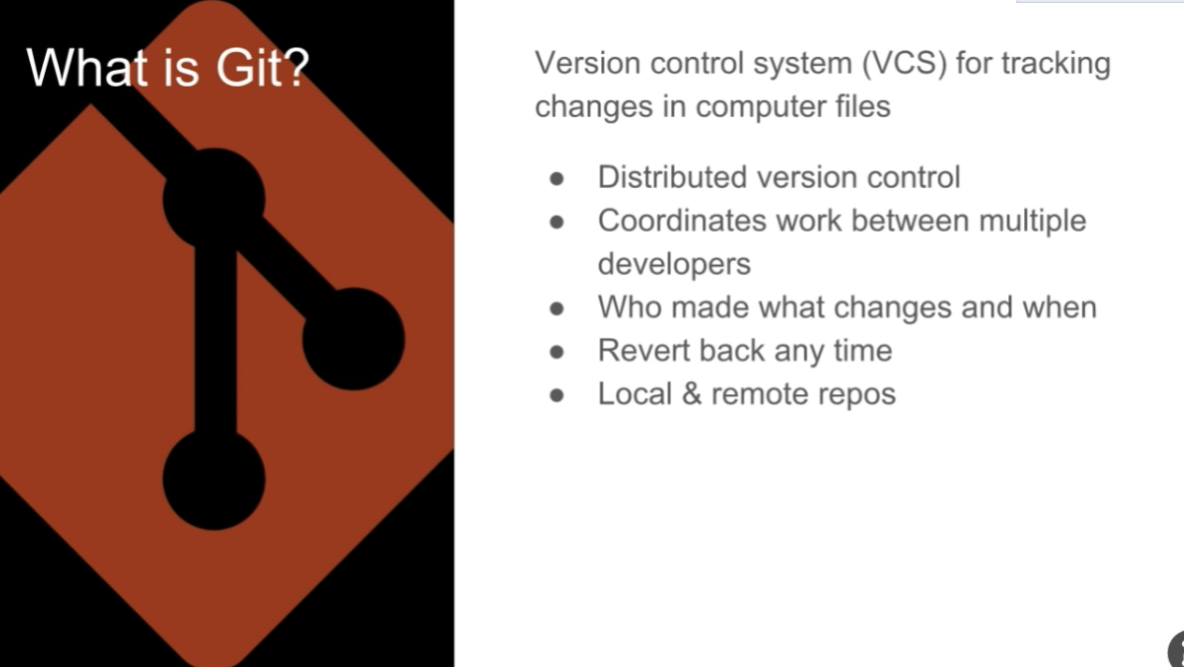
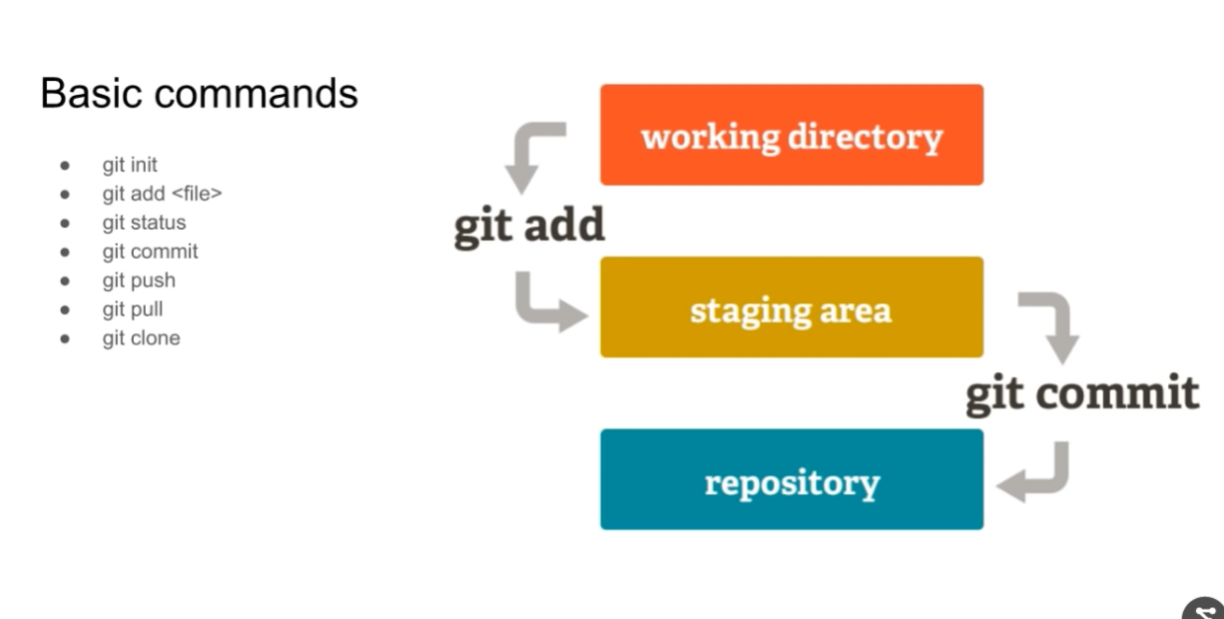
**GIT**





