

# **DESPLIEGUE DE APLICACIONES WEB**

**SISTEMA DE  
CONTROL DE  
VERSIONES**

Pablo Cidón Barrio  
Curso 2017-2018

## Sumario

Actividad 1.- Instalación de Gitlab en una máquina virtual en Ubuntu Server.....	3
Actividad 2.- Instalación de cliente git en Windows, <a href="https://git-scm.com/downloads">https://git-scm.com/downloads</a> .....	4
Actividad 3.- Trabajando con Git.....	7
Actividad 4.- Trabajando con ramas.....	10
Actividad 5.- Ramas y fusiones en local (FAST-Forward).....	10
a) Borrar el repositorio remoto DAW1718.....	10
b) Borrar el repositorio local DAW1718.....	11
c) Crear de nuevo el proyecto HOLAMUNDO, con un commit con el archivo README.md que contiene el nombre de la asignatura.....	11
d) Clonación del proyecto en local.....	12
e) Crear una página web, index.html, con en mensaje “Hola Mundo” sin estilos. Una vez terminada, subela al repositorio remoto.....	13
d) Añade la etiqueta v1.0 al commit anterior.....	15
.....	15
e) Se piden crear dos diseños con estilos(css/estilo.css) diferentes del index.html, para ello realizamos dos ramas en local, develop1 y develop2.....	15
.....	15
d) Una vez que el profesor decida que diseño es el más adecuado, se realizará la fusión en la master.....	15
.....	15
e) Se sube al servidor la rama master.....	15
.....	15
d) Se puede borrar las ramas develop1 y develop2.....	15
g) Se etiqueta el nuevo commit, como v1.1.....	15

# Actividad 1.- Instalación de Gitlab en una máquina virtual en Ubuntu Server

Máquina Virtual: PCB-GIT		
Sistema Operativo	Ubuntu Server 16.04.1 LTS x64	
Memoria RAM	Según los requisitos de instalación	
Particiones	Disco dinámico de 500GB	Sistema: 100GB Swap: 2GB Datos: 298GB
Nombre de la máquina	PCB-GIT	
Usuarios y contraseñas	miadmin	paso
	usuario operadorweb	paso paso
Configuración de red	IP: 192.168.3.150+nº de equipo/24 Puerta de enlace: 192.168.3.1 DNS: 192.168.20.20	

Partimos de la máquina que tenemos de entorno de desarrollo.

Realizamos la configuración de red:

```
iface enp0s3 inet static
    address 192.168.3.161
    netmask 255.255.255.0
    gateway 192.168.3.1
    dns-nameservers 192.168.20.20
```

También cambiamos el nombre del servidor:

```
PCB-GIT
```

Procedemos a la instalación del repositorio Git:

Actualizamos el repositorio:

```
miadmin@PCB-USED:~$ sudo apt-get update
Des:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu xenial InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Descargados 306 kB en 2s (134 kB/s)
```

Descargamos el servidor openssh:

```
miadmin@PCB-GIT:~$ sudo apt-get install -y curl openssh-server ca-certificates
[sudo] password for miadmin:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Luego instalaremos el Postfix:

```
miadmin@PCB-GIT:~$ sudo apt-get install -y postfix
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Instalamos el repositorio de GitLab:

```
miadmin@PCB-GIT:~$ curl -sS https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh | sudo bash
```

