**Rodríguez Morales Paloma Estefanía**

**Programación Web**

**Tarea: 1**

|  |  |  |
| --- | --- | --- |
| a) Buscando Ayuda: | 1. **git help** comando   ó  **git** comando –help | Muestra la ayuda para ese comando |
| b) Creación de un repositorio: | **Git init** | Crea un repositorio en el directorio actual |
|  | **gi**t clone url | Clona un repositorio remoto dentro de un directorio |
| c) Operaciones sobre Archivos: | Git add path | Adiciona un archivo o un directorio de manera recursiva |
|  | Git rm ruta | Remueve un archivo o directorio del árbol de trabajo |
|  | -f : | Fuerza la eliminación de un archivo del repositorio |
|  | git mv origen destino | Mueve el archivo o directorio a una nueva ruta |
|  | -f : | Sobre-escribe los archivos existentes en la ruta destino |
|  | **Git checkout** [rev] archivo | Recupera un archivo desde la rama o revisión actual |
| d) Trabajando sobre el código: |  |  |
|  | 8. **git status** | Imprime un reporte del estado actual del árbol de trabajo local |
|  | 9. **git diff** [ruta] | Muestra la diferencia entre los cambios en el árbol de trabajo local |
|  | 10. **git d**iff HEAD ruta | Muestra las diferencias entre los cambios registrados y los no registrados |
|  | 11. **git add** path | Selecciona el archivo para que sea incluido en el próximo commit |
|  | 12. **git reset** HEAD ruta | Marca el archivo para que no sea incluido en el próximo commit |
|  | 13. **git commit** | Realiza el commit de los archivos que han sido registrados (con git-add) |
|  | -a : | Automáticamente registra todos los archivos modificados |
|  | 14. **git reset** --soft HEAD^ | Deshace commit & conserva los cambios en el árbol de trabajo local |
|  | 15. **git reset** --hard HEAD^ | Restablece el árbol de trabajo local a la versión del ultimo commit |
|  | 16. **git clean** | Elimina archivos desconocidos del árbol de trabajo local |
| e) Examinando el histórico: |  |  |
|  | 17. **git log** [ruta] | Muestra el log del commit, opcionalmente de la ruta especifica |
|  | 18. **git log** [desde [..hasta]] | Muestra el log del commit para un rango de revisiones dado |
|  | --stat : | Lista el reporte de diferencias de cada revisión |
|  | -S'pattern' : | Busca el historial de cambios que concuerden con el patrón de búsqueda |
|  | 19. **git blame** [archivo] | Muestra el archivo relacionado con las modificaciones realizadas |
| f) Repositorios remotos: |  |  |
|  | 20. **git fetch** [remote] | Trae los cambios desde un repositorio remoto |
|  | 21. **git pull** [remote] | Descarga y guarda los cambios realizados desde un repositorio remoto |
|  | 22. **git push** [remote] | Guarda los cambios en un repositorio remoto |
|  | 23. **git remote** | Lista los repositorios remotos |
|  | 24. **git remote add** remote url | Añade un repositorio remoto a la lista de repositorios registrados |
|  |  |  |
| g) Ramas: |  |  |
|  | 25. **git checkout** rama | Cambia el árbol de trabajo local a la rama indicada |
|  | -b rama : | Crea la rama antes de cambiar el árbol de trabajo local a dicha rama |
|  | 26. **git branch** | Lista las ramas locales |
|  | 27. **git branch** -f rama rev | Sobre-escribe la rama existente y comienza desde la revisión |
|  | 28. **git merge** rama | Guarda los cambios desde la rama |
| h) Exportando e importando: |  |  |
|  | 29. **git apply** - < archivo | Aplica el parche desde consola (stdin) |
|  | 30. **git format-patch** desde [..hasta] | Formatea un parche con un mensaje de log y un reporte de diferencias (diffstat) |
|  | 31. **git archive** rev > archivo | Exporta resumen de la revisión (snapshot) a un archivo |
|  | --prefix=dir/ | : Anida todos los archivos del snapshot en el directorio |
|  | --format=[tar|zip] | : Especifica el formato de archivo a utilizar: tar or zip |
| i) Etiquetas: |  |  |
|  | 32. **git tag** name [revision] | Crea una etiqueta para la revisión referida |
|  | -s : Firma la etiqueta con su llave privada usando GPG |  |
|  | -l [patrón] | : Imprime etiquetas y opcionalmente los registros que concuerden con el patrón de búsqueda |
| j) Banderas de Estado de los Archivos: | M (modified) : El archivo ha sido modificado  C (copy-edit) : El archivo ha sido copiado y modificado  R (rename-edit) : El archivo ha sido renombrado y modificado  A (added) : El archivo ha sido añadido  D (deleted) : El archivo ha sido eliminado  U (unmerged) : El archivo presenta conflictos después de ser guardado en el servidor (merge) |  |

**Crear un Repositorio Nuevo**

|  |  |
| --- | --- |
| Crea un directorio nuevo, ábrelo y ejecuta  para crear un nuevo repositorio de git. | git init |

**Hacer Chekout a un Repositorio**

|  |  |
| --- | --- |
| Crea una copia local del repositorio ejecutando | git clone /path/to/repository |
| Si utilizas un servidor remoto, ejecutar. | **git clone username@host:/path/to/repository** |

**Add & commit**

|  |  |
| --- | --- |
| Para registrar cambios (añadirlos al Index) | git add <filename>  git add . |
| Este es el primer paso en el flujo de trabajo básico. Para hacer commit a estos cambios | git commit -m "Commit message" |

Ahora el archivo esta incluido en el HEAD, pero aún no en tu repositorio remoto.

**Envió de Cambios**

|  |  |
| --- | --- |
| Para enviar cambios a tu repositorio remoto | Git push origin master |

Reemplaza master por la rama a la que quieres enviar tus cambios.

|  |  |
| --- | --- |
| Si no has clonado un repositorio ya existente y quieres conectar tu repositorio local a un repositorio remoto. | git remote add origin <server> |

Ahora podrás subir tus cambios al repositorio remoto seleccionado.

**Ramas**

|  |  |
| --- | --- |
| Crea una nueva rama llamada "feature\_x" y cámbiate a ella usando | git checkout -b feature\_x |
| Volver a la rama principal | **Git checkut master** |
| Borrar la Rama | **Git branch -d feature\_x** |
| Una rama nueva no estará disponible para los demás a menos que subas (push) la rama a tu repositorio remoto | **git push origin <branch>** |

|  |  |
| --- | --- |
|  |  |

Bibliografía:

<http://www.maefloresta.com/portal/es/git.es>

http://rogerdudler.github.io/git-guide/index.es.html