Nombre: Brandon René Portillo González
CUI: 3075926690603
Carnet: 999011994

1. **¿Qué es GIT?**
   Es un sistema de control de versiones distribuido, esto permite a los usuarios trabajar sobre repositorios locales sin necesidad de conexión o en forma remota.

2. **Control de versiones con GIT.**
   El control de versiones ayuda a realizar de los cambios realizados en archivos o códigos a lo largo del tiempo, esto permite disponer de un conjunto de escenarios que contienen características o propiedades que se modifican en un momento dado. Esto presenta diferentes ventajas tales como: Creación de flujos de trabajo, trabajo con versiones, código juntos, mantener un historial, automatización de tareas.

   Implementar GIT para un sistema de control de versiones es una buena opción ya que es una excelente opción para cada equipo.

3. **Estados de un archivo en GIT:**
   Git tiene tres estados principales en los que se pueden encontrar los archivos: confirmado (committed), modificado (modified), y preparado (staged). Confirmado: significa que los datos están almacenados de manera segura en tu base de datos local. Modificado: significa que se ha modificado el archivo pero todavía no se ha confirmado a la base de datos. Preparado: significa que se ha marcado un archivo modificado en la versión actual para que vaya en tu próxima confirmación.

4. **Como se configura un repositorio:**
   Hay dos opciones para crear un repositorio de Git. Puede crearse uno a partir del código de una carpeta de un equipo o clonar uno de un repositorio existente. Si se trabaja con código que se encuentra en el equipo local, cree un repositorio local mediante el código de esa carpeta. Pero la mayoría de las veces, el código ya se comparte en un repositorio de Git, por lo que clonar el repositorio existente en el equipo local es la manera recomendada de ir.

   - A partir de código existente:

     - Use el git init comando para crear un nuevo repositorio a partir de una carpeta existente en el equipo. En la línea de comandos, vaya a la carpeta raíz que contiene el código y ejecute:
     - > git init
     - para crear el repositorio. A continuación, agregue los archivos de la carpeta a la primera confirmación mediante los siguientes comandos:
     - > git add --all
     - > git commit -m "Initial commit"

- Desde el escritorio remoto:
  git clone es un comando para copiar el contenido de un repositorio existente en una carpeta del equipo. En la línea de comandos, vaya a la carpeta para que contenga el repositorio clonado y, a continuación, ejecute:

  > git clone

  Se debe asegurar la dirección URL real en el repositorio existente en lugar de la dirección URL del marcador de posición que se muestra en este ejemplo. Esta dirección URL, denominada dirección URL de clonación, apunta a un servidor donde el equipo coordina los cambios. Obtenga esta dirección URL del equipo o del botón clonar del sitio donde se hospeda el repositorio.

  No es necesario agregar archivos ni crear una confirmación inicial cuando se clone el repositorio, ya que todo se copió, junto con el historial, desde el repositorio existente durante la operación de clonación.

5. Comandos en GIT

   El comando git config sirve para definir valores de configuración de Git a nivel de un proyecto global o local. Estos niveles de configuración se corresponden con archivos de texto con extensión .gitconfig.

   Git init. Este comando inicializa un nuevo repositorio en el repositorio local.

   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 add. Una vez realizados los cambios necesarios en el área de trabajo (working area), para comenzar la confirmación de dichos cambios, es necesario pasar todos los archivos que queramos confirmar al área de preparación (staging area).

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

El comando para ver todas las confirmaciones realizadas en el repositorio es Git log.

Git diff es un comando permite ver las diferencias que existen entre las confirmaciones que se determinen. Se hace referenciando las confirmaciones mediante su número hash.

Se puede utilizar el comando Git revert para deshacer de forma segura una confirmación que ya se haya enviado.

Para listar una rama, el comando utilizado es git branch. Si se desea crear una nueva rama hay que especificar un nombre para dicha rama nueva.

Para conectar al repositorio remoto con el cual hay una comunicación entre el repositorio local y dicho repositorio remoto el comando utilizado es git remote.

Git status permite comprobar en cualquier momento el estado de la rama en la que se encuentra para comprobar si existen archivos (o directorios) que tienen cambios que deben de ser confirmados o rechazados.

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.

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

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

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.