

Solución subir A Heroku

Jairo Rastrojo Rodríguez

Administración de sistemas gestores de bases de datos

1. Instalar heroku cli

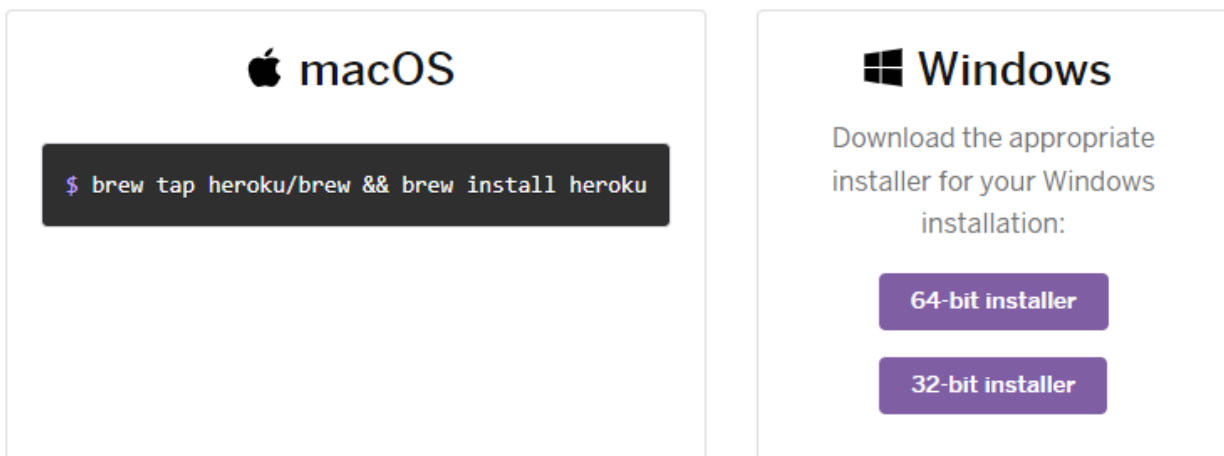
Suponemos que a estas alturas tienes git instalado en tu proyecto, por si acaso en este link encontrarás información de la instalación:

https://github.com/rastrojorodriguezjairo2/Primer_Proyecto/blob/main/doc/practica_git_github_jairo.pdf

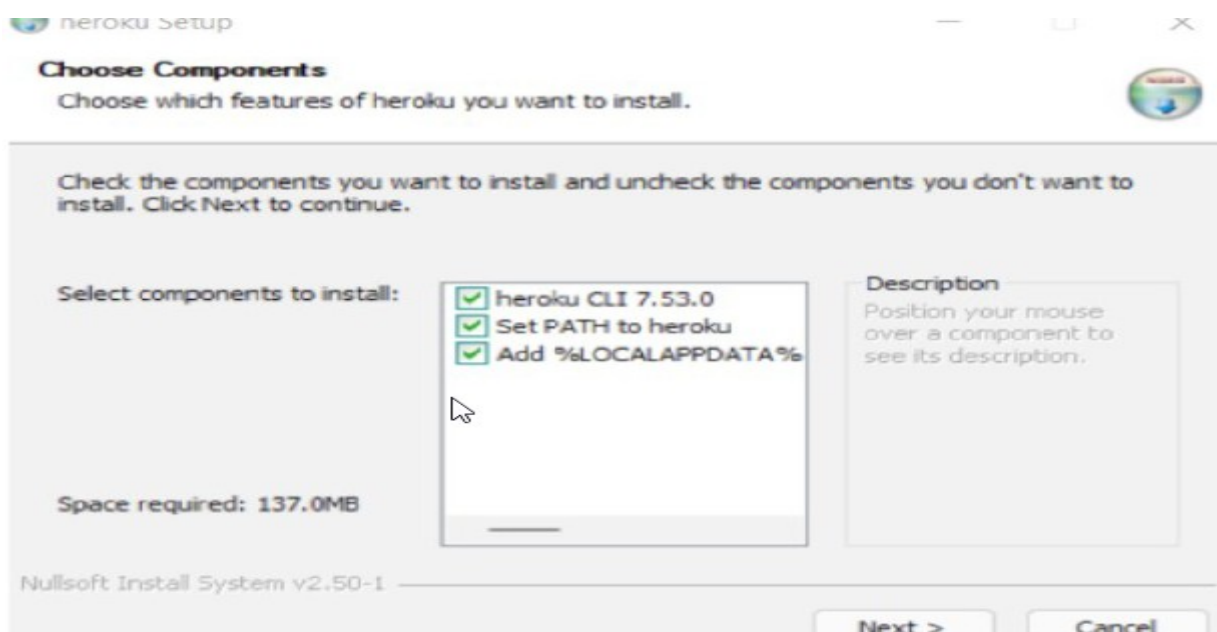
Para instalarlo, primero descarguemos heroku cli, (suponiendo que no lo tengas ya instalado)