Instalamos el paquete de GitLab:

```
miadmin@PCB-GIT:~$ sudo apt-get install gitlab-ce
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

<https://about.gitlab.com/installation/#ubuntu>

## Actividad 2.- Instalación de cliente git en Windows, <https://git-scm.com/downloads>.

Nos dirigimos a la página de GIT, y automáticamente se iniciará la descarga automáticamente.

### Downloading Git



#### Your download is starting...

You are downloading the latest (2.14.1) 64-bit version of Git for Windows. This is the most recent [maintained build](#). It was released [about 2 months ago](#), on 2017-08-10.

If your download hasn't started, [click here to download manually](#).

#### Other Git for Windows downloads

Git for Windows Setup

[32-bit Git for Windows Setup](#).

[64-bit Git for Windows Setup](#).

Git for Windows Portable ("thumbdrive edition")

[32-bit Git for Windows Portable](#).

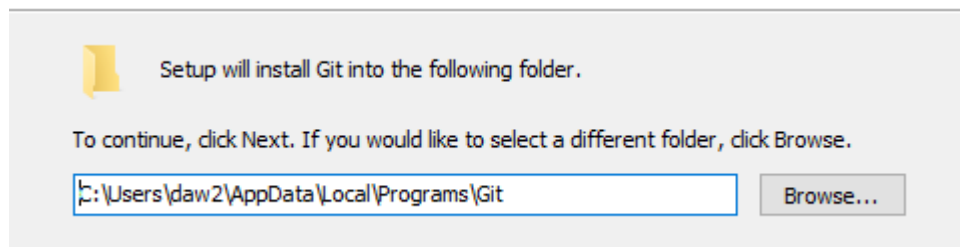
[64-bit Git for Windows Portable](#).

