

Universidad Don Bosco



Asignatura: Desarrollo de Aplic. Web con Soft. Interpret. en el Cliente

Ciclo / Grupo: 03-2022 / G03T

### **Investigación aplicada 1**

Integrantes:

Rodrigo Isaac Vásquez Vásquez	VV221013
Billy Alexis Valdez Hernández	VH221907
Christian José Peña Laínez	PL211289
Levi Isaac Valle Lovato	VL220247
Edgardo Giovanni Cortez Valladares	VR221500

# **Sistemas de control de versiones y sus ventajas**

## Los sistemas de control de versiones

Un sistema de control de versiones es una herramienta utilizada en el desarrollo de software para evitar el riesgo de conflictos que puedan surgir al trabajar en colaboración con otros equipos de desarrollo. A medida que los entornos de desarrollo se aceleran, los sistemas de control de versiones ayudan a los equipos de software a trabajar de forma más rápida e inteligente.

El software de control de versiones realiza un seguimiento de todas las modificaciones en el código en un tipo especial de base de datos. Si se comete un error, los desarrolladores pueden ir hacia atrás en el tiempo y comparar las versiones anteriores del código para ayudar a resolver el error, al tiempo que se minimizan las interrupciones para todos los miembros del equipo. Cada versión de software tiene un nombre. Por lo general, estos nombres se asignan numéricamente y existen varios sistemas estándar para la progresión de las versiones de software.

El control de versiones de software es importante porque ayuda tanto a los usuarios como a los proveedores de software a rastrear las diferentes versiones que lanza la empresa. Los usuarios confían en los desarrolladores de software para mantenerse actualizados y esperan una forma metódica de comprender cuándo y qué actualizaciones se publican. Para los proveedores de software, el control de versiones es aún más crítico. Realizan un seguimiento de las versiones de software y utilizan la información que recopilan para impulsar sus planes de segmentación y monetización. Al separar las actualizaciones esenciales y secundarias, los proveedores pueden destacar, segmentar y cobrar por ciertas funciones.

El software de control de versiones es una parte esencial del día a día de las prácticas profesionales del equipo de software moderno. Los desarrolladores de software.

*(Thales.org 3 de abril de 2022)*

## **Ventajas**

### **Edición compartida:**

Permite que varias personas trabajen simultáneamente en un solo proyecto. Cada persona edita su propia copia de los archivos y elige cuándo compartir esos cambios con el resto del equipo. Por lo tanto, las ediciones temporales o parciales de una persona no interfieren con el trabajo de otra.

### **Creación de ramas y fusiones:**

Si se tiene a miembros del equipo trabajando al mismo tiempo, es algo evidente; pero incluso las personas que trabajan solas pueden beneficiarse de la capacidad de trabajar en flujos independientes de cambios. La creación de una "rama" en las herramientas de VCS mantiene múltiples flujos de trabajo independientes los unos de los otros al tiempo que ofrece la facilidad de volver a fusionar ese trabajo, lo que permite que los desarrolladores verifiquen que los cambios de cada rama no entran en conflicto.

*(Atlassian.org 2013)*

### **Automatización de tareas:**

Las características de automatización del control de versiones ahorran tiempo y generan resultados coherentes. Automatizar las pruebas, el análisis de código y la implementación cuando se guardan nuevas versiones en el control de versiones.

### **Un completo historial de cambios a largo plazo de todos los archivos:**

Esto quiere decir todos los cambios realizados por muchas personas a lo largo de los años. Los cambios incluyen la creación y la eliminación de los archivos, así como los

cambios de sus contenidos. Las diferentes herramientas de VCS difieren en lo bien que gestionan el cambio de nombre y el movimiento de los archivos.

*(Atlassian.org 2013)*

### **Creación de flujos de trabajo:**

Los flujos de trabajo de control de versiones impiden el caos de todos los usuarios que usan su propio proceso de desarrollo con herramientas diferentes e incompatibles. Los sistemas de control de versiones proporcionan permisos y cumplimiento de procesos para que todos permanezcan en la misma página.

### **Actúa como backup:**

Una funcionalidad adicional de agradecer cuando utilizamos un VCS distribuido como Git es que puede actuar como backup. Cada miembro del equipo tiene una versión completa del proyecto en su disco, incluyendo la historia completa del proyecto. Si se da el caso de que el cosmos ha conspirado hoy en tu contra y tu servidor se va a freír espárragos (y sus unidades de respaldo fallan), todo lo que necesitas para la recuperación es uno de los repositorios Git locales de tus compañeros de equipo.

### **Trazabilidad:**

Ser capaz de trazar cada cambio que se hace en el software y conectarlo con un software de gestión de proyectos y seguimiento de errores como Jira, además de ser capaz de anotar cada cambio con un mensaje que describa el propósito y el objetivo del cambio, no solo te ayuda con el análisis de la causa raíz y la recopilación de información.

