

DESPLIEGUE -IMPLANTACIÓN DE APLICACIONES WEB / ENTORNOS DE DESARROLLO/ LENGUAJE DE MARCAS



Profesor: Raúl Rodríguez

Github Pages como Hosting Web

Contenido

Contenido	1
Presentación.....	2
GitHub como repositorio web.....	3

Presentación

GitHub es un sitio "social coding" que permite subir repositorios de código para almacenarlo en el **sistema de control de versiones** Git.

A través de estas herramientas, podemos colaborar en proyectos de código, donde el sistema será de código abierto por defecto, lo que significa que cualquiera en el mundo podría encontrar tu código en GitHub, usarlo, aprender de este, y mejorarlo.

Esta acción es recíproca, es decir, que nosotros podemos hacer eso con el código de otras personas también.

En los siguientes apartados, aprenderemos a utilizar el repositorio social de código GitHub para utilizarlo como repositorio web de nuestros proyectos. Para ello, utilizaremos una de sus características disponibles como es **gh-pages** de GitHub.

Primeros Pasos con Git y Github

Lo primero que tenemos que tener en cuenta para poder utilizar GitHub, es habernos registrado previamente en <https://github.com/> y crearnos una cuenta.

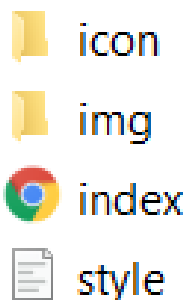
Además de esto, es necesario tener instalado en nuestro equipo las aplicaciones de GIT y GitHub para trabajar con el repositorio de versiones.



GitHub como repositorio web

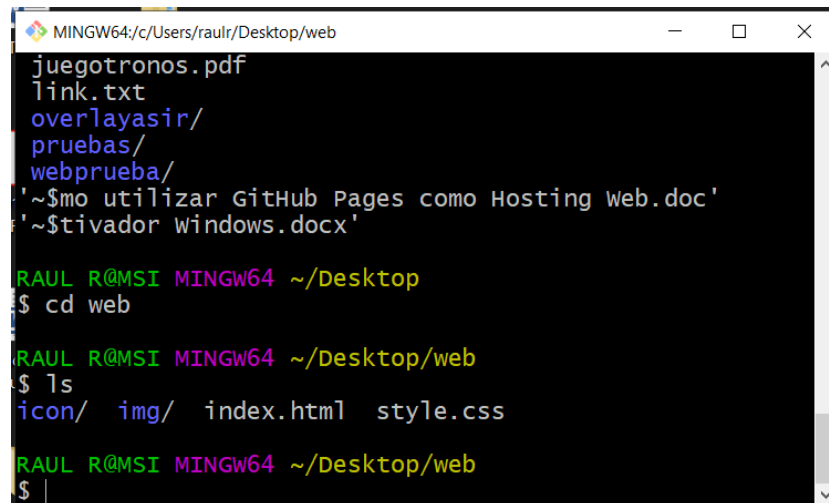
Una vez creada nuestra cuenta en GitHub e instalado todas las herramientas Git y GitHub en nuestro equipo de manera local, procederemos a la localización del proyecto que queremos “subir” al repositorio.

Recordar, que la página principal de nuestra web se tiene que llamar index.html y que las hojas de estilo (CSS), imágenes, etc. tienen que ir colocadas en distintas carpetas:



Despliegue - Implantación de Aplicaciones Web / Entornos de Desarrollo / Lenguaje de Marcas

Una vez localizado el proyecto, procedemos a abrir el terminal de la consola (Git Bash), y nos vamos a la carpeta donde está localizada el proyecto:



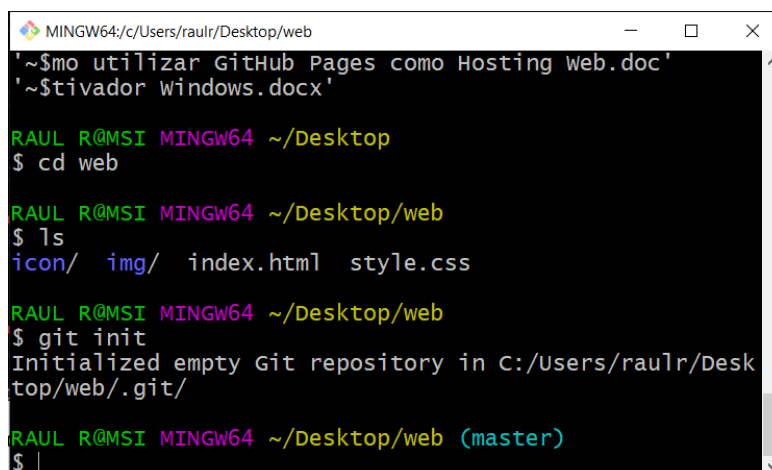
```
MINGW64/c/Users/raulr/Desktop/web
juegotronos.pdf
link.txt
overlayasir/
pruebas/
webprueba/
'$mo utilizar GitHub Pages como Hosting web.doc'
'$tivador Windows.docx'

RAUL R@MSI MINGW64 ~/Desktop
$ cd web

RAUL R@MSI MINGW64 ~/Desktop/web
$ ls
icon/  img/  index.html  style.css

RAUL R@MSI MINGW64 ~/Desktop/web
$
```

Una vez localizado nos metemos dentro de la carpeta del proyecto (web en este caso), y lo inicializamos:



```
MINGW64/c/Users/raulr/Desktop/web
'$mo utilizar GitHub Pages como Hosting web.doc'
'$tivador Windows.docx'

RAUL R@MSI MINGW64 ~/Desktop
$ cd web

RAUL R@MSI MINGW64 ~/Desktop/web
$ ls
icon/  img/  index.html  style.css

RAUL R@MSI MINGW64 ~/Desktop/web
$ git init
Initialized empty Git repository in C:/Users/raulr/Desktop/web/.git/

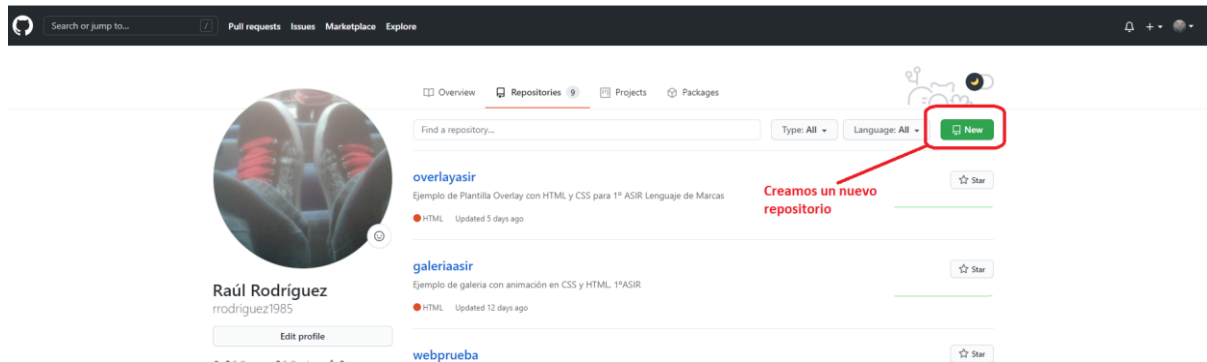
RAUL R@MSI MINGW64 ~/Desktop/web (master)
$
```

A continuación, definimos donde vamos a subir nuestro proyecto en el repositorio remoto.

Despliegue - Implantación de Aplicaciones Web / Entornos de Desarrollo / Lenguaje de Marcas

Para ello, realizamos los siguientes pasos:

Entramos en GitHub y creamos un nuevo repositorio:



El nombre del repositorio es necesario que sea el mismo que el nombre del proyecto:

Create a new repository

Owner * / Repository name *

/

Great repository names are short and unique. e. Need inspiration? How about [potential-spoon?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

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

Initialize this repository with:
Skip this step if you're importing an existing repository.

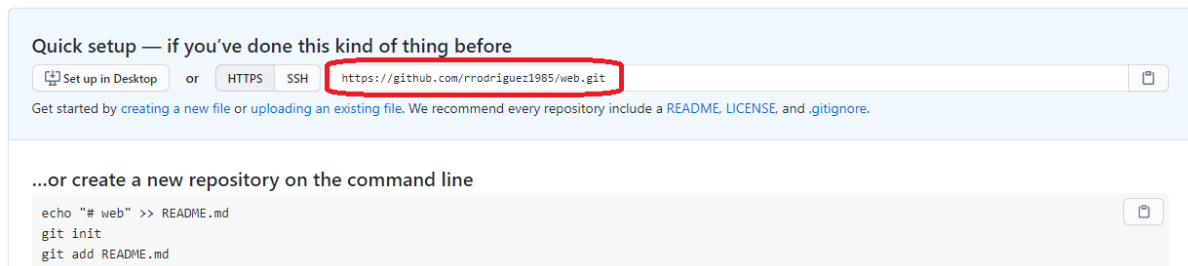
☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

[Create repository](#)

Y copiamos la dirección que nos facilitan del repositorio:



El siguiente paso es indicar en el terminal donde queremos subir nuestro proyecto, en qué repositorio.

