



## Cómo hacer un deploy

### Objetivo

Siguiendo estos pasos vamos a poder subir **nuestro proyecto de Mercado Liebre** a **Heroku** y de esta forma el mismo quedará disponible para que todos lo puedan ver.

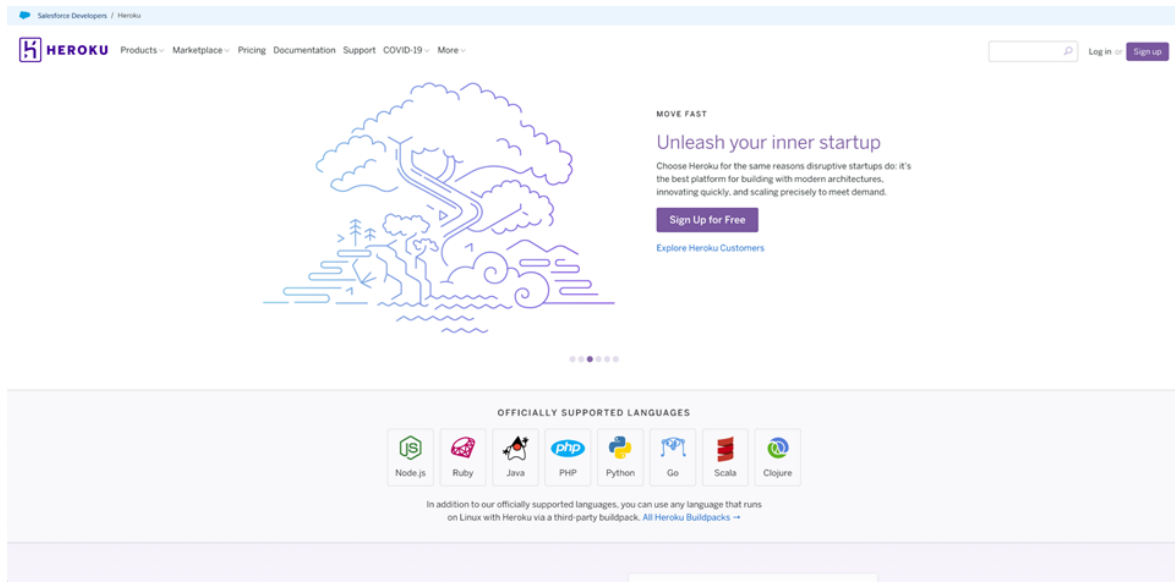
### Antes de empezar...

Antes de hacer cualquier cosa, asegurémonos de tener listos estos **4 puntos**:

1. Nuestro proyecto de Mercado Liebre, listo para subir a [Heroku](#).
2. Una cuenta en [Heroku](#).\*
3. **Puertos** configurados.\*
4. Archivo **procfile**.\*

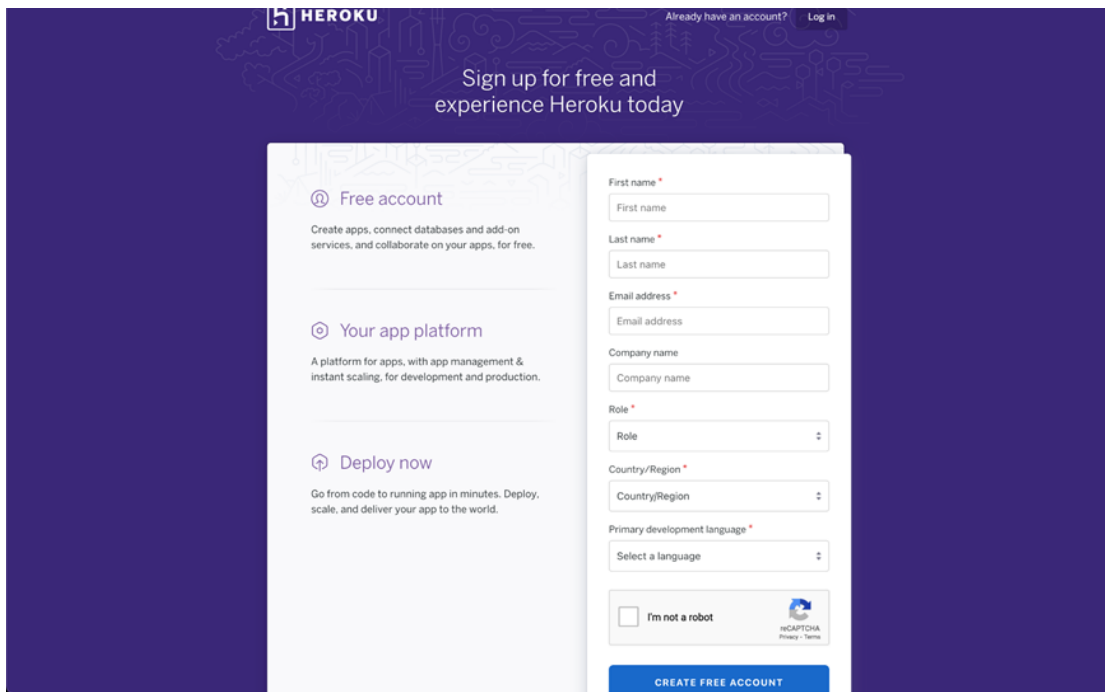
## \* Una cuenta en Heroku

Cuando ingresamos al siguiente enlace: <https://www.heroku.com/> estaríamos accediendo a la **Heroku**, que es una plataforma que ofrece un servicio de computación en la Nube que soporta distintos lenguajes de programación.



