

Gestión de dependencias durante el desarrollo en proyectos ágiles

Subdirección de las Tecnologías de la Información y Comunicaciones

Área de Gobernanza y Calidad



Contenido

- [Histórico de cambios](#)
- [Introducción](#)
- [Gestión de dependencias](#)
- [Mismo módulo](#)
- [Diferentes módulos](#)

Resumen



- **Versión:** v01r01
- **Fecha publicación:** 29 de Octubre de 2020

Histórico de cambios

Los cambios en la documentación de apoyo vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución.

Versión	v01r01	Fecha publicación	29 de Octubre de 2020
Alcance			
• <u>Descripción de los procesos GitFlow para la gestión de dependencias entre historias de usuario (a partir de ahora HU), que debe seguir el equipo de desarrollo de un proyecto de gestión ágil</u>			

Introducción

En esta página se describirán los procesos para la gestión de dependencias entre historias de usuario (a partir de ahora HU), que debe seguir el equipo de desarrollo de un proyecto de gestión ágil. Las situaciones que se abordan son:

- Sin dependencias con otra HU.
 - Dependencia con una HU de otro proyecto.
 - Dependencia con un desarrollo dentro del mismo proyecto.
 - Mismo módulo.
 - Módulos diferentes
 - Ambos módulos dentro de un proyecto de gestión ágil.
 - uno de los módulos no se encuentra dentro de un proyecto de gestión ágil.
-