3 estados:

Working -> Area donde yo estare trabajando con todos mis archivos. En local , en mi pc

Stating área-> Es donde se agregaran todos los archivos donde se preparan para el guardado, para la lista de versiones

Repository -> y cuando ya estamos listos ya pasamos todo a el repositorio de lo que hay en el stating área

COMANDOS

\* git init Oye Git, voy a usar estos documentos contigo ¿vale?

\* git add <file> Pasa los docs a staging area

\* git add . Pasa todos los archivos

\* git commit Pasa los docs de staging area a repository (Después de esto se te va a abrir el editor de código VIN en donde tendrás que escribir un comentario, si te quieres evitar abrir VIN entonces utiliza los siguientes comandos:)

\* git commit -m "comentario" Lo mismo que el commit regular, pero ahora no necesitas entrar a VIN

\* git push Subir los docs a un server (Github)

\* git pull Traer los docs de un server, traer los cambios de tus compañeros

\* git status Ver en que status (wd, sa, r) están los docs

\* git clone Hacerte una copia de lo que está en el server a tu PC

\* git checkout -- <file> Para revertir los cambios de los archivos

\* git diff <file> Para ver las diferencias hechas en los archivos

\* git branch Ver las ramas que hay ("master" es la rama default)

\* git branch "nombre" Crear una nueva rama

\* git checkout "nombre" Ir a una rama en especifico

\* git config -- global user.email "email" Para configurar email del usuario

\* git config -- global user.name "nombre" Para configurar nombre del usuario

vin Es el editor de código de git desde la consola, ahí escribes un comentario para la nueva versión que estés versionando (si no te deja escribir presiona a letra i . Cuando termines presiona esc y luego :wq (write & quit)pasa salir.

.gitignore Es un archivo reservado de git que tenemos que crear si queremos decirle a git los archivos que no vamos a utilizar y así decida ignorarlos.

Escribe dentro de el archivo .gitignore los nombres de los archivos que desees ignorar.

RAMAS

Git chekout –b nombre de la rama: nos permite crear una nueva rama y cambiar de rama a la misma vez con un solo comando

Git branch –m nombreActualDeLaRama NombreNuevaDeLaRama: para cambiar el nombre de la rama

Git branch –d NombreDeLaRamaQueQuieroEliminar: eliminar rama

Git chekout –b nombre de la rama: nos permite crear una nueva rama y cambiar de rama a la misma vez con un solo comando

Git branch –m nombreActualDeLaRama NombreNuevaDeLaRama: para cambiar el nombre de la rama

Git branch –d NombreDeLaRamaQueQuieroEliminar: eliminar rama