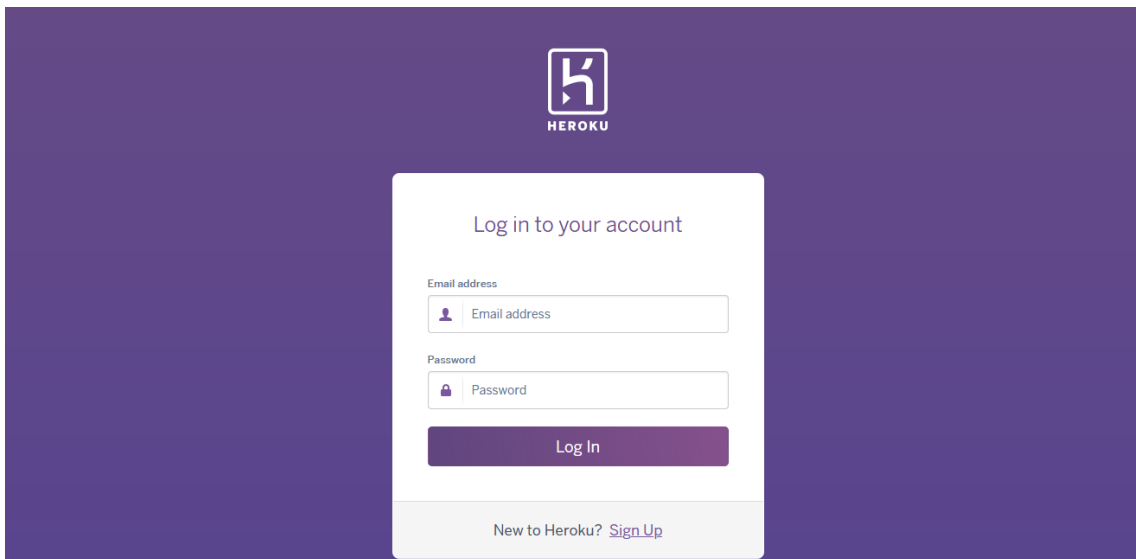
En la parte superior derecha debemos accionar clic, sobre el botón: (Inscribirse) y allí se apertura el formulario para que coloques tus datos:

Sign up



The image shows the Heroku sign-up page. At the top, there's a navigation bar with the Heroku logo, a link "Already have an account? Log in", and a "Sign up for free and experience Heroku today" headline. The main content area is divided into two columns. The left column contains three sections: "Free account" (with a sub-header "Create apps, connect databases and add-on services, and collaborate on your apps, for free."), "Your app platform" (with a sub-header "A platform for apps, with app management & instant scaling, for development and production."), and "Deploy now" (with a sub-header "Go from code to running app in minutes. Deploy, scale, and deliver your app to the world."). The right column contains a registration form with fields for "First name", "Last name", "Email address", "Company name", "Role" (a dropdown menu), "Country/Region" (a dropdown menu), and "Primary development language" (a dropdown menu). Below these fields is a checkbox for "I'm not a robot" and a CAPTCHA image. At the bottom of the form is a blue button labeled "CREATE FREE ACCOUNT".

Una vez que cree su cuenta, debe iniciar sesión en la plataforma, accionando clic sobre el botón, **Log in** y colocando los datos exigidos:



The image shows the Heroku login page. At the top, there's a navigation bar with the Heroku logo. The main content area is a white card with a purple background. The card has a heading "Log in to your account". Below the heading are two input fields: "Email address" and "Password". Each field has a small icon (a person for email and a lock for password) to its left. Below the input fields is a purple button labeled "Log In". At the bottom of the card, there's a link "New to Heroku? Sign Up".

## \* Configuración del puerto

Si utilizamos **express-generator** para crear nuestro proyecto, es posible que no necesitemos seguir esta indicación. Si iniciamos nuestro proyecto instalando el **Framework Express** a través de **npm** y así las diferentes dependencias, probablemente tengamos que hacer un pequeño cambio dentro de la configuración del puerto donde va a correr nuestro proyecto.