<https://devcenter.heroku.com/articles/heroku-cli>

En el enlace, bajamos hasta ver la siguiente imagen y descargamos la versión que deseamos.



Ejecutamos el instalador, y lo instalamos.



Para comprobar que se ha instalado correctamente abrimos un terminal y usamos el comando “heroku –help”, si se muestra la ayuda es que se ha instalado correctamente y podemos proseguir.

```
C:\WINDOWS\system32>heroku --help
» Warning: Our terms of service have changed: https://dashboard.heroku.com/terms-of-service
CLI to interact with Heroku

VERSION
  heroku/7.53.0 win32-x64 node-v12.21.0

USAGE
  $ heroku [COMMAND]

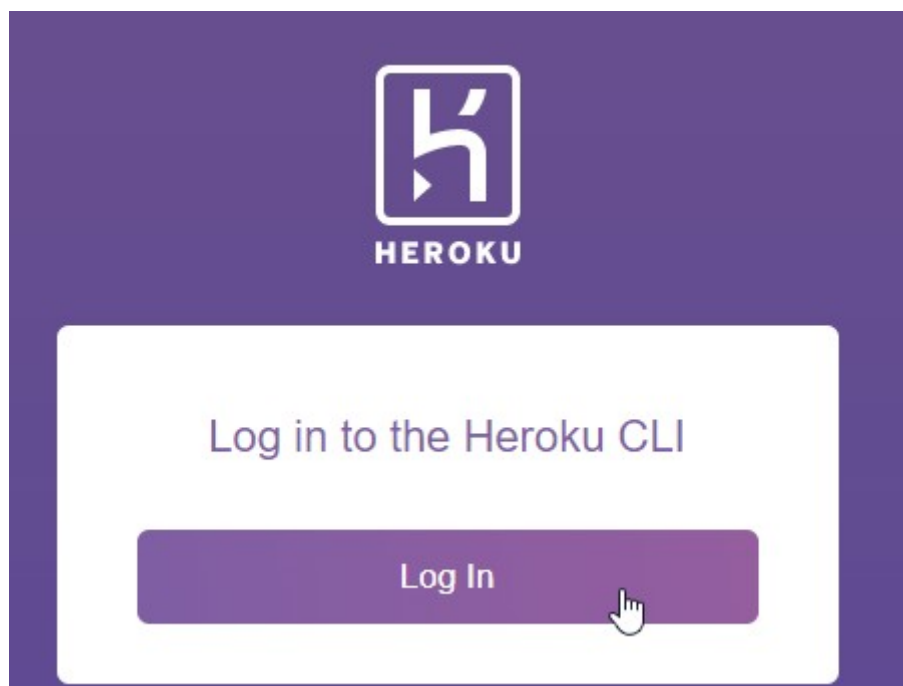
COMMANDS
  access          manage user access to apps
  addons          tools and services for developing, extending, and operating your app
  apps            manage apps on Heroku
  auth            check 2fa status
  authorizations  OAuth authorizations
  autocomplete    display autocomplete installation instructions
  buildpacks      scripts used to compile apps
  certs           a topic for the ssl plugin
  ci              run an application test suite on Heroku
  clients         OAuth clients on the platform
  config          environment variables of apps
  container       Use containers to build and deploy Heroku apps
  domains         custom domains for apps
  drains          forward logs to syslog or HTTPS
  features        add/remove app features
  git             manage local git repository for app
  help           display help for heroku
  keys            add/remove account ssh keys
  labs            add/remove experimental features
```

