**GUIA VERSIONAMIENTO**

**NOMBRE APRENDIZ**

Brayan David Casallas Candela

**NOMBRE INSTRUCTOR**

Javier Leonardo Pineda Uribe

|  |
| --- |

**INSTITUCIÓN**

Centro De Servicios Financieros

**1. ¿**Qué es un repositorio y mención 2 ejemplos de los mismos (proveedores online)**?**

**RTA:** Un repositorio es como un tipo de biblioteca digital donde se almacenan, gestionan y difunden colecciones de recursos digitales, como documentos, archivos, imágenes, videos, datos, o cualquier otro tipo de contenido digital. Estos repositorios están diseñados para organizar la información de manera eficiente y facilitar el acceso, la búsqueda y la recuperación de los recursos que contienen.

**Ejemplos: 1-ArXiv:** Es un repositorio de pre publicaciones científicas de acceso abierto, principalmente en campos como física, matemáticas y ciencias de la computación. Los investigadores pueden compartir sus trabajos antes de que sean publicados en revistas tradicionales, lo que agiliza la difusión de nuevas ideas y fomenta la colaboración a nivel mundial.

**2-PubMed Central (PMC):** Este repositorio digital gratuito almacena artículos académicos de texto completo en las áreas de biomedicina y ciencias de la vida. PMC facilita el acceso a una vasta cantidad de literatura científica, lo que es fundamental para la investigación en el campo de la salud.

**2. ¿**Cuáles son los principales componentes de un versionamiento en la herramienta GIT**?**

**RTA: Repositorio:** Es el espacio donde se almacenan todos los archivos y el historial de cambios. Puede estar de manera local en tu máquina o en un servidor remoto.

**Commit:** Representa un "snapshot" o instantánea de los cambios realizados en el proyecto en un momento dado. Cada commit incluye un mensaje descriptivo que indica qué cambios se hicieron.

**Branch (Rama):** Una rama es una línea de desarrollo independiente. La rama principal por defecto es normalmente main (o master), pero puedes crear otras ramas para trabajar en nuevas características o correcciones sin afectar la rama principal.

**Stage (Index o Área de preparación):** Antes de hacer un commit, los archivos modificados deben ser "stageados". Esto implica que los cambios están listos para ser confirmados, pero aún no han sido añadidos al historial del repositorio.

**Merge:** El proceso de combinar los cambios de una rama en otra, por ejemplo, incorporar los cambios de una rama secundaria en la rama principal (main). Esto puede requerir resolver conflictos si ambos han modificado las mismas partes del código.

**Pull:** Actualiza el repositorio local con los cambios más recientes del repositorio remoto. Combina dos acciones: fetch (descargar los cambios) y merge (integrar los cambios).

**Push:** Envia los cambios locales (commits) al repositorio remoto, haciendo que estén disponibles para otros usuarios.

**Clone:** Copia un repositorio remoto a tu máquina local, creando una versión local del repositorio con todo su historial de cambios.

**Checkout:** Cambia el estado de los archivos en tu directorio de trabajo al de un commit o rama específicos. Esto te permite moverte entre diferentes puntos de desarrollo del proyecto.

**Fetch:** Descarga los cambios desde el repositorio remoto, pero sin aplicarlos automáticamente al repositorio local.

**Tag:** Los tags son marcadores específicos que se colocan sobre un commit en particular, normalmente usados para identificar versiones importantes, como lanzamientos de software (v1.0, v2.0).

**3. ¿**Mencione con sus propias palabras las ventajas que tiene GIT frente a otros proveedores de

repositorios**?**

**RTA:** GIT tiene muchas ventajas las cuales son la flexibilidad y la autonomía ya que con esta herramienta podemos trabajar donde queramos cuando queramos y como queramos depende de las necesidades de cada usuario.

Otra de sus ventajas es su velocidad y eficiencia ya que desarrolla mas rapido y sin errores.

Aparte cuenta con ramas y facilita la creación y manejo de estas lo que permite trabajar en nuevas características o correcciones sin afectar el código principal.

Y por ultimo una de sus mas importantes ventajas es el control detallado sobre los cambios y permite ver el historial de versiones de manera detallada, facilitando la auditoría y el seguimiento del trabajo.

**4.** Mencione por lo menos 5 ejemplos de los comandos básicos que se usan en GIT.

**RTA: -git init:** Inicializa un nuevo repositorio de Git en el directorio actual.

Ejemplo: git init

**-git clone:** Clona un repositorio existente (desde una URL) en tu máquina local.

Ejemplo: git clone https://github.com/usuario/repo.git

**-git add:** Agrega cambios al área de preparación (staging area) para ser confirmados.

Ejemplo: git add archivo.txt

**-git commit:** Confirma los cambios añadidos al área de preparación, creando un nuevo registro en el historial del repositorio.

Ejemplo: git commit -m "Mensaje de confirmación"

**-git push:** Envía los commits locales al repositorio remoto, actualizando el código en el servidor.

Ejemplo: git push origin main

**5. ¿**Que son y cuáles son las funciones de los branch**?**

**RTA:** En programación, un "branch" (rama) es una bifurcación o una copia separada del código fuente de un proyecto, que permite desarrollar nuevas funcionalidades o corregir errores sin afectar la versión principal (usualmente llamada "master" o "main"). Esta técnica es común en sistemas de control de versiones distribuidos como Git.

**Funciones de los branchs**

1. **Desarrollo paralelo:** Permiten que múltiples desarrolladores trabajen en diferentes características o soluciones al mismo tiempo, sin interferir con el código en la rama principal.
2. **Aislar cambios:** Se puede trabajar en una característica o una corrección de errores de forma independiente. Esto asegura que cualquier error o problema no afecte el código principal hasta que esté listo para integrarse.
3. **Pruebas y revisión:** Las ramas se utilizan para probar nuevas funcionalidades y someter los cambios a revisión. Esto facilita la colaboración, ya que otros desarrolladores pueden revisar el código antes de integrarlo en la rama principal.
4. **Integración continua:** En muchas metodologías ágiles, como *GitFlow*, se crean ramas específicas para cada etapa del desarrollo, facilitando la integración gradual de cambios al sistema de producción.
5. **Seguridad del código:** Al trabajar en ramas, el código en producción o en el repositorio principal se mantiene seguro y libre de errores hasta que los cambios sean verificados y aprobados.

**6. ¿**Cuál es el Nombre del branch principal**?**

**RTA:** El nombre del branch principal en un sistema de control de versiones, especialmente en Git, solía ser comúnmente "master". Sin embargo, debido a consideraciones sobre el lenguaje y la inclusividad, muchas comunidades han comenzado a adoptar el término "main" como el nombre para la rama principal.

Por lo tanto, el nombre del branch principal puede ser:

* "master" (en versiones más antiguas de Git)
* "main" (en versiones más recientes y en muchas nuevas prácticas de desarrollo)