



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:

10/08/2023

Hora de inicio:

09:00

Introducción a la Programación y Computación 1 [Sección B]

Diego Fernando Cortez Lopez

Agenda

- Lectura de Enunciado
- Clase 3
- Dudas
- Ejemplo Práctico

Versionamiento

Clase 3

Versionamiento

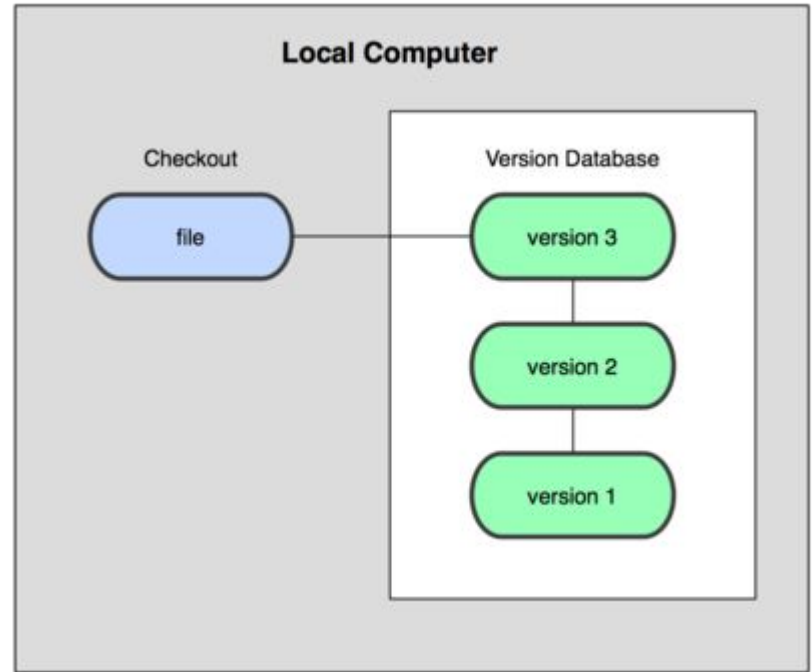
Se refiere al proceso de controlar y gestionar las diferentes versiones de un software a lo largo de su ciclo de vida. Es una práctica fundamental para el desarrollo de software colaborativo y para mantener un historial de cambios y mejoras en el código fuente.

El versionamiento permite a los desarrolladores a trabajar en paralelo, experimentar con nuevas características y corregir errores sin afectar la estabilidad del proyecto en su conjunto.

Hay tres tipos de modelo de control de versiones: Modelo local, modelo centralizado y modelo distribuido