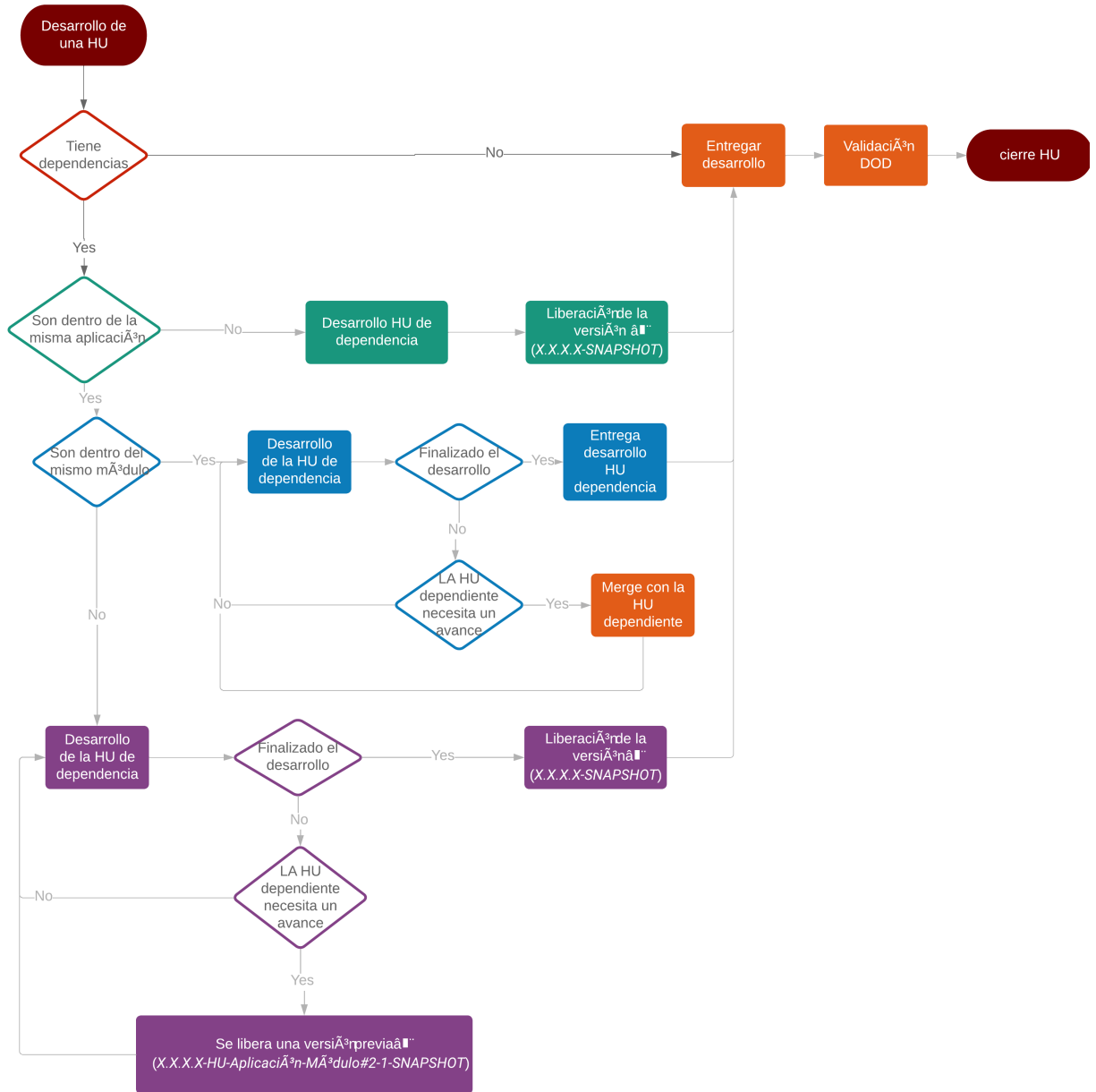
Gestión de dependencias

En esta sección se abordara de forma global el desarrollo de una HU para cualquier situación de las descritas en la sección introductoria. Con este flujo se pretenden sincronizar los desarrollos de las HU dependientes con los desarrollos de las HU de dependencia. Esto permitirá a las herramientas de ALM comportarse de manera satisfactoria.

[Editar](#)

Diagrama Flujo de dependencias

| February 8, 2024



La leyenda de colores del diagrama que se muestra a continuación es la siguiente:

Color	Situación
	Sin dependencias con otra HU
	Dependencia con una HU de otro proyecto
	Dependencias con uno de los módulos que no se encuentra dentro de un proyecto de gestión ágil
	Dependencia con otro desarrollo del mismo módulo

	Dependencias entre dos módulos diferentes, estando ambos módulos dentro de un proyecto ágil
--	---

Nota sobre las dependencias en las que al menos un módulo no se encuentra dentro de un proyecto de gestión ágil

A efectos de gestión de dependencias se considerará como si fuera una dependencia de otro proyectos, debido a que los tiempos de desarrollo y ejecución pueden no estar sincronizados. En este caso desde Arquitectura se recomienda primero abordar el desarrollo de la HU de dependencia del proyecto fuera de la gestión ágil y después abordar el desarrollo de la HU del proyecto ágil. De esta forma se pretende sincronizar los tiempos de desarrollo para que encajen en el sprint del proyecto ágil.

Mismo módulo

En esta sección se entrará a definir con más detalle la gestión de una HU que tiene dependencia con un cambio en dentro del mismo módulo del proyecto, haciendo hincapié en el uso del git para este tipo de desarrollos.

