

Tema 8

Git y Heroku

Git

Git es un sistema de control de versiones con funcionalidad plena. Se utiliza para manejar desde los más pequeños detalles hasta los más grandes proyectos con rapidez y eficiencia.

Se diseñó pensando en el mantenimiento de las versiones de las aplicaciones cuando éstas tenían un gran número de archivos de código fuente. Mediante Git, los desarrolladores podían subir versiones modificadas del proyecto sin sustituir todos los archivos del actual.

Git

Es una herramienta muy útil para que dos o más desarrolladores trabajen de forma simultánea, sin necesidad de ir intercambiando manualmente archivos modificados por otras personas.

Hay algunos proyectos de mucha relevancia que usan Git, como por ejemplo el grupo de programación del núcleo Linux. Mantiene una enorme cantidad de código distribuida y gestionada por mucha gente.

Git

Sus principales características son:

- Buena gestión de ramas y mezclado de diferentes versiones.
- Gestión distribuida. Cada programador tiene una copia local del historial del desarrollo entero. Los cambios se propagan entre los repositorios locales.
- Gestión eficiente de grandes proyectos.
- Los almacenes de información pueden publicarse por HTTP mediante una conexión TCP/IP o a través de cifrado **SSH**.

Git

Git se puede descargar de su página oficial para el sistema operativo que se desee. El enlace para su descarga es: <https://git-scm.com/downloads>.


Para alojar nuestros archivos utilizaremos [GitHub](#). Para poder utilizarlo deberás crear una cuenta en la siguiente dirección: <https://github.com/>. Una vez creada la cuenta, procederemos a utilizar Git en nuestra aplicación.

Git

En la página principal pulsaremos sobre *Start a project*. Crearemos un nuevo repositorio rellenando los campos de la siguiente manera:

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


Owner **Repository name**


 Javierjim2 ▾ / HelloWorld ✓

Great repository names are short and memorable. Need inspiration? How about **automatic-doodle**.

Description (optional)

Mi primera aplicación para Git

☒  **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: **None** ▾ | Add a license: **None** ▾ ⓘ

Create repository

Git

Una vez creado el repositorio, veremos que únicamente tendremos el archivo **README.md**. Para empezar a utilizar Git, crearemos una carpeta que servirá para almacenar los archivos. Situándonos en esa carpeta, ejecutaremos el siguiente comando en la línea de órdenes:

```
git clone https://github.com/USUARIO/NOMBRE_DEL_PROYECTO.git
```

De esta manera habremos clonado los archivos del repositorio en nuestra carpeta.

Git

Ahora que tenemos nuestro proyecto en el ordenador, podemos añadir los archivos que queramos a esa carpeta. Por ejemplo, podemos utilizar una aplicación que hayamos desarrollado en temas anteriores para subirla a GitHub.

Una vez añadidos los archivos, hay que registrar los cambios del repositorio mediante la orden “**\$ git add .**”. Ahora estos cambios estarán almacenados en un área temporal que Git llama *index*.

Git

Podemos comprobar el estado del repositorio mediante **\$ git status**. Esta orden nos mostrará todos los archivos que hayan sido creados o modificados respecto a la última versión del repositorio.

El siguiente paso será almacenar de forma permanente esos cambios en el repositorio. Para ello, hay que ejecutar la orden **\$ git commit -m "NombreDelCommit"**.

Git

Si todo ha ido correctamente, en la consola aparecerá una línea indicando que si queremos publicar nuestros *commits* locales, tendremos que hacer un ***push***.

Para subir los archivos a GitHub, deberemos hacer un *push* mediante la orden **\$ git push origin master**. Si abrimos GitHub en el navegador y buscamos el repositorio, veremos que se han subido correctamente los nuevos archivos.

Git

La manera de descargar los archivos del repositorio a tu carpeta es mediante la orden **\$ git pull**. En caso de que nos aparezca un texto pidiendo que escribamos un mensaje, habrá que hacer:

- Pulsar la tecla “i”.
- Escribir el mensaje deseado para explicar la unión de archivos.
- Pulsar “esc”.
- Escribir “:wq”.
- Apretar “enter”.

