



- INSTALACIÓN
- COMANDOS BÁSICOS UTILIZANDO GIT BASH
- SUBIR ARCHIVOS A REPOSITORIO LOCAL
- SUBIR ARCHIVOS A GITHUB

Prof. Juan Pablo Nardone
Curso Desarrollo Web Full Stack Python

ÍNDICE

Instalación de GIT	2
Comandos básicos para la terminal, moverse entre directorios	2
Comandos básicos para la terminal, crear carpetas, archivos, eliminarlos	3
Inicializar carpeta y configurar usuarios	3
Subir archivo a un repositorio local	3
Subir archivos a GitHub	4
Creando el repositorio desde GitHub	4
Subiendo a repositorio local	4
Subiendo a repositorio de GitHub.....	4
Actividad integradora.....	5
Etapa 1: Preparando las carpetas y archivos	5
Etapa 2: Configurar los usuarios y subir los archivos al repositorio local.	6
Etapa 3: Subir archivos al repositorio local.	6
Etapa 4: Subir archivos al repositorio remoto GitHub.	6
Actividad integradora (solución)	6
Etapa 1: Preparando las carpetas y archivos	6
Etapa 2: Configurar los usuarios y subir los archivos al repositorio local.	7
Etapa 3: Subir archivos al repositorio local.	7
Etapa 4: Subir archivos al repositorio remoto GitHub.	8

Instalación de GIT

Pasos para realizar la instalación de GIT (v. 2.35.1.2)¹

- 1) Ingresar en <https://git-scm.com/> y hacer clic en Download (detecta que SO tenemos)
- 2) Elegir qué versión de Windows tenemos y descargar. Luego ejecutar el archivo para iniciar la instalación (luego de cada paso presionar Next o Siguiente):
 - a. En el caso de que ya estaba instalado poner "Sí" en la primera ventana.
 - b. Marcar que haya un acceso en el escritorio.
 - c. Setear cuál va a ser la carpeta de inicio.
 - d. Seleccionar Visual Studio Code como editor de texto por defecto.
 - e. Dejar que Git decida cuál va a ser la rama por defecto (Master).
 - f. Dejar que se use la línea de comandos de Windows.
 - g. Dejar marcada la opción Use bundled OpenSSH.
 - h. Dejar marcada la opción Use bundled OpenSSL library.
 - i. Dejar marcada la primera opción Checkout Windows - style...
 - j. Dejar marcada Use MinTTY (dejar esta terminal por defecto).
 - k. Dejar en Default (otras configuraciones más avanzadas).
 - l. Dejamos Git Credential Manager.
 - m. Dejamos la primer opción marcada (Enable file system caching).
 - n. En el último paso no marcamos nada y ponemos **Install**.
- 3) Al terminar, en el mensaje que aparece ponemos que queremos lanzar Git Bash y nos aparecerá la consola. Con git --version podemos ver la versión:

```
$ git --version
git version 2.35.1.windows.2
```

Fuente: <https://youtu.be/ptXiQwE535s>

Comandos básicos para la terminal, moverse entre directorios



Estos comandos nos permiten movernos entre directorios (carpetas) dentro de la terminal. Para utilizarlos debemos abrir la terminal desde Git Bash.

Comando	Descripción
pwd	Ver en qué carpeta estamos ubicados
git --version	Ver la versión de Git que tenemos instalada
ls	Me permite ver qué tengo en la carpeta, archivos y subcarpetas (como un dir en Windows)
ls -l	Permite saber si son archivos o directorios (más detalle)
ls -lh	Lo mismo que antes pero mostrando el peso del archivo
ls -la	Muestra archivos y carpetas ocultos
clear o CTRL+L	Limpiamos la pantalla
cd carpeta	Me permite entrar en la subcarpeta (<i>cd: change directory</i>)
cd ..	Me permite regresar a la carpeta anterior (subir de nivel)
cd c:\	Ingresamos en el disco C
exit	Salimos de la terminal

Fuente: <https://www.youtube.com/watch?v=SlotISUq-3w>

¹ Última actualización: 12 de abril de 2022

Comandos básicos para la terminal, crear carpetas, archivos, eliminarlos



Estos comandos nos permiten realizar lo mismo que haríamos desde un Sistema Operativo (crear carpetas, archivos, ver su contenido, eliminarlos). Para utilizarlos debemos abrir la terminal desde Git Bash.