The current source code release is version 2.14.2. If you want the newer version, you can build it from [the source code](#).

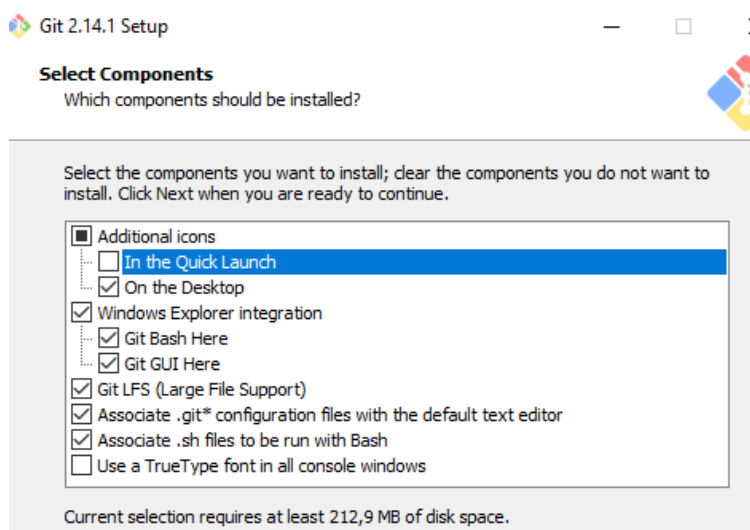
Una vez descargado el paquete lo ejecutamos y comenzaremos la instalación.



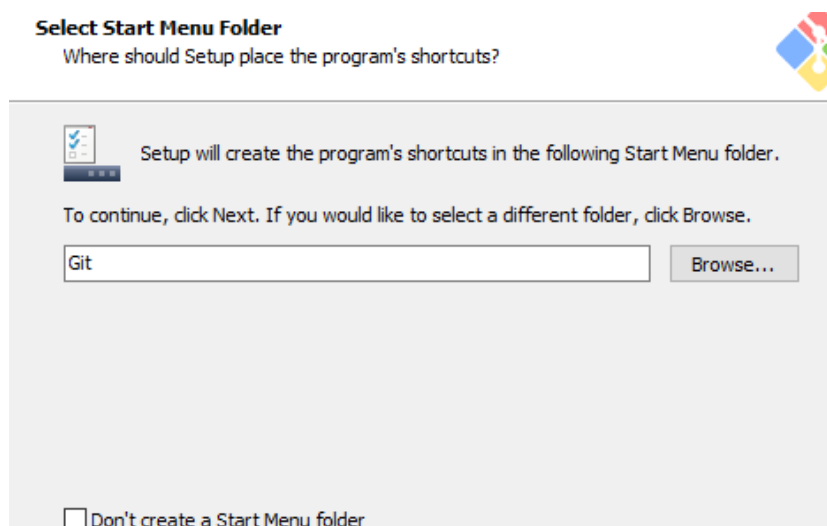
En primer lugar nos aparecerá la ubicación en la que instalaremos el programa. Aunque podremos dejar la que viene por defecto.



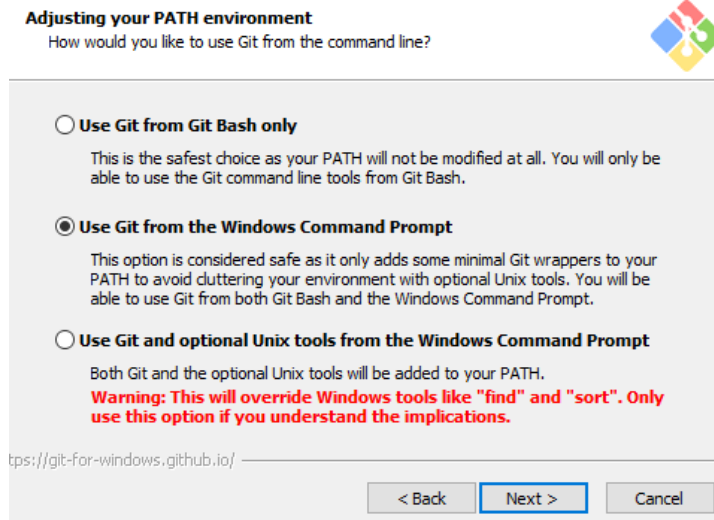
A continuación, elegiremos los componentes que querramos instalar:



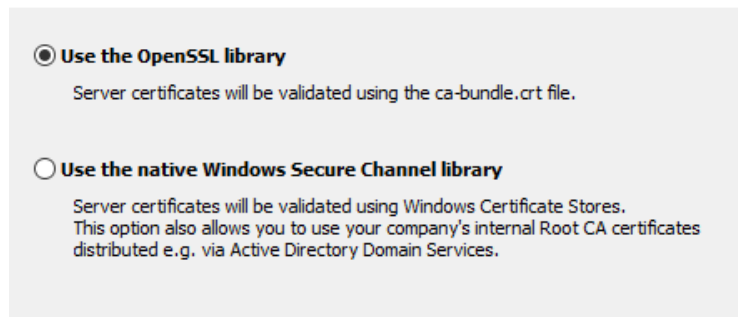
Luego nos aparecerá la opción de crear una carpeta en el menú de inicio, si no la queremos realizar solo tendremos que marcar la opción de no crearla y ya está.



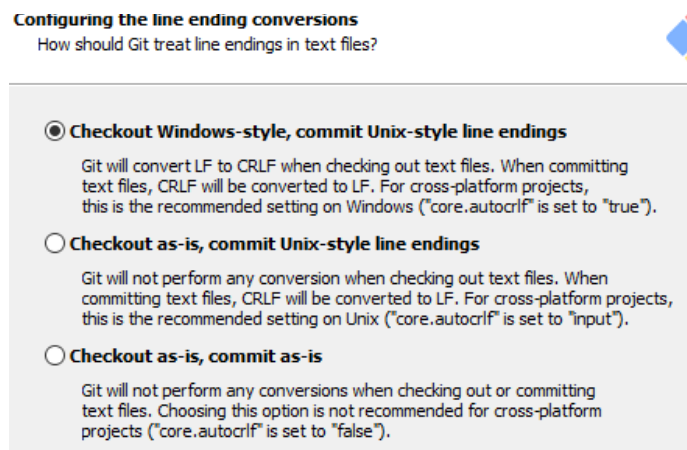
A continuación nos aparecerán las opciones para el uso del Git, en este caso dejaremos la segunda, que es la que aparece por defecto. Y nos permite usar git por consola de comandos.



A continuación nos aparece la opción de la instalación de las librerías, en este caso dejaremos la que viene por defecto.