Para ello, escribimos lo siguiente:

git remote add origin Estructura: ***git remote add origin*** + la dirección de nuestro repositorio

```
RAUL R@MSI MINGW64 ~/Desktop/web (master)
$ git remote add origin https://github.com/rrodriguez1985/web.git
```

Pulsamos intro y ya está definido el remoto.

Para que el github pages funcione y lo podamos utilizar como servidor, es necesario que esté situado en una rama llamada -gh pages.

Introducimos en el terminal ***git checkout -b gh-pages*** y creamos la rama.

```
RAUL R@MSI MINGW64 ~/Desktop/web (master)
$ git checkout -b gh-pages
Switched to a new branch 'gh-pages'

RAUL R@MSI MINGW64 ~/Desktop/web (gh-pages)
$ git status
On branch gh-pages

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    icon/
    img/
    index.html
    style.css

nothing added to commit but untracked files present (use "git add" to track)

RAUL R@MSI MINGW64 ~/Desktop/web (gh-pages)
$ |
```

Añadimos los archivos y hacemos un commit:

```
RAUL R@MSI MINGW64 ~/Desktop/web (gh-pages)
$ git add .

RAUL R@MSI MINGW64 ~/Desktop/web (gh-pages)
$ git commit -m 'Initial Commit'
[gh-pages (root-commit) 8ad577d] Initial Commit
12 files changed, 187 insertions(+)
create mode 100644 icon/favicon.png
create mode 100644 img/foto1.jpg
create mode 100644 img/foto2.jpg
create mode 100644 img/foto3.jpg
create mode 100644 img/foto4.jpg
create mode 100644 img/foto5.jpg
create mode 100644 img/foto6.jpg
create mode 100644 img/foto7.jpg
create mode 100644 img/foto8.jpg
create mode 100644 img/foto9.jpg
create mode 100644 index.html
create mode 100644 style.css

RAUL R@MSI MINGW64 ~/Desktop/web (gh-pages)
$ |
```

Nota: Si nos da un error al hacer commit, es debido a que tenemos que hacer la configuración inicial de nuestro git bash. Para ello, tenemos que poner los siguientes comandos:

```
$ git config --global user.name <nombre>
```

```
$ git config --global user.surname <apellido>
```

```
$ git config --global user.email <email>
```

```
$ git config --list → Comprobamos que se ha configurado correctamente
```

Sustituir el texto entre < > con tus datos, pero no pongas dichos símbolos en la instrucción.

Posteriormente lo subimos a la rama:

```
RAUL R@MSI MINGW64 ~/Desktop/web (gh-pages)
$ git push origin gh-pages|
```

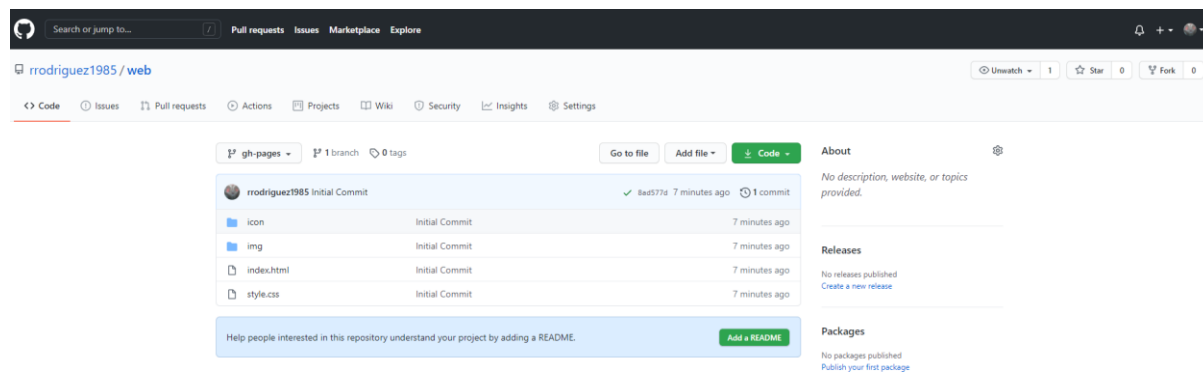
Si no hemos inicializado sesión en GitHub, nos pedirá inicializar sesión con nuestro usuario y contraseña, y empezará a subirse los archivos.

```
RAUL R@MSI MINGW64 ~/Desktop/web (gh-pages)
$ git push origin gh-pages
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 12 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (16/16), 1.54 MiB | 936.00 KiB/s, done.
Total 16 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/rrodriguez1985/web.git
 * [new branch]      gh-pages -> gh-pages

RAUL R@MSI MINGW64 ~/Desktop/web (gh-pages)
$ |
```