Comando	Descripción
mkdir nombrecarpeta	Nos permite crear una carpeta.
touch archivo.ext	Permite crear el archivo, donde ext es la extensión.
cat archivo.ext	Muestra el contenido del archivo, donde ext es la extensión.
rm archivo.ext	Elimina el archivo, donde ext es la extensión.
rm -r nombrecarpeta/	Elimina la carpeta
mv nombreoriginal.ext nombrenuevo.ext	Cambia el nombre a un archivo, donde ext es la extensión.
mv nombreoriginal nombrenuevo	Cambia el nombre a una carpeta

Fuente: <https://youtu.be/W-z0iTXEarw>

Inicializar carpeta y configurar usuarios



Estos comandos nos permiten configurar el usuario y el mail para el proyecto. Para utilizarlos debemos abrir la terminal desde Git Bash.

Comando	Descripción
git init	Inicializar la carpeta actual.
git config --global user.name "nombredeusuario"	Configurar el usuario.
git config --global user.mail "nombredeusuario@mail.com"	Configurar el mail del usuario.
git config user.name	Mostrar el nombre de usuario.
git config user.mail	Mostrar el mail del usuario.

Fuente: <https://youtu.be/KAY1j-MPpyA>

Subir archivo a un repositorio local



Estos comandos nos permiten subir los archivos a un repositorio local (en nuestra computadora), además de ver el estado de los mismos. Para utilizarlos debemos abrir la terminal desde Git Bash.

Recordemos:

- **Directorio de trabajo:** Se inicializa con git init y es el que contendrá mis archivos.
- **Staging area:** Es donde se almacenarán los archivos (memoria de GIT), el archivo aún no está subido.
- **Repositorio local:** Es donde se almacenan los archivos subidos.

Comando	Descripción
git add nombredelarchivo.ext	Agrega al staging área (memoria de GIT) el archivo deseado.
git commit -m "comentario"	Agrega el archivo al repositorio local desde el staging área. Siempre va acompañado de un breve comentario (no más de 140 caracteres).
git status	Indica el estado del staging área.
git log	Muestra los archivos subidos a la rama principal.

Fuente: <https://youtu.be/NRaLP7vtoOY>

Subir archivos a GitHub



Estos comandos nos permiten subir los archivos a un repositorio local (en nuestra computadora), además de ver el estado de los mismos. Para utilizarlos debemos abrir la terminal desde Git Bash.

Comando	Descripción
git remote add origin https://github.com/usuario/repositorio.git	Seteamos cuál será el repositorio de GitHub al que vamos a subir nuestros archivos
git push origin master	Permite subir el/los archivos al repositorio remoto en GitHub

Creando el repositorio desde GitHub

Antes que anda tenemos que tener nuestra cuenta activada de GitHub.

- 1) Ingresar en la cuenta, ir al + y elegir **New Repository**
- 2) Colocar un nombre al repositorio y una descripción
- 3) Dejamos el repositorio público y no marcar ninguna de las opciones de más abajo
- 4) Hacemos clic en **Create Repository**

Aparecerá una página que nos dice que tenemos creado nuestro repositorio.