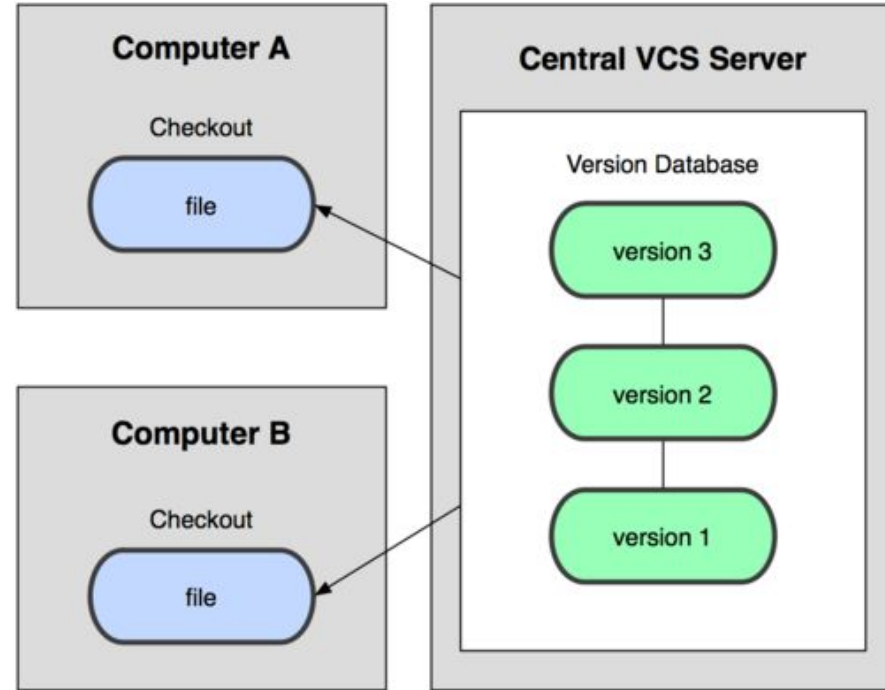
Modelo Local

- El modelo local utiliza una copia de la base de control de versiones y una copia de los archivos del proyecto.
- Es el más sencillo y no es recomendable cuando se trabaja en equipo ya que tienen que acceder a los mismos archivos.
- Los cambios son guardados localmente y no se comparten con nadie.



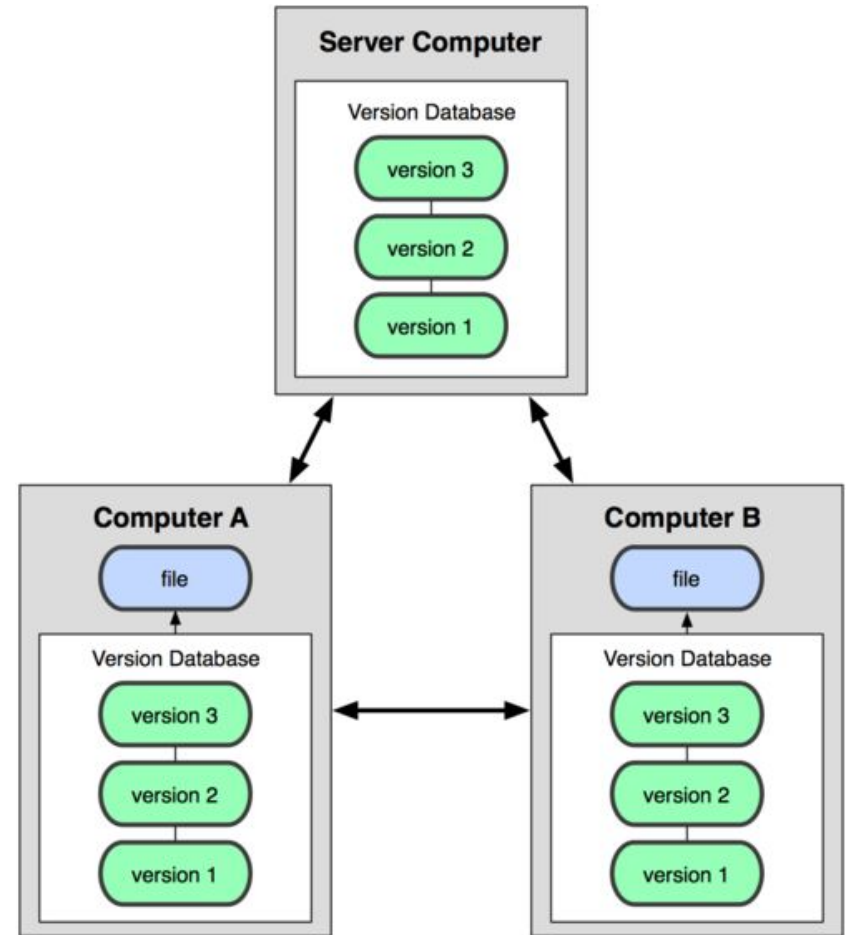
Modelo Centralizado

- Existe un repositorio centralizado de todo el código, del cual es responsable un único usuario.
- Se facilitan las tareas administrativas a cambio de reducir flexibilidad, ya que todas las decisiones, como crear una nueva rama, necesitan la aprobación del responsable.