Flujo para el desarrollo de una HU

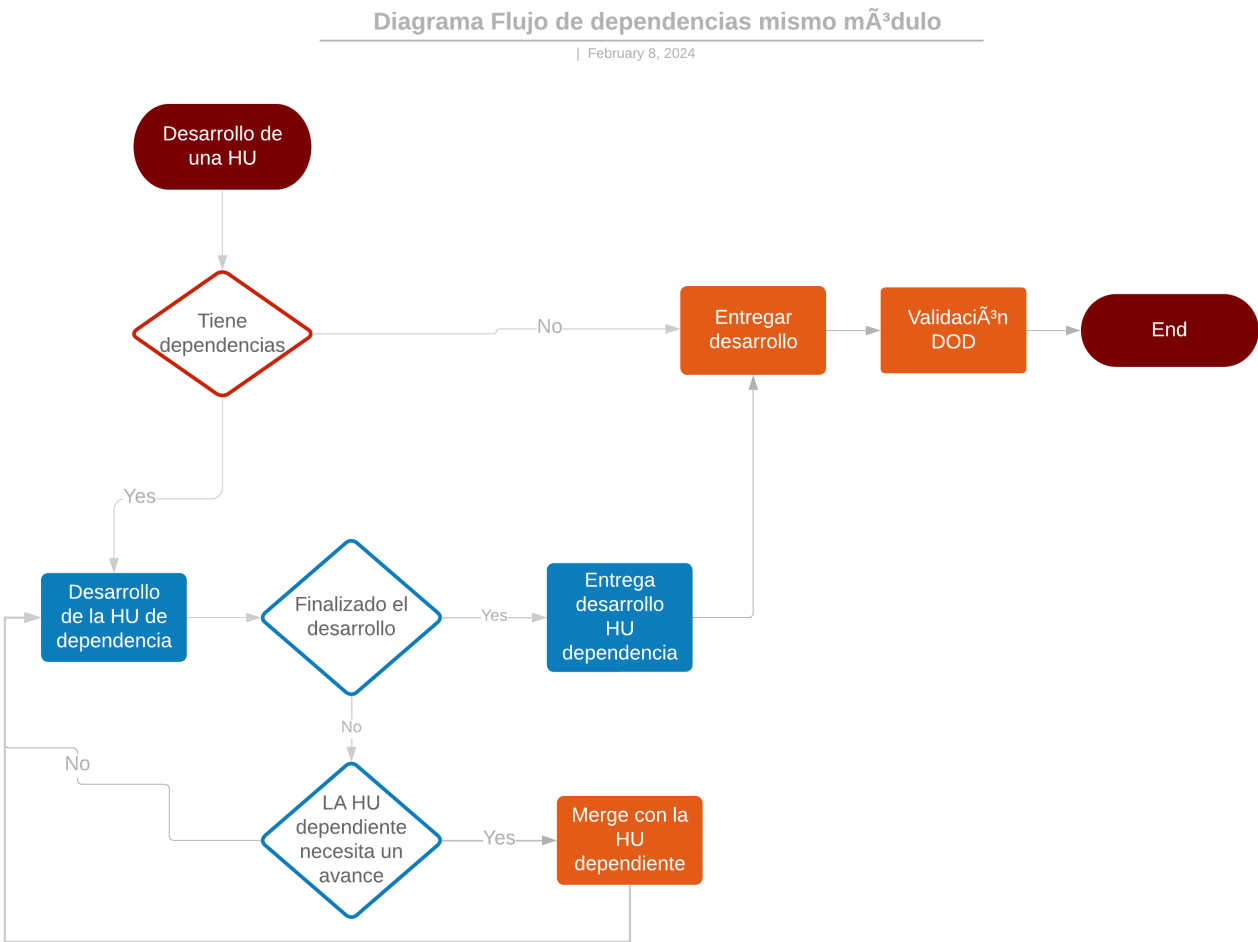


Diagram attachment access error: cannot display diagram

Gestión en Git de las HU's

Los pasos que se dan, asumiendo que la historia de usuario dependiente **(HU-Aplicación-Módulo#1-1)** ya está creada, son:

Se crea una HU **(HU-Aplicación-Módulo#1-2)** partiendo de la mejora que corresponda.

Cuando en la HU dependiente **(HU-Aplicación-Módulo#1-1)** se necesite algún avance realizado en las HU de dependencia **(HU-Aplicación-Módulo#1-2)**, este avance se puede mergear sobre la HU dependiente **(HU-Aplicación-Módulo#1-2 ---merge HU-Aplicación-Módulo#1-1)**.

Una vez terminada la historia de usuario dependiente **(HU-Aplicación-Módulo#1-1)** , se entrega (siempre después de haber completado la dependencia) y se sigue el flujo normal de entrega de una HU.

Nota

Para poder trabajar en paralelo, se aconseja que se definan los contratos de dependencia previos a los desarrollos (interfaces o funciones vacías) y que se usen mocks que contemplen los diferentes casos de uso de la HU. Con esto se pretende aumentar la eficacia de la integración de ambos módulos y mejorar la velocidad de desarrollo.