Después nos aparecerá la configuración de Checkout, en este caso también dejaremos la que viene señalada por defecto.



A continuación seleccionamos la consola que queremos utilizar, dejaremos la del programa, que viene señalada por defecto.

### Configuring the terminal emulator to use with Git Bash

Which terminal emulator do you want to use with your Git Bash?



☒ **Use MinTTY (the default terminal of MSYS2)**

Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via 'winpty' to work in MinTTY.

☐ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

A continuación marcamos las opciones que vienen por defecto y comenzará la instalación el programa.

### Installing

Please wait while Setup installs Git on your computer.

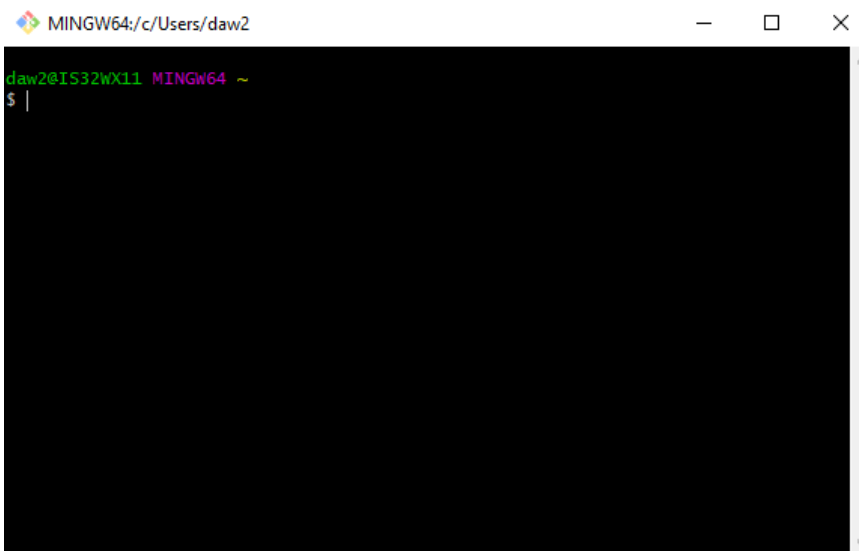


Extracting files...

C:\...\daw2\AppData\Local\Programs\Git\mingw64\lib\tcl8.6\tzdata\Australia\Victoria



Una vez acabada la instalación, se ejecutará el programa.



## Actividad 3.- Trabajando con Git

En primer lugar comprobamos que está instalado el git y la version:

```
operadorweb@PCB-USED:~$ git --version
git version 2.7.4
```

Vamos al directorio del usuario:

```
operadorweb@PCB-USED:~$ cd /var/www/html
```

Creamos el directorio del repositorio:

```
operadorweb@PCB-USED:~$ mkdir proyecto1
```

Accedemos al directorio y creamos el repositorio:

```
operadorweb@PCB-USED:~$ cd proyecto1/  
operadorweb@PCB-USED:~/proyecto1$ git init  
Initialized empty Git repository in /var/www/html/proyecto1/.git/
```

Hacemos un listado para ver los archivos del repositorio que están ocultos:

```
operadorweb@PCB-USED:~/proyecto1$ ls -a  
.  
..  
.git
```

Creamos dos archivos:

```
operadorweb@PCB-USED:~/proyecto1$ nano README.md
```

```
operadorweb@PCB-USED:~/proyecto1$ nano index.html
```

Comprobamos el status:

```
operadorweb@PCB-USED:~/proyecto1$ git status  
En la rama master  
  
Commit inicial  
  
Archivos sin seguimiento:  
  (use «git add <archivo>...» para incluir en lo que se ha de confirmar)  
  
    README.md  
    index.html
```

Cambiamos el archivo de zona: (a stage)

```
operadorweb@PCB-USED:~/proyecto1$ git add index.html  
operadorweb@PCB-USED:~/proyecto1$ git status  
En la rama master  
  
Commit inicial  
  
Cambios para hacer commit:  
  (use «git rm --cached <archivo>...» para sacar del stage)  
  
    nuevo archivo: index.html
```

Devolvemos el archivo a su zona: (a working)



```
operadorweb@PCB-USED:~/proyecto1$ git rm --cached index.html
rm 'index.html'
operadorweb@PCB-USED:~/proyecto1$ git status
En la rama master

Commit inicial

Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir en lo que se ha de confirmar)

      README.md
      index.html
```

Volvemos a mover el archivo y lo editamos modificamos el fichero: (a stage)

```
operadorweb@PCB-USED:~/proyecto1$ git status
En la rama master

Commit inicial

Cambios para hacer commit:
  (use «git rm --cached <archivo>...» para sacar del stage)

      nuevo archivo: index.html

Cambios no preparados para el commit:
  (use «git add <archivo>...» para actualizar lo que se confirmará)
  (use «git checkout -- <archivo>...» para descartar cambios en el directorio de trabajo)

      modificado:      index.html
```

Si queremos volver a moverlo y está modificado: (a working)

```
operadorweb@PCB-USED:~/proyecto1$ git rm --cached -f index.html
rm 'index.html'
```

Si queremos mover todos los archivos: (a stage)

```
operadorweb@PCB-USED:~/proyecto1$ git add .
operadorweb@PCB-USED:~/proyecto1$ git status
En la rama master

Commit inicial

Cambios para hacer commit:
  (use «git rm --cached <archivo>...» para sacar del stage)

      nuevo archivo: README.md
      nuevo archivo: index.html
```

Configuración de usuarios:

```
operadorweb@PCB-USED:~/proyecto1$ git config --global user.mail "pablo.cidbar@gmail.com"
operadorweb@PCB-USED:~/proyecto1$ git config --global user.name "DAW2PCB"
```

Realizar commits:

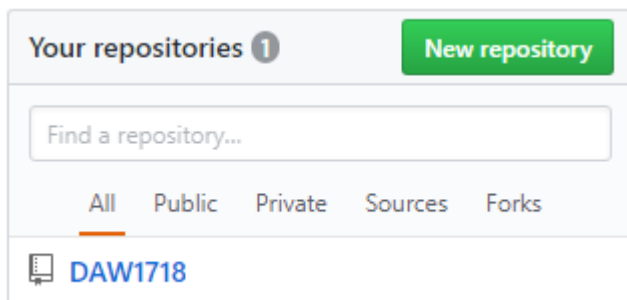
```
operadorweb@PCB-USED:~/proyecto1$ git commit -m "mi primer commit"
```

## Actividad 4.- Trabajando con ramas

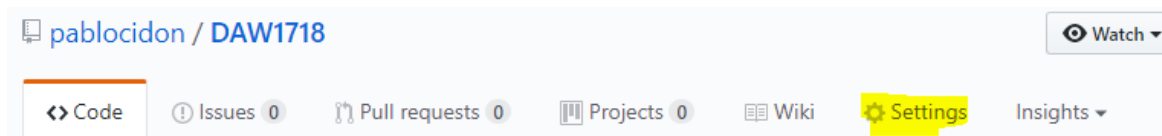
## Actividad 5.- Ramas y fusiones en local (FAST-Forward)

### a) Borrar el repositorio remoto DAW1718

En primer lugar vamos al repositorio que deseamos eliminar:

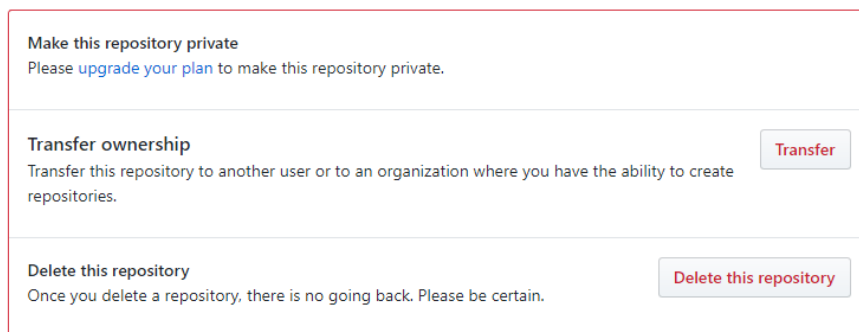


A continuación vamos a la pestaña settings:

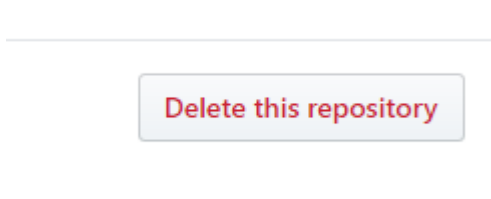


Una vez dentro, abajo del todo nos aparece la opción de eliminar el repositorio.

#### Danger Zone



Pulsamos en 'Delete this repository' y a continuación nos saldrá un diálogo de confirmación.



En el diálogo tendremos que introducir el nombre del repositorio y ya quedará eliminado.

Are you ABSOLUTELY sure?



Unexpected bad things will happen if you don't read this!

This action **CANNOT** be undone. This will permanently delete the **pablocidon/DAW1718** repository, wiki, issues, and comments, and remove all collaborator associations.


Please type in the name of the repository to confirm.

DAW1718

I understand the consequences, delete this repository

## b) Borrar el repositorio local DAW1718

Vamos al repositorio y con el botón derecho pinchamos en la opción de 'Eliminar'.

 DAW1718	03/10/2017 10:46	Carpeta de archivos
---	------------------	---------------------

## c) Crear de nuevo el proyecto HOLAMUNDO, con un commit con el archivo README.md que contiene el nombre de la asignatura.

Vamos a crear un nuevo repositorio para ello, vamos a 'New Repository', que se encuentra en la página principal de git

Your repositories 0 [New repository](#)


[All](#) [Public](#) [Private](#) [Sources](#) [Forks](#)

You don't have any repositories yet!

Una vez realizado, nos aparecerá una ventana en la que tendremos que introducir un nombre para el repositorio, una descripción y señalar la opción de crear el archivo README en caso de que querramos crear el archivo.

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner:  pablocidon / Repository name:

Great repository names are short and memorable. Need inspiration? How about [upgraded-spoon](#).

Description (optional):

☒ Public  
Anyone can see this repository. You choose who can commit.

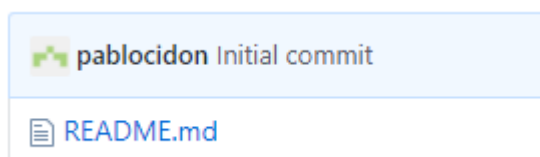
☐ Private  
You choose who can see and commit to this repository.

☒ Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

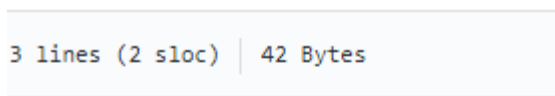
Add .gitignore:  Add a license:

[Create repository](#)

Automaticamente nos creará un commit con el archivo README



Si abrimos el archivo README, veremos que en él pone el nombre del repositorio y el nombre de la asignatura que ha sido puesta como descripción.

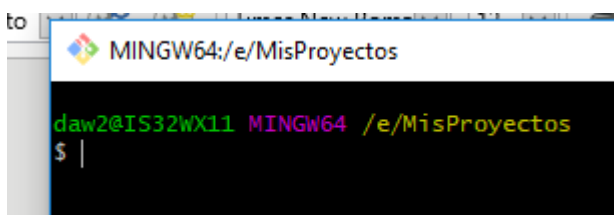


# HOLAMUNDO

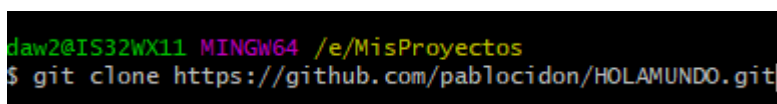
Despliege de Aplicaciones Web

## d) Clonación del proyecto en local

Una vez creado, vamos a la carpeta local donde vayamos a guardar el repositorio, y con el botón derecho seleccionamos la opción de abrir GIT Bash. Una vez hecho se abrirá la consola.



Para descargar el repositorio tendremos que introducir el siguiente comando



Listamos y comprobamos que se ha descargado