2. Conexión con heroku

Ahora vamos a ejecutar el comando `git init` para que se cree la carpeta `.git` y poder usar sus comandos en el proyecto, comandos que usaremos para subir el proyecto, usamos el comando “`heroku login`” para poder acceder a nuestra cuenta de heroku

```
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDL\RES_API_HOSPITAL> git init
Reinitialized existing Git repository in C:/Jairo/Estudios Jairo/3_GRADO_SUPERIOR/2º/GBDDL/RES_API_HOSPITAL/.git/
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDL\RES_API_HOSPITAL> heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/18f7f324-c2a6-4bec-8dac-ce636de0ea48?requestor=SFMyNTY.g2gDbQAAAA8xOTQuMjIwLjIyNC4xNDJuBgCQklrTgAFiAAFRgA.4Xg0aV-fVq0FrXg_QteCAMDVgwBYrz_G3Hs3cFoD2Q8
Logging in... done
Logged in as jaiorasrod@gmail.com
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDL\RES_API_HOSPITAL> |
```

En el navegador vemos que se abre una ventana para iniciar sesión en heroku



A continuación usamos el comando “heroku git:remote -a prueba056” (prueba056 es el nombre del proyecto que se ha puesto en heroku) debido a este comando, estamos determinando donde se va a subir nuestro proyecto, usamos “git add .” para poner en espera el proyecto a su subida,

```
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDL\RES_API_HOSPITAL> heroku git:remote -a prueba056
set git remote heroku to https://git.heroku.com/prueba056.git
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDL\RES_API_HOSPITAL> git add .
warning: LF will be replaced by CRLF in package-lock.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory
```

Ahora usamos “git commit -am ‘version1’” para diferenciar las versiones que vallamos subiendo.

```
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDL\RES_API_HOSPITAL> git commit -am "version1"
[main 50e33d6] version1
3 files changed, 17 insertions(+), 1 deletion(-)
```

ATENCIÓN si consultáis algunas guías en internet tened en cuenta que para subir los datos a heroku os mostrarán este comando “git push heroku master”, este comando puede fallar como a mí:

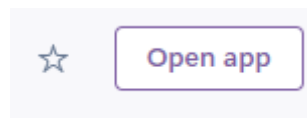
```
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDL\RES_API_HOSPITAL> git push heroku master
error: src refspec master does not match any
error: failed to push some refs to 'https://git.heroku.com/prueba056.git'
```

El comando que ha funcionado en este caso es “git push heroku”