Despliegue - Implantación de Aplicaciones Web / Entornos de Desarrollo / Lenguaje de Marcas

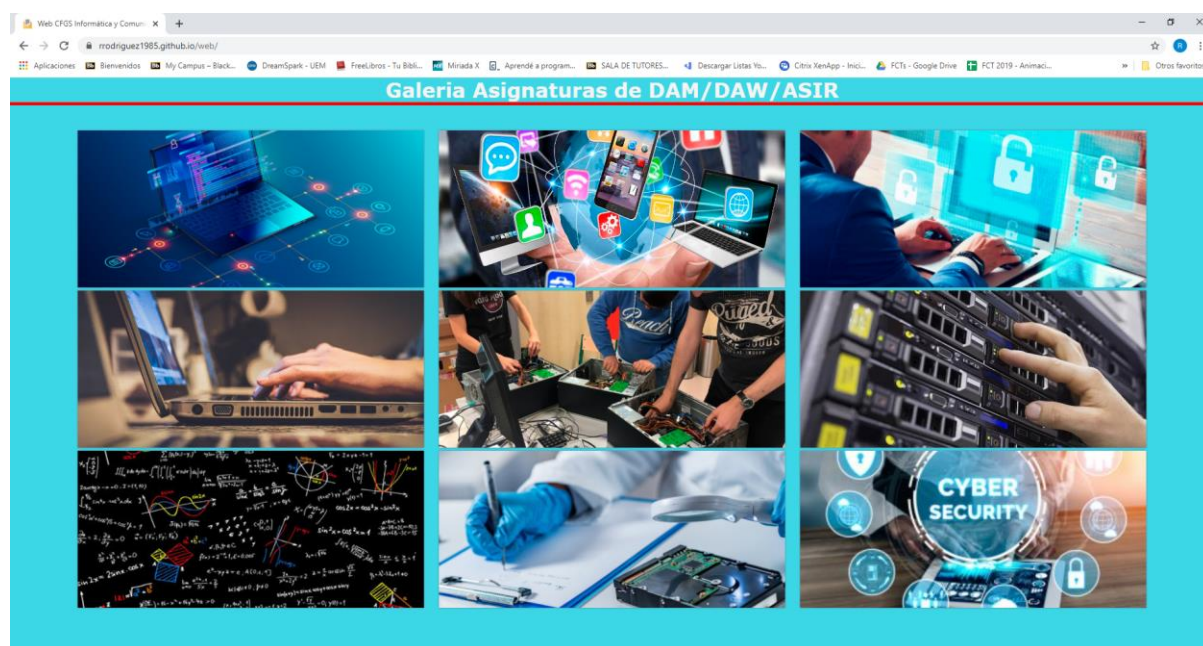
Si ahora vamos a github veremos que está el repositorio con el proyecto actualizado y subido, listo para ser utilizado.



Para poder ver nuestra web, tenemos que poner el siguiente link:

nuestro nombrederepositorio.github.io/nombredelacarpetadondeestáelproyecto

<https://rrodriguez1985.github.io/web>



Si queremos realizar más cambios en nuestro sitio web de prueba y cargarlos en GitHub, simplemente tendremos que realizar el cambio en los archivos como antes.

Posteriormente, actualizaremos estos cambios en nuestro repositorio de la siguiente manera:

git add . → Para añadir todo en nuestro repositorio.

git commit -m 'another commit' → Para hacer un nuevo commit con los cambios que hemos hecho

git push origin gh-pages → Para actualizar estos cambios y subirlos a nuestra web.