*El cierre las HU dependientes **(HU-Aplicación-Módulo#1-2)**, queda sujeto a la aprobación de la HU dependiente **(HU-Aplicación-Módulo#1-1)**.*

Diagrama Git

| February 8, 2024

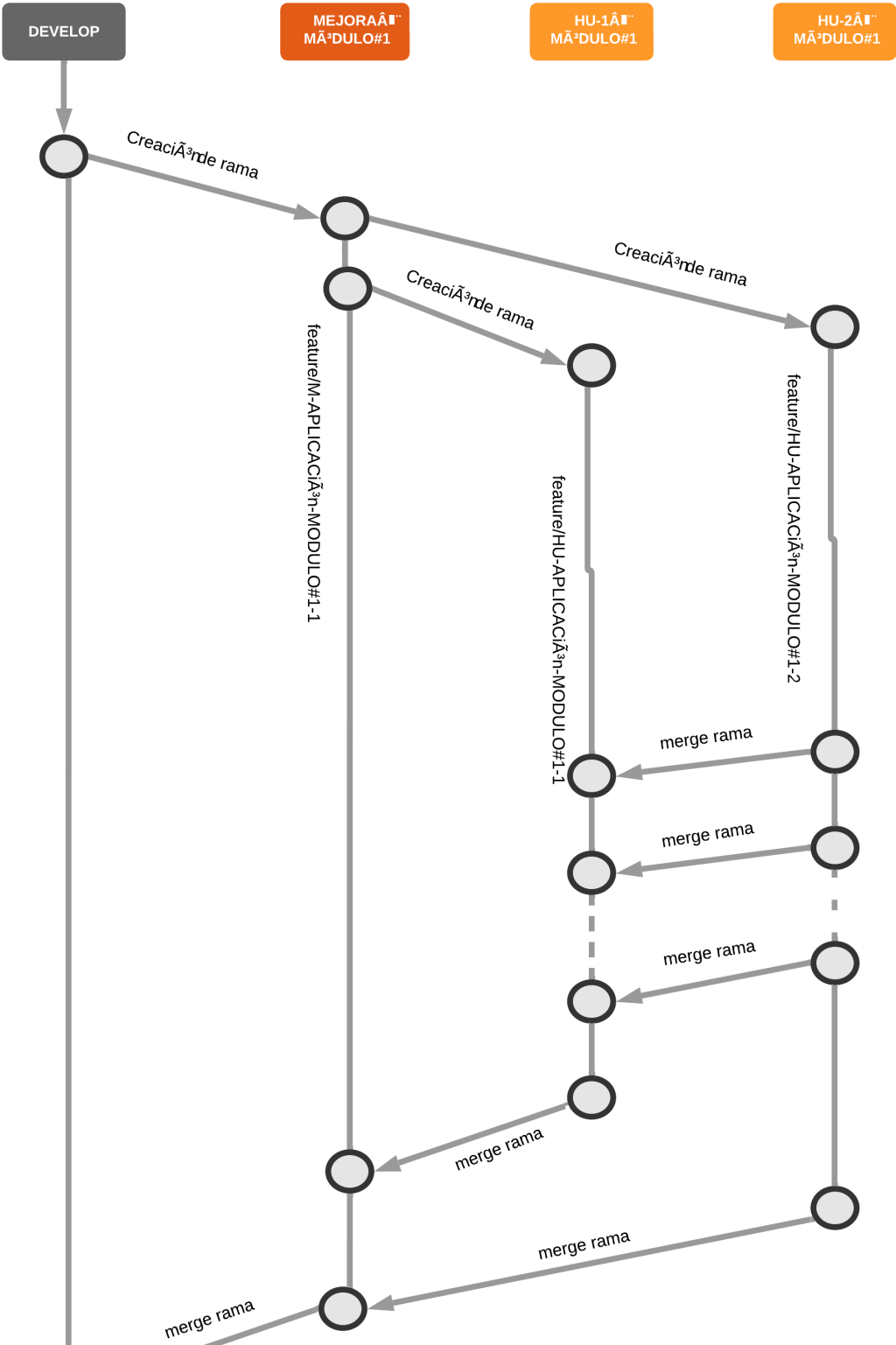




Diagram attachment access error: cannot display diagram

Diferentes módulos

En esta sección se entrará a definir con más detalle la gestión de una HU que tiene dependencia con un cambio en otro módulo del mismo proyecto, haciendo hincapié en el uso del git para este tipo de desarrollos.