```
daw2@IS32WX11 MINGW64 /e/MisProyectos
$ ls
HOLAMUNDO/
```

Este equipo > DATOS (E:) > MisProyectos				
Nombre	Fecha de modifica...	Tipo	Tamaño	
HOLAMUNDO	04/10/2017 13:42	Carpeta de archivos		

### e) Crear una página web, index.html, con en mensaje “Hola Mundo” sin estilos. Una vez terminada, subela al repositorio remoto.

En primer lugar nos dirigiremos al directorio del repositorio con el que vamos a trabajar.

Nos aparecerá entre parentesis master, que indica en la rama en la que vamos a trabajar.

```
daw2@IS32WX11 MINGW64 /e/MisProyectos
$ cd HOLAMUNDO/

daw2@IS32WX11 MINGW64 /e/MisProyectos/HOLAMUNDO (master)
$ |
```

A continuación creamos el archivo y listamos para comprobar que está creado

```
daw2@IS32WX11 MINGW64 /e/MisProyectos/HOLAMUNDO (master)
$ touch index.html

daw2@IS32WX11 MINGW64 /e/MisProyectos/HOLAMUNDO (master)
$ ls
index.html  README.md
```

Ahora podemos abrir el documento y realizar en él todos los cambios que deseemos.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8">
    <title> Hola Mundo</title>
    <meta name="description" content="CSS"/>
    <meta name="author" content="Pablo"/>
  </head>
  <body>
    <h1>Hola Mundo</h1>
  </body>
</html>
```

Una vez realizados los cambios ya podremos realizar el commit.

En primer lugar cambiamos el archivo de area

```
daw2@IS32WX11 MINGW64 /e/MisProyectos/HOLAMUNDO (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        index.html
```

```
daw2@IS32WX11 MINGW64 /e/MisProyectos/HOLAMUNDO (master)
$ git add .
```

```
daw2@IS32WX11 MINGW64 /e/MisProyectos/HOLAMUNDO (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   index.html
```

**d) Añade la etiqueta v1.0 al commit anterior.**

```
daw2@IS32WX11 MINGW64 /e/MisProyectos/HOLAMUNDO (master)
$ git tag -a v1.0 -m 'Version 1.0'
```

**e) Se piden crear dos diseños con estilos(css/estilo.css) diferentes del index.html, para ello realizamos dos ramas en local, develop1 y develop2.**

```
daw2@IS32WX11 MINGW64 /e/MisProyectos/HOLAMUNDO (master)
$ git branch develop1

daw2@IS32WX11 MINGW64 /e/MisProyectos/HOLAMUNDO (master)
$ git branch develop2
```

**d) Una vez que el profesor decida que diseño es el más adecuado, se realizará la fusión en la master.**

```
daw2@IS32WX11 MINGW64 /e/MisProyectos/HOLAMUNDO (master)
$ git merge develop1
Already up-to-date.
```

**e) Se sube al servidor la rama master.**

```
daw2@IS32WX11 MINGW64 /e/MisProyectos/HOLAMUNDO (master)
$ git push origin master
```

**d) Se puede borrar las ramas develop1 y develop2.**

**g) Se etiqueta el nuevo commit, como v1.1.**