

Alumno: ANTXÓN MOÇO TORRALBA
Profesor: JUAN LLADÓ MARTORELL
Curso: 1 DAM
Instituto: IES Son Ferrer

ENTORNOS DE DESARROLLO

Actividad 3. Control de versiones

ENUNCIADO:

Entregar un documento en el que se investigue:

- Control de versiones
- Versionado semántico

Puedes consultar los apuntes de la asignatura como punto de partida.

RESPUESTA:

¿QUÉ ES EL CONTROL DE VERSIONES?

Es una herramienta que permite **gestionar los cambios** realizados en el **código fuente** y otros archivos a lo largo del tiempo.

Es como una herramienta que va haciendo fotos de forma regular cada cierto tiempo, sobre el estado del código.



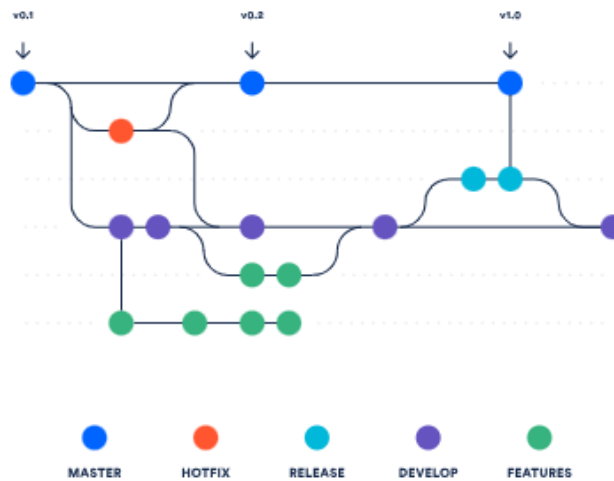
Funcionalidades principales de los sistemas de control de versiones:

- **Historial de cambios:** Registra todas las modificaciones realizadas en los archivos a lo largo del tiempo.
- **Gestión de ramas:** Permite crear ramas o líneas de desarrollo paralelas.
- **Fusiones:** Facilita la combinación de cambios realizados en diferentes ramas.
- **Identificación de versiones:** Permite etiquetar puntos específicos en la historia del código para identificar versiones importantes o lanzamientos.
- **Detección y resolución de conflictos:** Cuando dos o más personas realizan cambios en el mismo archivo o línea de código.
- **Colaboración:** Facilita el trabajo en equipo al permitir que varios desarrolladores trabajen en el mismo proyecto de manera asistente.
- **Seguridad y respaldo:** Proporciona una capa de seguridad.
- **Seguimiento de problemas:** Algunas plataformas de control de versiones están integradas con sistemas de seguimiento de problemas o proyectos.
- **Revisión de código:** Algunos sistemas de control de versiones ofrecen herramientas para realizar revisiones de código colaborativas.

Pueden haber diferentes tipos de control de versiones:

- Sistemas de Control de Versiones Locales
- Sistemas de Control de Versiones Centralizados
- Sistemas de Control de Versiones Distribuidos

Cómo funciona un control de versiones:



El diagrama de control de versiones es una representación gráfica que muestra la evolución de un proyecto a lo largo del tiempo. Registra las diferentes versiones o cambios realizados en los archivos o código fuente. Funciona de manera visual con los siguientes puntos:

- Diagrama de ramificación: Visualiza como las ramas se separan, se desarrollan y se fusionan
- Commit: Los cambios se agrupan en commits
- Ramas(Branches): Los proyectos pueden tener diferentes líneas llamadas ramas. Una rama principal (main) representa la versión estable del proyecto. Y pueden haber ramas adicionales para desarrollar nuevas características o arreglar errores.
- Historial y seguimiento: Registra quien y cuando se hizo el cambio
- Resolución de conflictos: Ayuda a manejar conflictos

Ejemplos de algunos software de control de versiones:

- Git
- CVS
- Apache Subversion (SVN)
- Mercurial
- Monotone

Algunos comandos principales que podemos usar (en el software git):

- git config (configura variables)
- git init (inicia un repositorio)
- git add (agrega archivos al área de preparación)
- git commit (guarda los cambios en el repositorio)
- git status (muestra el estado de los archivos)
- etc...

¿QUÉ ES EL VERSIONADO SEMÁNTICO?

Se conoce también como **SemVer**, es un sistema de numeración de versiones para software que busca **comunicar cambios** en el software de manera clara. Este sistema utiliza un formato de **tres números separados**(X.Y.Z):



- **X (Mayor):** Aumenta cuando se realizan cambios **incompatibles** con versiones anteriores.
- **Y (Menor):** Aumenta cuando se agregan nuevas funcionalidades de una manera **compatible** con versiones anteriores.
- **Z (Parche):** Aumenta cuando se realizan **correcciones de errores compatibles** con versiones anteriores

SemVer es tan importante en el desarrollo de software porque al aplicarla a cualquier desarrollo mostrará el estado de un software de manera rápida a las personas que contribuyen en él, y al mismo tiempo es posible crear o encontrar documentación.



Hay varias herramientas que nos pueden ayudar en facilitar la publicación de versiones en nuestro código:

- Release It!: Automatización de versionamiento y publicaciones
- Release Please: Automatización de archivos changelog, versionamiento y publicación
- Sui: Paquete de gestión de commits y releases en proyectos monorepo