Modelo Distribuido

- Cada usuario tiene su propio repositorio.
- Los distintos repositorios pueden intercambiar y mezclar revisiones entre ellos.
- Es frecuente el uso de un repositorio que sirve de punto de sincronización de los distintos repositorios locales.



Herramientas de control de versiones

Git

Es un sistema de control de versiones de software, diseñado para ser rápido y manejar grandes proyectos con muchos archivos de manera eficiente.

Git permite rastrear cambios realizados en el código fuente y mantener un historial de versiones, lo que facilita el trabajo en equipo y la resolución de conflictos.



GitHub

Es una plataforma en línea que ofrece alojamiento y control de versiones para proyectos de software utilizando Git.

Proporciona herramientas para colaboración, documentación y equipos de software para trabajar en productos juntos.



GitLab

Es una plataforma en línea que ofrece funciones similares a GitHub para control de versiones y colaboración en proyecto de software.

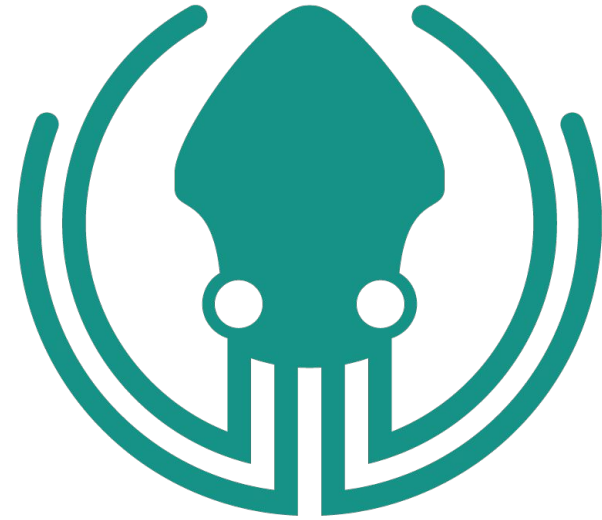
Es una solución de código abierto que se puede instalar en los servidores de la empresa.



Git Kraken

Es una plataforma que ofrece un conjunto de herramientas para ayudar a los desarrolladores en la productividad.

Permite a los equipos de trabajo puedan colaborar de forma más fácil sin importar el lugar donde estén.



GitKraken

Terminología

- **Repositorio (repository):** Es el lugar en el que se almacenan los datos actualizados e históricos de cambios.
- **Revisión (revisión):** Es una versión determinada de información que se gestiona.
- **Etiqueta (tag):** Permiten identificar de forma fácil revisiones importantes en el proyecto. Se suelen usar para identificar el contenido de las versiones publicadas del proyecto.
- **Rama (branch):** Un conjunto de archivos pueden ser ramificado o bifurcado en un punto en el tiempo de manera que, a partir de ese momento, dos copias de esos archivos se pueden desarrollar de formas diferentes independiente el uno del otro.

Terminología

- **Cambio (change):** Representa una modificación específica de un documento bajo el control de versiones.
- **Desplegar (checkout):** Es crear una copia de trabajo local desde el repositorio. Se puede especificar una revisión en concreto u obtener la última.
- **Confirmar (commit):** Es escribir los cambios realizados en la copia de trabajo del repositorio.
- **Conflicto (conflict):** Se produce cuando diferentes partes realizan cambios en el mismo documento, y el sistema es incapaz de conciliar los cambios. Se debe resolver mediante la integración de los cambios o mediante la selección de un cambio en favor del otro.