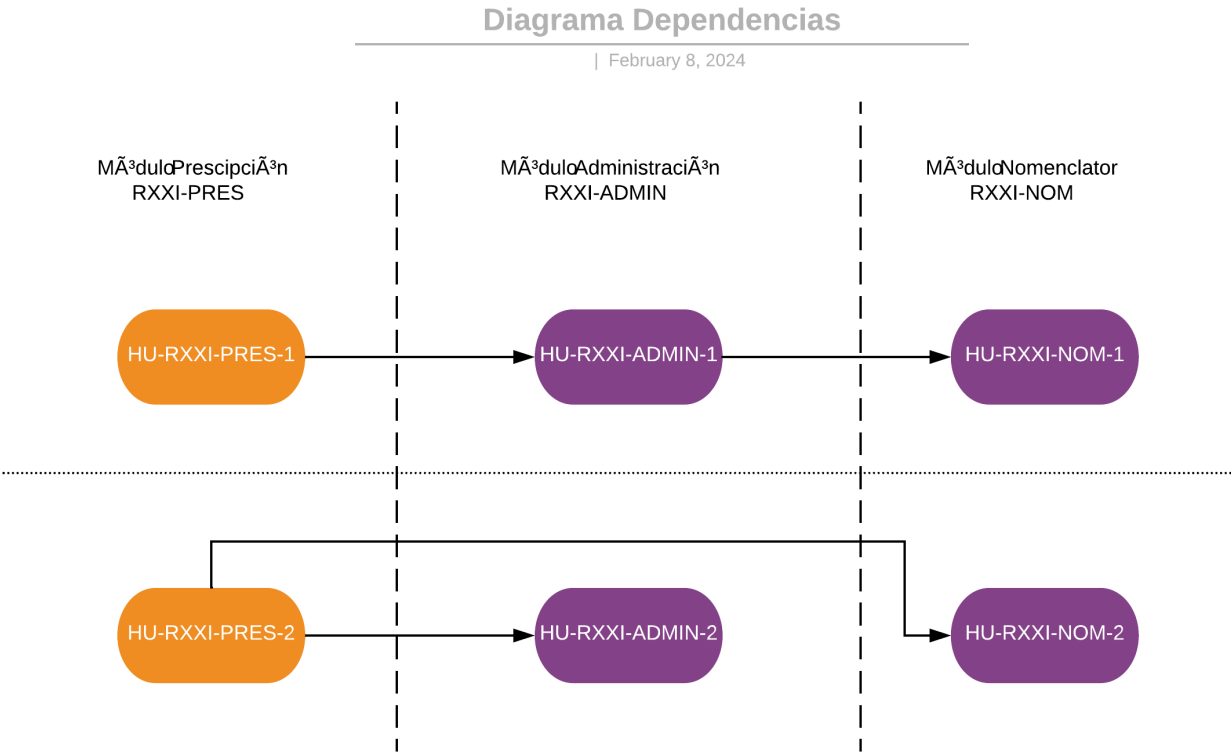


Diagram attachment access error: cannot display diagram

Diagrama Flujo de dependencias intra-módulos

| February 8, 2024

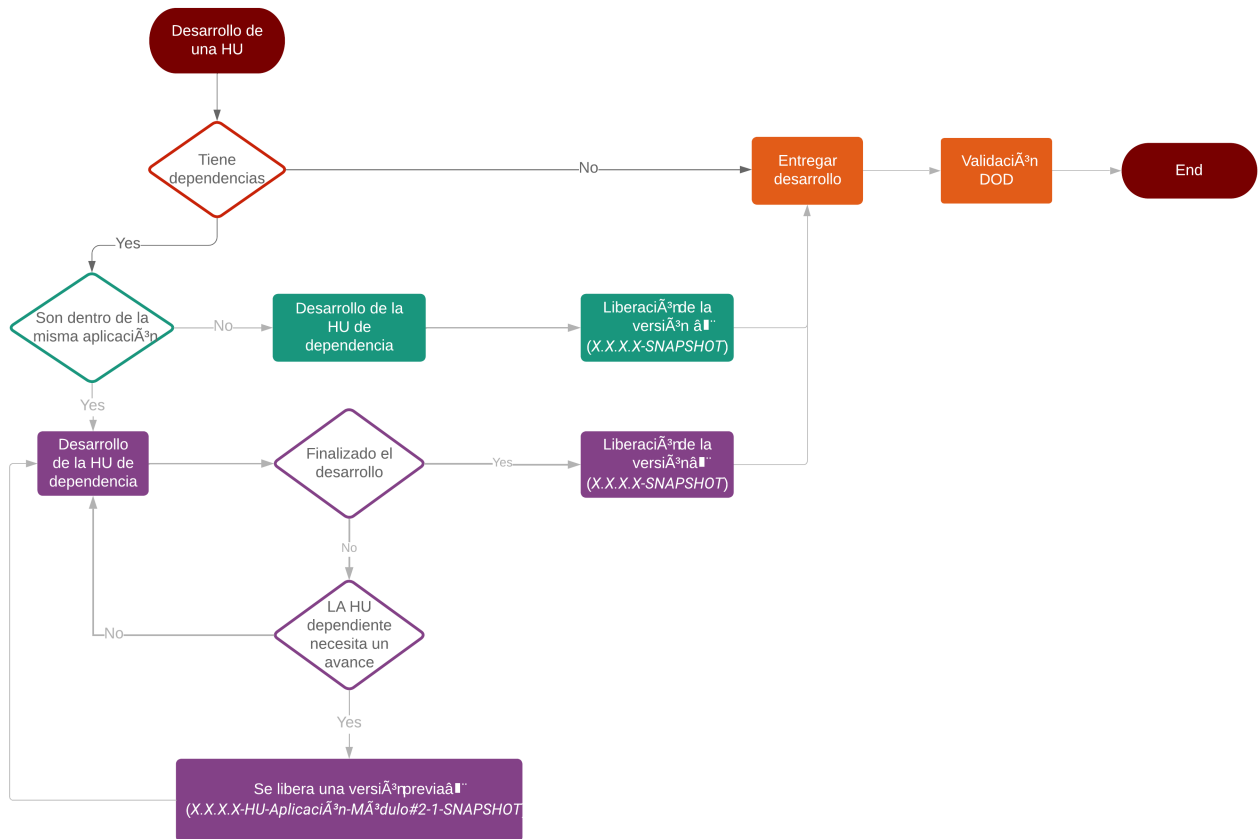


Diagram attachment access error: cannot display diagram

Gestión en Git de las HU's

Los pasos que se dan, asumiendo que la historia de usuario dependiente (**HU-Aplicación-Módulo#1-1**) ya está creada, son:

Se crea una HU (**HU-Aplicación-Módulo#2-1**) en el módulo del que se tiene dependencia que se encargará de resolver la dependencia.

Una vez se complete el desarrollo se generará una versión snapshot específica con los cambios de la HU (**HU-Aplicación-Módulo#2-1**). (**X.X.X.X-snapshot-HU-Aplicación-Módulo#2-1**)

La historia dependiente (**HU-Aplicación-Módulo#1-1**) apunta su dependencia con el módulo a esta versión temporal generada. (**X.X.X.X-snapshot-HU-Aplicación-Módulo#2-1**)

Una vez terminada la historia de usuario dependiente (**HU-Aplicación-Módulo#1-1**), se entrega (siempre después de haber completado la dependencia) y se sigue el flujo normal de entrega de una HU.

Nota

Para poder trabajar en paralelo, se aconseja que se definan los contratos de dependencia previos a los desarrollos (interfaces o funciones vacías) y que se usen mocks que contemplen los diferentes casos de uso de la HU. Con esto se pretende aumentar la eficacia de la integración de ambos módulos y mejorar la velocidad de desarrollo.

