UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE ESTUDIOS DE POSTGRADO
MAESTRIA EN INGENIRIA PARA LA INDUSTRIA CON ESPECIALIZACION EN CIENCIAS DE LA COMPUTACION
FUNDAMENTOS DE PROGRAMCION Y SCRIPTING

NOMBRE: LUIS ARMANDO DIAZ CIRAIZ **CARNET: 999012384**

INVESTIGACION TAREA 1

¿Qué es GIT?

Es un sistema de control de versiones distribuido, esto significa que un clon local del proyecto es un repositorio de control de versiones completo. Estos repositorios locales completamente funcionales permiten trabajar sin conexión o de forma remota. Los desarrolladores trabajan en su proyecto de manera local y a continuación sincronizan su copia del repositorio con la copia que existe del servidor. Este funcionamiento es distinto de los paradigmas de control de versiones centralizado, donde los clientes deben sincronizar el código con un servidor.

Conceptos Básicos

Cada vez que se guarda un trabajo Git crea una confirmación. Una confirmación es una instantánea para visualizar los archivos en un momento especifico.

Rama

Git permite hacer muchos cambios en función de la misma confirmación el el propio repositorio local, por lo que GIT proporciona herramientas para aislar los cambios y combinarlos posteriormente. Las ramas que son punteros ligeros en curso, administran esta separación. Una vez finalizado el trabajo creado en una rama se puede combinar de nuevo en la rama principal.

Archivos y confirmaciones

Los archivos de GIT se pueden encontrar en uno de los tres estados:

- Modificados
- Almacenados provisionalmente
- Confirmados

Cuando se modifica un archivo por primera vez los cambios solo existen en el directorio de trabajo, todavía no forman parte de una confirmación. El desarrollador debe almacenar provisionalmente los archivos modificados. El área de almacenamiento provisional contiene todos los archivos que se incluirán en la siguiente confirmación. Una vez que el desarrollador esta satisfecho con los archivos almacenados se empaquetan para la confirmación final.

Control de Versiones

Los sistemas de control de versiones son software que ayudan a realizar un seguimiento de los cambios realizados en el código a lo largo del tiempo. A medida que un desarrollador edita el código, el sistema de control de versiones toma una instantánea de los archivos. Después, guarda esa instantánea de forma permanente para que se pueda recuperar más adelante si es necesario.

Sin control de versiones, los desarrolladores tienen la tentación de mantener varias copias de código en su equipo. Esto es peligroso porque podría fácil cambiar o eliminar un archivo en la copia incorrecta del código, lo que haría perder trabajo. Los sistemas de control de versiones resuelven este problema.

Configuración de un Repositorio

Un repositorio de Git es una carpeta en la que Git realiza un seguimiento de los cambios. Puede haber cualquier número de repositorios en un equipo, cada uno almacenado en su propia carpeta. Y esto ayuda ya que cada repositorio en Git de un sistema es independiente, por lo que los cambios de uno no afectan al otro.

Git almacena las versiones del archivo en una carpeta oculta .git junto con otra información que necesita para administrar el código.

La mayoría de los equipos usan un repositorio central hospedado en un servidor al que todos los usuarios pueden acceder para coordinar sus cambios. El repositorio central normalmente se hospeda en una solución de administración de control de código fuente, como GitHub o Azure DevOps.

Creación de un repositorio de Git

Existen dos formas para crear un repositorio en Git. Uno es a partir de código de una carpeta en un equipo o clonar uno de un repositorio existente. Si se trabaja con código de un equipo local, hay que crear un repositorio local mediante el código de esa carpeta, pero la mayoría de veces el código ya se comparte en un repositorio de Git.

Creación de un repositorio a partir de código existente

Use el git init para crear un nuevo repositorio apartir de una carpeta existente. En la l{inea de comandos vaya a la carpeta raíz que contiene el código y ejecute

> git init

luego

> git add—all

> git commit -m “initial commit”

Creación de un repositorio desde un repositorio remoto

Se usa el comando git clone como en el ejemplo siguiente:

> git clone
https://<fabrikam.visualstudio.com/DefaultCollection/Fabrikam/_git/FabrikamProject>

Guardado y uso compartido de código con GIT

Guardar y compartir versiones de código con un equipo son las cosas más comunes que se realizan al usar el control de versiones. Git tiene un flujo de trabajo sencillo de tres pasos para estas tareas:

1. Creación de una rama para el trabajo

2. Confirmación de cambios
3. Inserción de la rama para compartirla con el equipo

Crear una rama

Cree una rama basada en el código de una rama actual, como main, al iniciar un nuevo trabajo. Se recomienda comprobar qué rama está seleccionada mediante git status antes de crear una nueva rama.

Cree ramas en Git mediante el git branch comando:

> git branch <branchname>

El comando que se va a intercambiar entre ramas del repositorio es git checkout.

> git checkout <branchname>

Para crear la rama y cambiar a ella al mismo tiempo:

> git checkout -b <branchname>