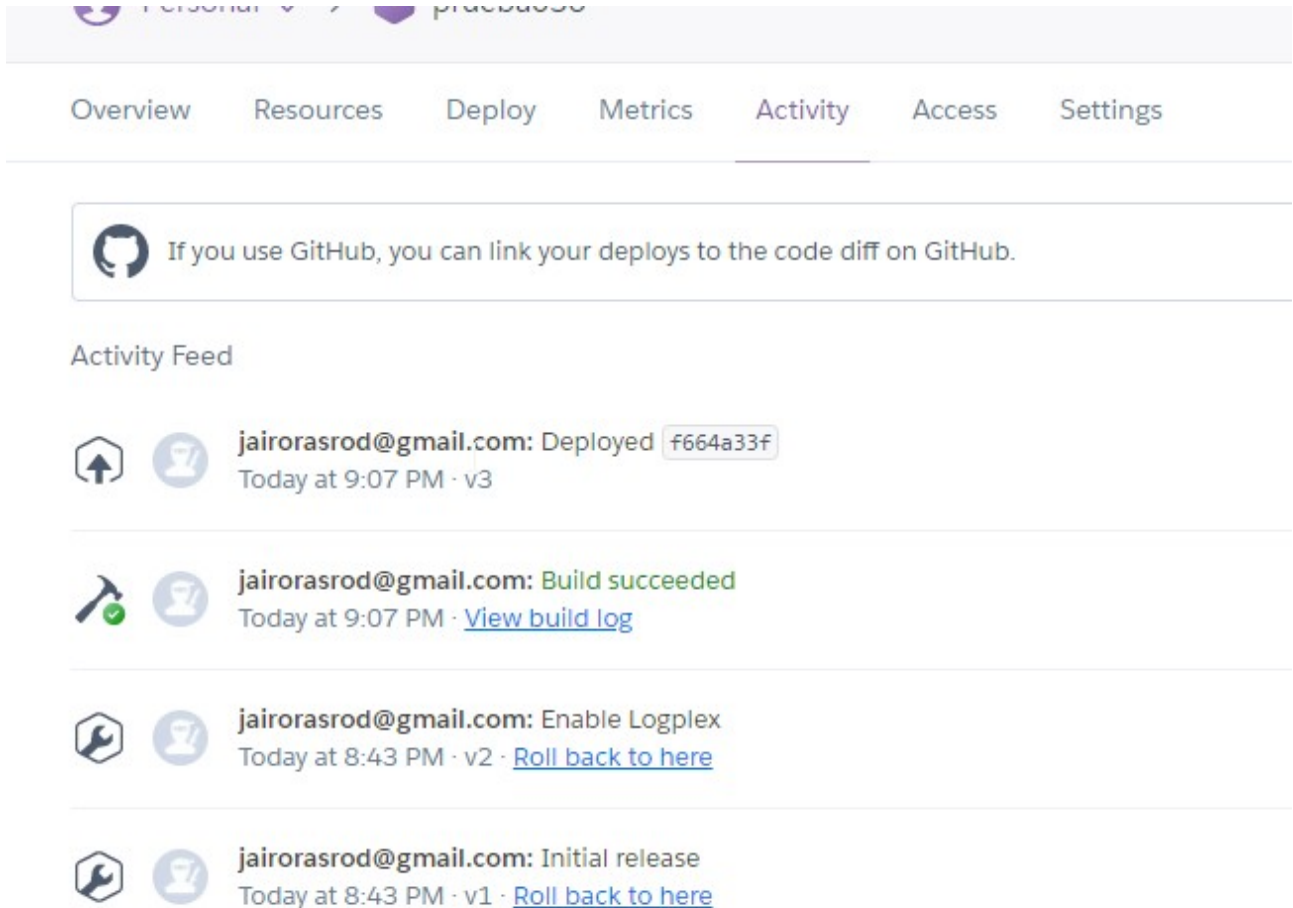
```
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDL\RES_API_HOSPITAL> git push heroku
Enumerating objects: 222, done.
Counting objects: 100% (222/222), done.
Delta compression using up to 4 threads
Compressing objects: 100% (181/181), done.
Writing objects: 100% (222/222), 1006.52 KiB | 919.00 KiB/s, done.
Total 222 (delta 84), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Node.js app detected
remote:
```

```
remote:      https://prueba056.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/prueba056.git
* [new branch]      main -> main
```

Podemos comprobar abriendo la app que ya funciona y se ha subido



Pero también consultando la actividad que ha habido en heroku



Si queremos subir una nueva versión, simplemente tenemos que repetir los pasos desde el 'git add .' teniendo en cuenta que tenemos que logearnos de nuevo si hemos cerrado el proyecto en cuestión.

3. ¿En caso de qué?

