuencial exclusivo para las candidatas a release.
P26	Las ramas de soporte, tagearán sobre la misma rama de soporte la versión prevista, sin hacer uso de ramas release.
P27	En proyectos que empaquetan haciendo uso de Maven, al tagear sobre el repositorio de código fuente la versión identificada en los artefactos generados ha de ser coincidente con la versión tageada, siguiendo el formato de versión indicado en P23 .
P28	En proyectos que empaquetan haciendo uso de node.js, al tagear sobre el repositorio de código fuente, la versión identificada en los artefactos generados ha de ser coincidente con la versión tageada, siguiendo un formato de 3 dígitos similar al de P23 y prescindiendo del dígito Z.

Ámbito: Gestión de la entrega de código

Pauta	Descripción
P29	Las entregas al repositorio de la STIC sólo soportarán como ramas abiertas aquellas que sean del tipo, soporte, develop o master.
P30	Las ramas de tipo hotfix o release, así como cualquier otra rama que haya sido de uso interno en su repositorio empresarial, deben haberse cerrado correctamente sobre las ramas de soporte o develop, previamente a su entrega al repositorio de la STIC.
P31	Al entregar, la rama MASTER estará como máximo al mismo nivel que la rama DEVELOP, nunca la rama MASTER podrá estar por encima de la rama DEVELOP. En todo caso la rama DEVELOP sí que podrá estar por encima de la rama MASTER, dado su carácter de rama de trabajo.
P32	Cuando se realice una entrega de una nueva versión, deberá existir el TAG (desde la rama MASTER) correspondiente a dicha versión.
P33	Nunca se entregarán tags al repositorio de la stic que no cumplan escrupulosamente con las reglas definidas en P22 .

3. Referencias

- [Guia de entregas de código fuente en la DGSIC](#)