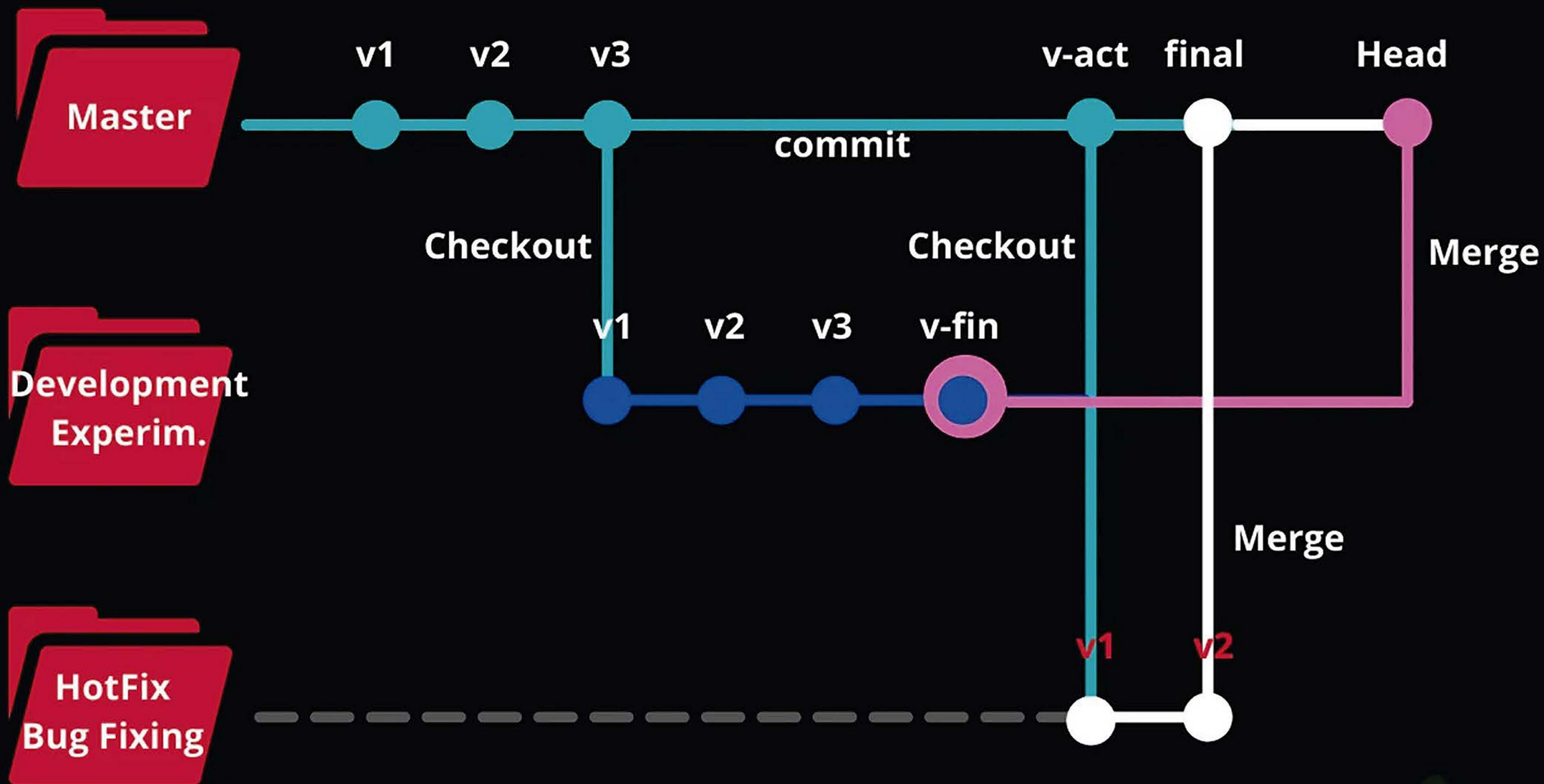


## ¿Qué es un Branch (rama) y cómo funciona un Merge en Git?





# ¿Qué es un Branch (rama) y cómo funciona un Merge en Git?

## ► Ideas Clave

Crear una nueva rama lo conocemos como **Checkout**.  
Unir dos ramas lo conocemos como **Merge**.



## Preguntas

### ¿Qué es Merge?

git merge rama, comando que permite unir el contenido de dos ramas en un solo commit.

### ¿Que es HotFix?

Rama de solución de errores apenas son detectados (en caliente).



## ► Notas de Clase

Todos los commits se aplican sobre una rama. Por defecto, siempre empezamos en la rama master (pero puedes cambiarle el nombre si no te gusta) y creamos nuevas ramas, a partir de esta, para crear flujos de trabajo independientes.

Crear una nueva rama se trata de copiar un commit (de cualquier rama), pasarlo a otro lado (a otra rama) y continuar el trabajo de una parte específica de nuestro proyecto sin afectar el flujo de trabajo principal (que continúa en la rama master o la rama principal).

### Estándar en equipos de desarrollo

**Rama master:** Todo va a producción.

**Rama "development":** Para las nuevas features, características y experimentos (para unirse a master cuando estén definitivamente listas)

**Rama "hotfix":** Para solucionar los issues o errores para luego unirse a master tan pronto como sea posible.

Solo ten en cuenta que combinar estas ramas (sí, hacer "merge") puede generar conflictos. Algunos archivos pueden ser diferentes en ambas ramas. Git es muy inteligente y puede intentar unir estos cambios automáticamente, pero no siempre funciona. En algunos casos, somos nosotros los que debemos resolver estos conflictos "a mano".



## Resumen

Podemos crear todas las ramas y commits que queramos. De hecho, podemos aprovechar el registro de cambios de Git para crear ramas, traer versiones viejas del código, arreglarlas y combinarlas de nuevo para mejorar el proyecto.







# git

Comandos basicos de Git



## ¿Que es un Branch y cómo funciona un Merge en Git?

### Ideas



#### Master:

Todo lo que esta en esta rama va a producción.



#### Development:

Las nuevas features, características y experimentos



#### HotFix:

Aqui van los errores se solucionan tan pronto como sea posible.



Si las ramas tienen algun conflicto para unirse git te avisara y te pedira que los corrijas, los conflictos pueden deberse a que se modificaron las mismas lineas del archivo en las dos ramas.



### Notas Clase



Cuando creas tu repositorio en tu carpeta de trabajo se crea por defecto una rama (**Branch**) principal llamada **master**.

**Branch** se puede ver como el mapa lineal de los **commits** que haz realizados al archivo.



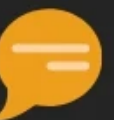
Usas **checkout** para crear una nueva rama desde el **commit** que desees de tu rama master, esta rama te sirve para hacer experimentos o reparar errores de tu código principal sin afectar al mismo.



Y para unir los cambios de esta rama de prueba con **master** utilizas **merge**, de esta manera las dos ramas se unirán formando una nueva rama.



### Resumen



**Branch** es aquella que representan los commits como un mapa lineal de tiempo, es posible crear nuevas ramas para realizar modificaciones de nuestro código sin afectar la rama principal.

**Merge** es el comando que se usa para unir dos ramas, generalmente los merge se hacen desde la rama master, se debe tomar en cuenta que puede haber conflicto entre las ramas.