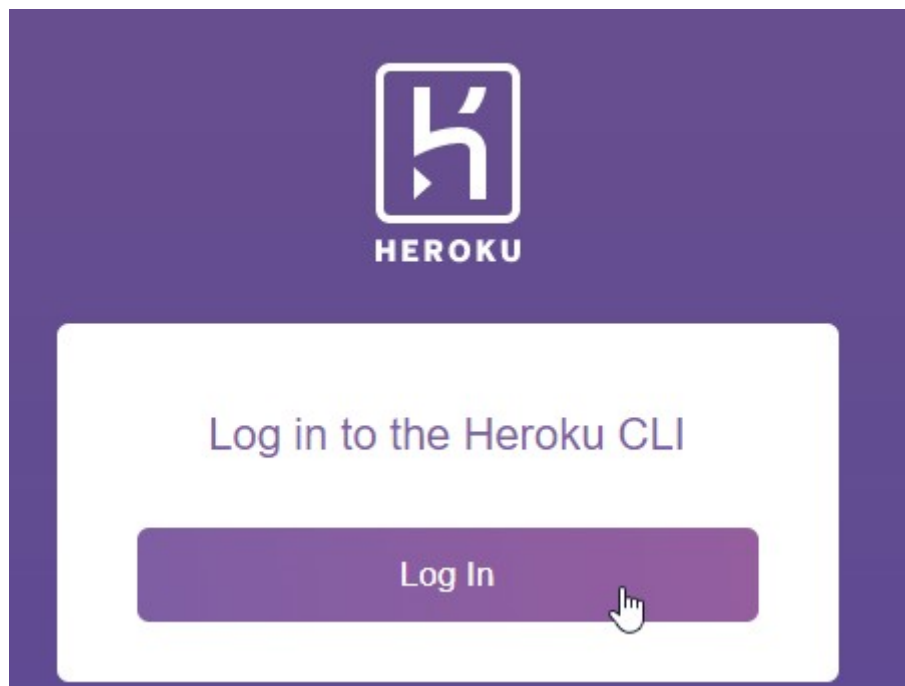
En caso de que no hayáis creado un lugar en heroku para el proyecto simplemente seguid los siguientes pasos:

Primero al igual que si estuviese creado usamos “git add .” y hacemos el “git commit -am ‘version2’”

```
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDLC\RES_API_HOSPITAL> git add .
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDLC\RES_API_HOSPITAL> git commit -am "version2"
[main 52948a9] version2
1 file changed, 1 insertion(+)
```

De igual forma nos logeamos en heroku

```
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDLC\RES_API_HOSPITAL> heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/18f7f324-c2a6-4bec-8dac-ce636de0ea48?requestor=SFMyNTY.g2gDbQAAAA8xOTQuMjIwLjIyNC4xNDJuBgCQklrTgAFiAAFRgA.4Xg0aV-fVq0FrXg_QteCANDVgwBYrz_G3Hs3cFoD2Q8
Logging in... done
Logged in as jaiorasrod@gmail.com
```



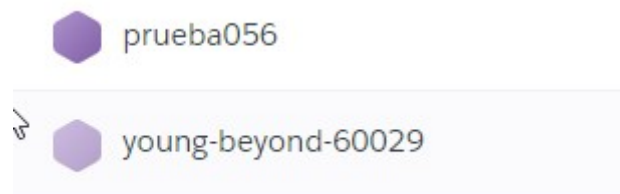
Y usamos el comando “heroku create”, lo cual creará en heroku un nuevo lugar para nuestro proyecto, pero generado de forma automática, pero tranquilos, por que podréis cambiarle el nombre y demás opciones

```
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDLC\RES_API_HOSPITAL> heroku create
Creating app... done, ● young-beyond-60029
https://young-beyond-60029.herokuapp.com/ | https://git.heroku.com/young-beyond-60029.git
```

Usamos el comando “git push heroku” y se subirá nuestro proyecto

```
PS C:\Jairo\Estudios Jairo\3_GRADO_SUPERIOR\2º\GBDDL\RES_API_HOSPITAL> git push heroku
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 547 bytes | 273.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
```

Podemos ver nuestro proyectos



Y también consultando la actividad vemos que se ha subido