*La validación funcional (DoD), dentro de este caso de uso, siempre se hará sobre la HU dependiente (**HU-Aplicación-Módulo#1-1**).*

*El cierre las HU dependientes (**HU-Aplicación-Módulo#2-1**), queda sujeto a la aprobación de la HU dependiente (**HU-Aplicación-Módulo#1-1**).*

Actualmente el versionado se realiza de forma manual.

Diagrama Git

| February 8, 2024

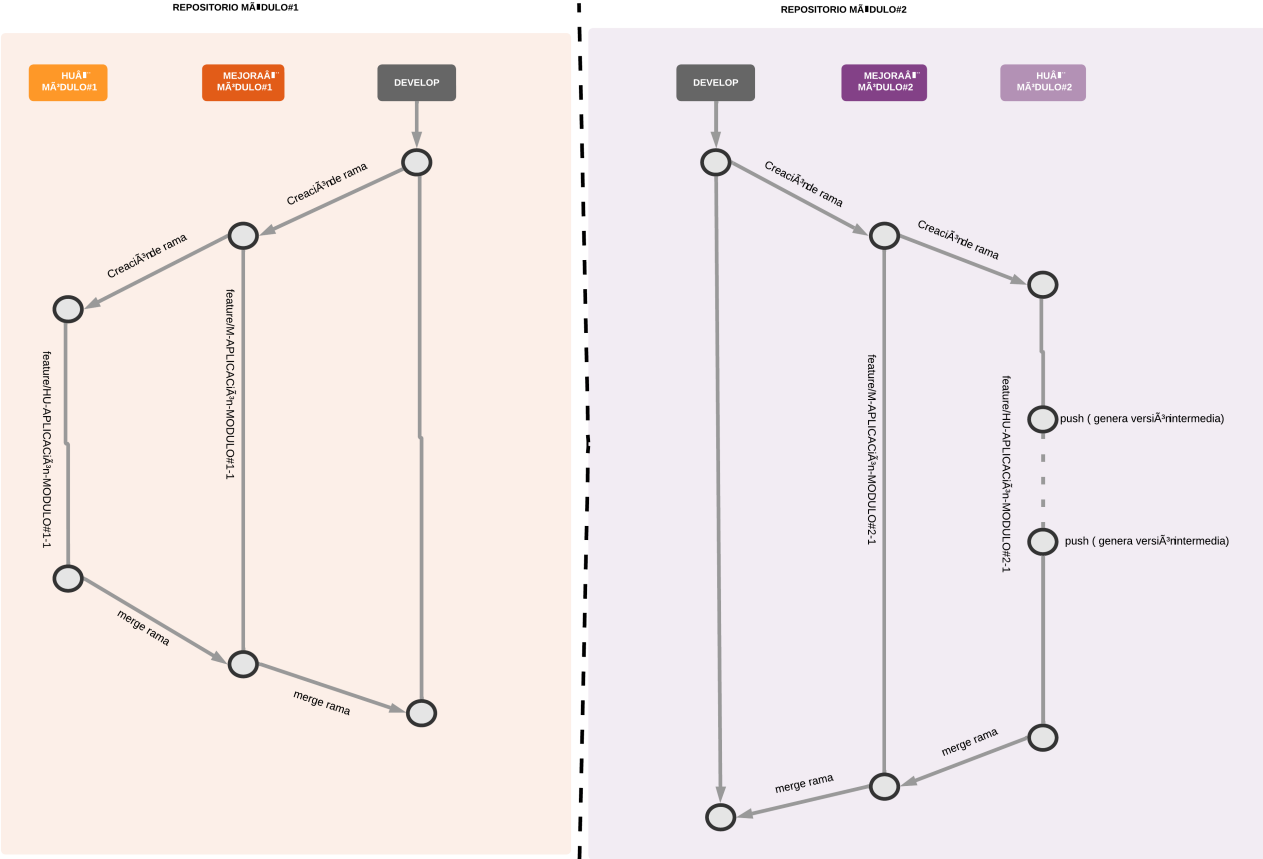


Diagram attachment access error: cannot display diagram