Git

Si a la hora de realizar el *pull* hay algún conflicto con uno o más archivos, podemos proceder de dos maneras:

- Eliminar los cambios locales sobre esos archivos y quedarnos con los que se han descargado del repositorio. Para ello se ejecuta:

1. **\$ git reset — —hard**

2. **\$ git pull**

Git

Fusionar los archivos en conflicto de manera que se queden tanto los cambios locales como los descargados. Por ejemplo, en un archivo .txt tendríamos el texto que hemos escrito localmente y el texto nuevo que hemos descargado.

1. **\$ git stash**
2. **\$ git pull**
3. **\$ git stash pop**

Git

Algunos otros comandos útiles de Git son:

- `git reset —hard HEAD`: Elimina los cambios que aún no se hayan hecho *commit*.
- `git branch`: Lista todas las ramas locales.
- `git branch -a`: Lista todas las ramas locales y remotas.
- `git revert <hash_commit>`: Revierte el commit realizado, identificado por “hash_commit”.

Heroku

Heroku es una plataforma como servicio de computación en la nube que soporta distintos lenguajes de programación. De forma parecida a GitHub, almacena repositorios en la nube que, además, pueden ser ejecutados directamente desde allí.

Heroku sabe el tipo de aplicación que el usuario desarrolla y soporta la mayoría de los frameworks más modernos mediante lo que se denominan *buildpacks*. Únicamente haciendo un push, Heroku sabrá qué tipo de aplicación es y cómo ejecutarla, sin instalar ningún software dependiente.

Heroku

Heroku se puede encontrar en la categoría de los PaaS (plataforma como servicio), ya que el servicio que ofrecen es su plataforma o hardware.

Sin embargo, Heroku es mucho más que un VPS o un computador en la nube que puedes gestionar como cualquier otro. Este se enfoca en ser una plataforma de aplicaciones. Maneja cada una de tus aplicaciones por separado de manera encapsulada.

Heroku

Las unidades que proveen capacidad de cómputo dentro de Heroku se denominan **Dynos**. Cada Dyno está aislado del resto, de manera que los comandos se ejecutan y los archivos se almacenan en cada uno de ellos. Sus principales características son:

- **Elasticidad y crecimiento.** La cantidad de Dynos asignados a una aplicación se puede cambiar en cualquier momento.
- **Tamaño.** Hay distintos tipos de Dynos con diferentes capacidades de procesamiento y memoria.
- **Seguimiento.** El manejador de Dynos monitorea de forma continua los dynos en ejecución.

Heroku

- **Routing.** Internamente los routers realizan un seguimiento de la ubicación de los Dynos que estén corriendo y redirigen el tráfico de acuerdo a la misma.
- **Distribución y redundancia.** Los dynos se encuentran aislados uno de otro. Esto implica que de existir fallos en la infraestructura interna de alguno de ellos, los demás no se ven afectados, ni tampoco la aplicación.

Heroku

Utiliza git para crear los repositorios y actualizarlos. Para instalar Heroku, podemos descargarlo de desde su página web para nuestro sistema operativo:

<https://devcenter.heroku.com/articles/heroku-cli#download-and-install>

Tendremos que crear una cuenta, igual que lo hicimos con GitHub. Para ello entramos a <https://www.heroku.com/> y le damos a **Sign up**. Una vez creada la cuenta, podemos empezar a utilizar Heroku con nuestra aplicación React.

Heroku

Podemos crear nuestra aplicación React y subirla a Heroku de manera muy sencilla. Sólo necesitaremos ejecutar en la consola estos comandos:

```
npm install -g create-react-app
create-react-app my-app
cd my-app
git init
heroku create -b https://github.com/mars/
create-react-app-buildpack.git
git add .
git commit -m "react-create-app on Heroku"
git push heroku master
heroku open
```

De esta manera, tendremos nuestro repositorio en Heroku y podremos ejecutar la aplicación directamente desde allí.