GUIA #1 - VERSIONAMIENTO

APRENDIZ: DIEGO FELIPE MARTINEZ

INGENIERÍA DE SOFTWARE 1

FICHA: 3065835-2



1. ¿Qué es un repositorio y mencione 2 ejemplos de estos (proveedores online)?

Es un espacio centralizado donde se almacena, organiza, mantiene y difunde información digital, habitualmente archivos informáticos, que pueden contener trabajos científicos, conjuntos de datos o software. Los repositorios tienen sus inicios en los años 90, en el área de la física y las matemáticas, donde los académicos aprovecharon la red para compartir sus investigaciones con otros colegas. Este proceso era valioso porque acelera el ciclo científico de publicación y revisión de resultados.​ Son sistemas de información que preservan y organizan materiales científicos y académicos como apoyo a la investigación y el aprendizaje, a la vez que garantizan el acceso a la información.​

Ejemplos:

1. Databit: repositorio de datos de investigación desarrollado con el apoyo del Instituto de Servicios de Museos y Bibliotecas.
2. Odisea: es un inventario internacional de los depósitos que admiten conjuntos de datos de investigación a escala mundial. Permite buscar y sugerir nuevos bancos de datos.

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

En Git, el versionamiento se basa en los siguientes componentes principales:

1. Repositorio (Repo): Es el lugar donde se almacena todo el historial de cambios del proyecto.

2. Commits: Son instantáneas del estado del proyecto en un momento determinado. Cada commit tiene un mensaje descriptivo.

3. Ramas (Branches): Permiten trabajar en diferentes versiones del proyecto de manera simultánea.

- Rama principal (Master): Es la rama principal del proyecto.

- Rama de desarrollo (Develop): Es la rama donde se realizan los cambios.

4. Merge: Es el proceso de combinar cambios de una rama a otra.

5. Tags: Son referencias a commits específicos, utilizados para marcar versiones importantes.

6. Historial (Log): Es el registro de todos los commits realizados.

7. Etiquetas (Labels): Son categorías para commits y issues.

8. Issues: Son problemas o tareas pendientes.

9. Pull Request (PR): Es una solicitud para revisar y mergear cambios.

10. Fork: Es una copia del repositorio original.

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

Una de las grandes ventajas que tiene GIT frente a otros repositorios es la flexibilidad ya que cuenta con muchas más herramientas que otros repositorios, un ejemplo seria que GIT cuenta con Git Hub que es la herramienta que nos ayuda con los proyectos ya sean individuales o grupales.

Otra ventaja es la seguridad ya que GIT nos permite clonar un repositorio y así mismo se obtiene una copia lo cual da un respaldo al repositorio

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

- git init: Inicializa un repositorio.

- git branch: Crea una nueva rama.

- git commit: Crea un commit.

- git push: Envía cambios al repositorio remoto.

- git pull: Descarga cambios del repositorio remoto.

- git merge: Combina cambios.

- git add: Agrega cambios al área de staging.

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

El branch es la forma de hacer cambios en nuestro proyecto sin afectar el flujo de trabajo de la rama principal, esto porque queremos trabajar una parte muy específica de la aplicación.

Una de las funciones es que se pueden trabajar nuevas tareas, pero sin hacer cambios en la parte principal, en pocas palabras podemos experimentar sin dañar la parte principal o la base del trabajo

También se puede corregir el trabajo principal pero en una rama diferente y después de ser corregida esta se puede fusionar en la parte principal del trabajo.

En proyectos el branch sirve para que el miembro del equipo haga su trabajo en específico y así la colaboración es más fácil al momento de terminar el trabajo.

6. ¿Cuál es el nombre del branch principal?

El nombre del branch principal es Main.

2 Cómo se puede observar en el material de apoyo y con ayuda de lo mencionado en clase realice la investigación de 1 de gestor de GIT de forma local y realice el paso a paso de para su instalación y funcionamiento como lo muestra el ejemplo de Netbeans.

Instalación del gestor de GIT (GitHub Desktop)

* Buscar en el navegador GitHub Desktop
* Meterse en la página de GitHub Desktop y darle a descargar
* Después de descargar, abrir o ejecutar la aplicación
* Al entrar a la aplicación toca sincronizar con la cuenta de GitHub
* Autorizar la cuenta

FUNCIONAMIENTO

* Al entrar a la aplicación le aparecen los respectivos repositorios que uno tiene
* Elige el repositorio que quiere clonar y configura la carpeta en la que se clonara el repositorio
* Se le da a la opción clonar y prácticamente se estará clonando satisfactoriamente

3 • Crear cuenta en GitHub

• Clonar el repositorio de forma local en el equipo de trabajo

• Realizar el Readme definiendo el componente metodológico de

su proyecto individual.

• Crear un Branch llamado (Project).

• Crear por lo menos 3 Commits subiendo los trabajos realizados

en su proyecto.