Tenemos que copiar la dirección del repositorio, donde lo vamos a alojar. Ejemplo:

<https://github.com/NombredeUsuario/repositorio.git>

Subiendo a repositorio local

Recordar que primero debemos hacer los comandos para agregar al repositorio local y luego del local los subiremos al repositorio remoto:

- 1) Abrimos Git Bash
- 2) Ir a la carpeta donde tenemos el repositorio (si lo tenemos creado en el disco C debemos llegar a C con **cd /c** e ingresar con **cd nombredelrepositorio**)
- 3) Inicializar la carpeta con **git init** (la convertiremos en la rama principal)
- 4) Crearemos un archivo llamado **pagina.html** con el comando **touch pagina.html**, la abriremos en VSC y le haremos algún cambio que luego guardaremos.
- 5) Agregaremos el archivo al repositorio con **git add pagina.html** y haremos un commit con **git commit -m "Subiendo a repositorio local"**
- 6) Comprobaremos que haya quedado subido al repositorio local con **git log**

Subiendo a repositorio de GitHub

- 1) Para subir a nuestro repositorio primero seteamos cuál será el repositorio de GitHub al que subiremos nuestro archivo.
git remote add origin https://github.com/NombredeUsuario/repositorio.git
- 2) Para comprobar escribiremos **git remote -v** que nos dirá que tenemos un lugar de origen donde podemos hacer un fetch y un push.
- 3) Para subir los archivos hacemos **git push origin master** y luego de unos segundos veremos que los subió a GitHub.
- 4) Para comprobar esto vamos al sitio de GitHub, a nuestro repositorio y veremos que se ha subido el archivo (actualizar con F5).
- 5) Si queremos repetir estos pasos con otros archivos debemos hacer lo siguiente:
 - a. Crear el nuevo archivo (puede ser **estilos.css**).
 - b. Agregarlo al repositorio local con **git add estilos.css**.

- c. Hacer un commit con `git commit m "Subiendo página de estilos"`.
- d. Hacer `git push origin master` y actualizar la página de Github.

Fuente: <https://youtu.be/9p7Avpcs7Bk>

Actividad integradora

En esta actividad...

- Administraremos las carpetas y archivos para un proyecto Web simple (crear, cambiar nombre, eliminar).
- Navegaremos entre carpetas y veremos los contenidos de los archivos.
- Inicializaremos la carpeta para subir nuestros repositorios locales, configuraremos los usuarios y subiremos los archivos al repositorio local.
- Haremos modificaciones a los archivos y los actualizaremos en el repositorio local.
- Agregaremos estos archivos al repositorio remoto.

Realizaremos todas estas tareas desde Git Bash, por lo tanto, te invitamos a que la abras en tu Sistema Operativo.

Etapa 1: Preparando las carpetas y archivos

- 1) Verificar el directorio en donde estamos ubicados.
- 2) Si no estamos ubicados en disco C ingresar en él.
- 3) Crear la carpeta proyectogit.
- 4) Verificar que se haya creado la carpeta.
- 5) Ingresar en la carpeta proyectogit.
- 6) Crear cuatro subcarpetas llamadas img, css, js y bbdd.
- 7) Crear un archivo index.html
- 8) Verificar con más nivel de detalle si el archivo y carpetas se encuentran en la carpeta del proyecto.
- 9) Ingresar en la carpeta css, comprobar que esté vacía.
- 10) Crear un archivo styles.css y un archivo notas.txt
- 11) Abrir desde el explorador el archivo notas.txt, agregar el texto "Este es un archivo de notas para mi proyecto". Guardar y cerrar
- 12) Abrir el archivo notas desde Git Bash
- 13) Renombrar el archivo styles.css por estilos.css
- 14) Eliminar el archivo notas.txt
- 15) Volver a la carpeta proyectogit
- 16) Renombrar la carpeta img por imgs
- 17) Eliminar la carpeta bbdd
- 18) Abrir desde el editor de código de tu preferencia el archivo index.html. Agregar una estructura básica vinculando el archivo html con el archivo css, agregar un encabezado, un párrafo y guardar los cambios.
- 19) Volver a Git Bash y consultar los archivos y carpetas que tenemos, mostrando el peso del archivo
- 20) Abrir el archivo index.html desde Git Bash
- 21) Limpiar la consola
- 22) Salir de la consola

Etapas 2: Configurar los usuarios y subir los archivos al repositorio local.