*(Atlassian.org 2013)*

# **Comandos más comunes de Git/GitHub**

# Comandos de Git/Github

## Git commit

Ya los archivos preparados en el área de preparación, para confirmar dichos archivos y crear una confirmación de cambios la sentencia utilizada es git commit.

```
git commit [opciones] [<Archivo_1>] [<Archivo_2>]
[<Archivo_N>] [-m<" Mensaje de confirmación" >]
```

## Git log

Cada vez que confirmamos cambios (hacemos un git commit), se crea una confirmación con un número hash identificativo. El comando para ver todas las confirmaciones realizadas en nuestro repositorio es Git log.

git log

## Git reset

Puede descartar confirmaciones que ya no necesita usar utilizando el comando git reset.

```
Git reset hash_commit_al_que_volvemos
```

## Git revert

Puede utilizar dicho comando para deshacer de forma segura una confirmación que ya se haya enviado.

```
Git revert hash_commit
```

## Git tag

Una etiqueta Git (tag) se utiliza para etiquetar y marcar una confirmación específica en el historial.

```
git tag [opciones] [<Nombre_etiqueta>]
[<Hash_commits>] [<Mensaje_etiqueta>]
```

## Git remote

Para conectar al repositorio remoto con el cual nos comunicaremos entre nuestro repositorio local y dicho repositorio remoto el comando utilizado es git remote.

```
git remote add <Nombre repositorio> <URL remoto>
git remote show <Nombre repositorio>
```

## Git checkout

Para moverse entre las ramas creadas y realizar los cambios necesarios sobre dichas ramas, el comando utilizado es git checkout.

```
$ git checkout Nombre_rama
```

## Git config

El comando git config sirve para definir valores de configuración de Git a nivel de un proyecto global o local.

```
git config --global user.name <Nombre de usuario>
```

```
git config --global user.name <dirección email>
```

## Git init

Este comando inicializa un nuevo repositorio en el repositorio local.

```
git init <nombre_repositorio>
```

## Git clone

Este comando inicializa un nuevo repositorio en el repositorio local clonando íntegramente el contenido de un repositorio remoto que le indiquemos mediante una URL.

```
git clone <URL_repositorio>
```

## Git add

Realiza los cambios en nuestra área de trabajo, para comenzar la confirmación es necesario pasar todos los archivos que queramos confirmar al área de preparación.

```
git add [opciones] [<Archivo_1>]
[<Archivo_2>] ... [<Archivo_N>]
```

## Git diff

Este comando permite ver las diferencias que existen entre las confirmaciones que determinemos. Se hace referenciando las confirmaciones mediante su número hash.

```
git diff <hash commit 1> <hash commit 2>
```

## Git Branch

Las ramas es uno de los conceptos más interesantes dentro de GIT. Permite el trabajo en paralelo sobre distintas partes del código base para añadir funcionalidades, corregir errores o realizar versiones de nuestros proyectos. Para listar una rama, el comando utilizado es git branch.

```
Git branch [opciones] [<Nombre_rama>]
[<Commit de comienzo>]
```

## Git status

Podemos comprobar en cualquier momento el estado de la rama en la que nos encontramos para comprobar si existen archivos (o directorios) que tienen cambios que deben de ser confirmados o rechazados.

git status

## Git merge

Si se desea conservar todos los cambios e historiales de la rama combinada, el comando a utilizar sería git merge que realiza una fusión entre las ramas seleccionadas.

```
git merge <Nombre_rama> [-no-commit]
```

## Git rebase

Este comando reorganiza las ramas y crea una única revisión de dicha combinación de ramas.

```
git rebase <Nombre_ramas>
```

## Git push

Para enviar todos los cambios que tenemos en nuestro repositorio local al repositorio remoto, el comando a utilizar sería git push.

```
git push <Nombre repositorio> <Nombre_rama>
```

## Git pull

Para extraer todos los cambios del repositorio remoto en nuestro repositorio local y poder tener así la última confirmación de dicho repositorio, el comando a utilizar es git pull.

```
git pull <Nombre_repositorio / Nombre_rama>
```

Fuente: (INESEM.org 8 de abril de 2022)

# **Instalación para el sistema operativo Windows**



# INSTALACIÓN DEL SISTEMA OPERATIVO DE WINDOWS

DAW404 G03T

## 1. La instalación desde DVD:

Este método permite instalar y reparar el sistema operativo Windows 10 tras insertar el medio de instalación. El usuario deberá previamente configurar la BIOS (Basic Input Output System) para que el ordenador arranque por defecto desde el lector de DVD.



## 2. La instalación desde una llave USB:

A partir de Windows 98, las versiones de los sistemas operativos de Microsoft se comercializaban en soportes ópticos (CD o DVD). Si posee un ordenador portátil que no cuenta con un lector óptico, puede utilizar un soporte muy extendido, la llave USB, que contenga los archivos de instalación de Windows 10.



## 3. El uso de un disco virtual con arranque nativo:

El arranque nativo permite a un disco virtual contener Windows 10 Enterprise y ejecutarlo en un ordenador físico sin un hipervisor como Hyper-V o un producto de terceros.



## 4. El uso de Windows To Go:

Permite a los usuarios arrancar el sistema operativo a partir de un lector USB externo.

(Atlassian.org 2013)



## 5. El despliegue de una imagen a través de la red:

Con la ayuda de los servicios de despliegue.