Terminología

- **Cabeza (head):** Se refiere a la última confirmación, ya sea en el trunk o en una rama.
- **Tronco (trunk):** Es la única línea de desarrollo que no es una rama a veces llamada **mastre**.
- **Mezclar (merge):** Es una operación en la que se aplican dos tipos de cambios en un archivo o conjunto de archivos. Algunos escenarios serían:
 - Un usuario actualiza su copia de trabajo con los cambios confirmados por otros usuarios en el repositorio.
 - Se crea una rama con archivos independientes y la rama actualizada se incorpora más tarde en la rama master.

Tareas Básicas

Creación de Repositorio

Para crear un repositorio se usa el comando `git init`

```
$ git init  
Initialized empty Git repository in /home/cc0gobas/git/curso-de-git/.git/
```

Añadir la aplicación

Para colocar tus archivos en el directorio del repositorio se usa el comando `git add` para agregar los archivos al area de preparacion para el commit. Se puede utilizar el comando para agregar un archivo en especifico `git add HolaMundo.js` o para agregar todos los archivos `git add .`.

```
$ git add .  
$ git commit -m "Creación del proyecto"  
[master (root-commit) e19f2c1] Creación del proyecto  
1 file changed, 2 insertions(+)  
create mode 100644
```

Comprobar el estado del repositorio

Con el comando `git status` podemos ver en qué estado se encuentran los archivos de nuestro repositorio

```
$ git status
# On branch master
nothing to commit (working directory clean)
```

Creación de ramas

Cuando se trabaja en una nueva funcionalidad, es conveniente hacerlo en una nueva rama, para no modificar la rama principal y dejarla inestable. Aunque el comando para manejar ramas es `git branch` también `git checkout`.

```
$ git branch hola
```

```
$ git checkout hola  
Switched to branch 'hola'
```

Nota: Si usamos `git branch` sin ningún argumento, nos devolverá la lista de ramas disponibles.

Clonar un repositorio

Clonar un repositorio significa crear una copia local de un repositorio remoto en tu máquina. Es útil cuando se desea trabajar en código de un proyecto existente, colaborar con otros desarrolladores o contribuir a proyectos de código abierto. Para clonar un repositorio se utiliza el comando `git clone url`.

```
git clone <URL_del_repositorio_remoto>
```

Push y Pull

Estas operaciones permiten sincronizar los cambios entre un repositorio local y un repositorio remoto, lo que facilita la colaboración entre desarrolladores.

Push

Se utiliza para enviar los cambios realizados en tu repositorio local a un repositorio remoto.

```
git push <nombre_remoto> <nombre_rama_local>
```

Si se desea enviar los cambios desde la rama local “main” al repositorio remoto “origin”

```
git push origin main
```

Pull

Se utiliza para traer los cambios realizados en el repositorio remoto y fusionarlos en el repositorio local.

```
git pull <nombre_remoto> <nombre_rama_remota>
```

Si se desea obtener los cambios de la rama remota "main" al repositorio local

```
git pull origin main
```


Comandos Git

- Clonar repositorio: `git clone <url del repositorio>`

- Subir cambios al repositorio:

- a. Inicializar un repositorio en el directorio local:

- `git init`

- b. Agregar los archivos modificados:

- `git add .`

- c. Realizar un "commit" de los cambios:

- `git commit -m "Mensaje de commit"`

- d. Conectar a un repositorio remoto:

- `git remote add origin <url del repositorio>`

- e. Sube los cambios al repositorio remoto:

- `git push -u origin <nombre de la rama>`

- Descargar cambios:

- a. `cd <nombre del directorio>`

- b. `git pull origin <nombre de la rama>`

Tarea 1

Investigar y crear una tabla comparativa que muestre las diferencias entre tipos de datos primitivos y no primitivos en un lenguaje de programación a su elección. Incluir ejemplos de cada tipo de dato.

Investigar que es el casteo implícito y explícito, y dar ejemplos de cada uno con un lenguaje de programación a su elección.

- Entrega en formato pdf
- Ejemplos realizados por usted no computadora
- Identificar el documento con su nombre y carne
- Fecha de entrega: 16/08/2023

¿Dudas?