- 1) Verificar si estamos ubicados en la carpeta proyectogit.
- 2) Inicializar la carpeta proyectogit. Recordar que la palabra master que es la rama principal del repositorio local.
- 3) Comprobar que se ha inicializado la carpeta (recordar que se verán los archivos y carpetas ocultos).
- 4) Configurar el usuario.
- 5) Configurar el e-mail.
- 6) Confirmar que han quedado configurados.

Etapas 3: Subir archivos al repositorio local.

- 1) Verificar si estamos ubicados en la carpeta proyectogit.
- 2) Agregar al staging area el archivo index.html.
- 3) Comprobar que los archivos se han subido al staging area.
- 4) Hacer un commit agregando el index.html, agregar un comentario.
- 5) Comprobar que los archivos ya no están en el staging area, pero nos queda pendiente subir el resto de las carpetas que contienen archivos (en este caso estilos.css).
- 6) Agregar al staging area el resto de las carpetas y archivos, luego comprobar que están en el staging area
- 7) Hacer un commit para el resto de los elementos, agregar un comentario.
- 8) Comprobar los dos commits mostrando los archivos subidos a la rama principal
- 9) Abrir desde el editor de código de tu preferencia el archivo estilos.css. Agregar estilos al encabezado y al párrafo.
- 10) Volver a Git Bash, dirigirse a la carpeta css y verificar que el archivo css tenga contenido. Luego agregar el archivo estilos.css al staging area, luego hacer un commit con un comentario adecuado. Puede verificar los commits si lo desea.

Etapas 4: Subir archivos al repositorio remoto GitHub.

- 1) Ingresar en nuestra cuenta de GitHub y crear un repositorio público nuevo (nombre y descripción)
- 2) Volver a Git Bash y setear cuál será el repositorio de GitHub al que subiremos nuestro archivo (en este caso la dirección URL del repositorio recién creado).
- 3) Comprobar que existe ese repositorio al cual queremos subir los archivos.
- 4) Subir los archivos al repositorio remoto
- 5) Volver a GitHub y actualizar la página para ver que los archivos fueron subidos.
- 6) Hacer algún cambio en el archivo html desde el editor de código, buscar algún archivo de imagen y guardarlo en la carpeta imgs
- 7) Agregar todo al staging area, verificar los cambios y hacer un commit con un comentario. Agregar los cambios al repositorio de GitHub

Actividad integradora (solución)

Etapas 1: Preparando las carpetas y archivos

- | | |
|---|--------------------------------|
| 1) Verificar el directorio en donde estamos ubicados. | <code>pwd</code> |
| 2) Si no estamos ubicados en disco C ingresar en él. | <code>cd /c</code> |
| 3) Crear la carpeta proyectogit. | <code>mkdir proyectogit</code> |
| 4) Verificar que se haya creado la carpeta. | <code>ls</code> |

5) Ingresar en la carpeta proyectogit.	<code>cd proyectogit</code>
6) Crear cuatro subcarpetas llamadas img, css, js y bbdd.	<code>mkdir img</code> <code>mkdir css</code> <code>mkdir js</code> <code>mkdir bbdd</code>
7) Crear un archivo index.html	<code>touch index.html</code>
8) Verificar con más nivel de detalle si el archivo y carpetas se encuentran en la carpeta del proyecto.	<code>ls -l</code>
9) Ingresar en la carpeta css, comprobar que esté vacía.	<code>cd css</code> <code>ls</code>
10) Crear un archivo styles.css y un archivo notas.txt	<code>touch styles.css</code> <code>touch notas.txt</code>
11) Abrir desde el explorador el archivo notas.txt, agregar el texto "Este es un archivo de notas para mi proyecto". Guardar y cerrar	-
12) Abrir el archivo notas desde Git Bash	<code>cat notas.txt</code>
13) Renombrar el archivo styles.css por estilos.css	<code>mv styles.css</code> <code>estilos.css</code>
14) Eliminar el archivo notas.txt	<code>rm notas.txt</code>
15) Volver a la carpeta proyectogit	<code>cd ..</code>
16) Renombrar la carpeta img por imgs	<code>mv img imgs</code>
17) Eliminar la carpeta bbdd	<code>rm -r bbdd/</code>
18) Abrir desde el editor de código de tu preferencia el archivo index.html. Agregar una estructura básica vinculando el archivo html con el archivo css, agregar un encabezado, un párrafo y guardar los cambios.	-
19) Volver a Git Bash y consultar los archivos y carpetas que tenemos, mostrando el peso del archivo	<code>ls -lh</code>
20) Abrir el archivo index.html desde Git Bash	<code>cat index.html</code>
21) Limpiar la consola	<code>clear</code>
22) Salir de la consola	<code>exit</code>

Etapas 2: Configurar los usuarios y subir los archivos al repositorio local.

1) Verificar si estamos ubicados en la carpeta proyectogit.	<code>pwd</code>
2) Inicializar la carpeta proyectogit. Recordar que la palabra master que es la rama principal del repositorio local.	<code>git init</code>
3) Comprobar que se ha inicializado la carpeta (recordar que se verán los archivos y carpetas ocultos).	<code>ls -la</code>
4) Configurar el usuario.	<code>git config --global user.name</code> <code>"prof. juan pablo"</code>
5) Configurar el e-mail.	<code>git config --global user.mail</code> <code>"profjuanpablo@mail.com"</code>
6) Confirmar que han quedado configurados.	<code>git config user.name</code> <code>git config user.mail</code>

Etapas 3: Subir archivos al repositorio local.

1) Verificar si estamos ubicados en la carpeta proyectogit.	<code>pwd</code>
2) Agregar al staging area el archivo index.html.	<code>git add index.html</code>

3) Comprobar que los archivos se han subido al staging area.	<code>git status</code>
4) Hacer un commit agregando el index.html, agregar un comentario.	<code>git commit -m "agregamos archivo html"</code>
5) Comprobar que los archivos ya no están en el staging area, pero nos queda pendiente subir el resto de las carpetas que contienen archivos (en este caso estilos.css).	<code>git status</code>
6) Agregar al staging area el resto de las carpetas y archivos, luego comprobar que están en el staging area	<code>git add .</code> <code>git status</code>
7) Hacer un commit para el resto de los elementos, agregar un comentario.	<code>git commit -m "agregamos la carpeta css y el archivo de estilos"</code>
8) Comprobar los dos commits mostrando los archivos subidos a la rama principal	<code>git log</code>
9) Abrir desde el editor de código de tu preferencia el archivo estilos.css. Agregar estilos al encabezado y al párrafo.	-
10) Volver a Git Bash, dirigirse a la carpeta css y verificar que el archivo css tenga contenido. Luego agregar el archivo estilos.css al staging area, luego hacer un commit con un comentario adecuado. Puede verificar los commits si lo desea.	<code>Cd css</code> <code>Cat estilos.css</code> <code>Git add estilos.css</code> <code>git commit -m "Agregado al archivo de estilos"</code> <code>git log</code>

Etapas 4: Subir archivos al repositorio remoto GitHub.

1) Ingresar en nuestra cuenta de GitHub y crear un repositorio público nuevo (nombre y descripción)	-
2) Volver a Git Bash y setear cuál será el repositorio de GitHub al que subiremos nuestro archivo (en este caso la dirección URL del repositorio recién creado).	<code>git remote add origin https://github.com/JuanPabloNardone/poryectogit.git</code>
3) Comprobar que existe ese repositorio al cual queremos subir los archivos.	<code>git remote -v</code>
4) Subir los archivos al repositorio remoto	<code>git push origin master</code>
5) Volver a GitHub y actualizar la página para ver que los archivos fueron subidos.	-
6) Hacer algún cambio en el archivo html desde el editor de código, buscar algún archivo de imagen y guardarlo en la carpeta imgs	-
7) Agregar todo al staging area, verificar los cambios y hacer un commit con un comentario. Agregar los cambios al repositorio de GitHub	<code>Git add .</code> <code>Git status</code> <code>git commit -m "Mas cambios hechos por Juan"</code> <code>git push origin master</code>