Normalmente utilizamos el puerto **3000 ó 3001**, pero, a la hora de publicar el proyecto dentro de **Heroku**, esta plataforma establece otro diferente. En ese caso, vamos a cambiar el número de puerto dentro del **listen** a la siguiente línea:

### // CAMBIAR ESTE BLOQUE:

```
app.listen(3000, function() {  
  console.log(`Servidor corriendo en el puerto 3000`);  
})
```

### // POR ESTE BLOQUE

```
app.listen(process.env.PORT || 3000, function() {  
  console.log(`Servidor corriendo en el puerto 3000`);  
})
```

Heroku utiliza [variables de entorno](#) para guardar cierta información. Dentro de la variable llamada **PORT** estará guardado aquel dato que necesitamos. Entonces, ¿podríamos traducir esa línea que cambiamos al español? Claro que sí:

```
app.listen(**Si existe la variable PORT, usá ese dato** || **Sino directo  
3000**, function() {  
  console.log(`Servidor corriendo en el puerto 3000`);  
})
```

Aquí podemos observar todo el código para la configuración del puerto:

```
JS index.js ×
JS index.js > app.listen() callback
1  const express = require('express');
2  const path = require('path');
3  const app = express();
4  //MIDDLEWARE
5  app.use(express.static('public'));
6  app.set('puerto', process.env.PORT || 3001);
7  //----- RUTAS -----
8  app.get('/',function(req,res){
9  |   res.sendFile(path.resolve(__dirname, './views/home.html'))
10 | })
11 app.get('/registro', (req,res)=>{
12 |   res.sendFile(path.resolve(__dirname, './views/registro.html'));
13 | })
14 app.listen (app.get('puerto'), ()=> console.log(`Servidor corriendo de manera
15 |satisfactoria ${app.get('puerto')}` ));
```

## \* Procfile

Otra cosa que Heroku necesita es un **Procfile**. Esto le dice a Heroku qué comandos ejecutar para iniciar nuestro sitio web. Para ello, hace falta crear un archivo llamado **Procfile** (sin extensión) en la **raíz del proyecto** y guardar dentro la siguiente línea:

- web: node rutaAlArchivoPrincipal.js

```
web: node app.js
web: node ./src/app.js
```

Esta línea significa que vamos a desplegar una aplicación **web** ejecutando el comando **gunicorn mysite.wsgi** (**Gunicorn** es un programa que es la versión más poderosa del comando de Django **runserver**).

Para finalizar, tan solo habrá que guardarlo y ¡listo!



## Paso 1

Debemos crear nuestro proyecto en Heroku, accionando clic sobre la opción **Crear una nueva app**.

The screenshot shows the Heroku dashboard interface. At the top, there's a navigation bar with the Heroku logo, a search bar, and a user profile section labeled 'Personal'. Below this is a purple banner with the text 'Welcome to Heroku' and a 'Dismiss' button. The main content area features two primary actions: 'Create a new app' and 'Create a team'. The 'Create a new app' section includes a sub-header 'Create a new app', a description 'Create your first app and deploy your code to a running dyno.', and a 'Create new app' button. The 'Create a team' section includes a sub-header 'Create a team', a description 'Create teams to collaborate on your apps and pipelines.', and a 'Create a team' button. Below these, there's a section titled 'Looking for help getting started with your language?' with a link to 'Get started by reading one of our language guides in the Dev Center'. This section lists various programming languages with their respective logos: Node.js, Ruby, Java, PHP, Python, Go, Scala, and Clojure. At the bottom, there's a footer with links to 'heroku.com', 'Blogs', 'Careers', 'Documentation', and 'Support' (highlighted in a purple button). On the right side of the footer, there are links for 'Terms of Service', 'Privacy', 'Cookies', and a copyright notice '© 2022 Salesforce.com'.



## Paso 2

Ponerle un **nombre** a la **app**. El nombre que elijamos es el nombre que vamos a usar para acceder a la página. Ejemplo: *mi-app-de-node.herokuapp.com*.

En la **región**, dejar "United States" y hacer click en **Create App**.

Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Create New App

App name

dh-fullstack-app

dh-fullstack-app is available

Choose a region

United States

Add to pipeline...

Create app

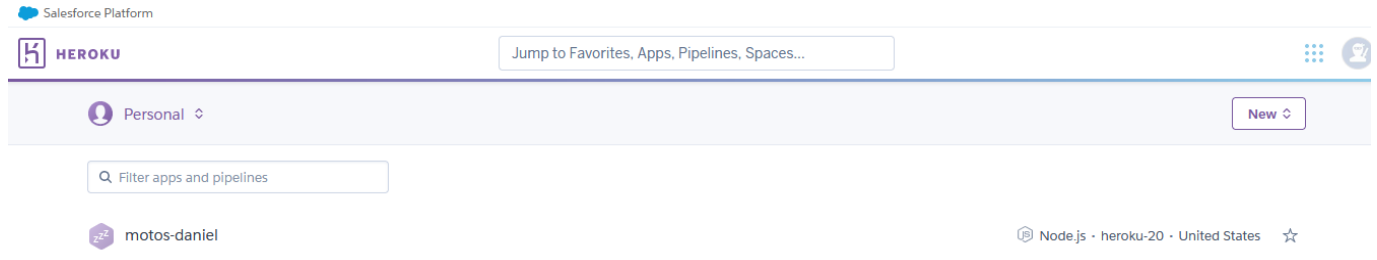
heroku.com Blogs Careers Documentation Support

Terms of Service Privacy Cookies © 2022 Salesforce.com

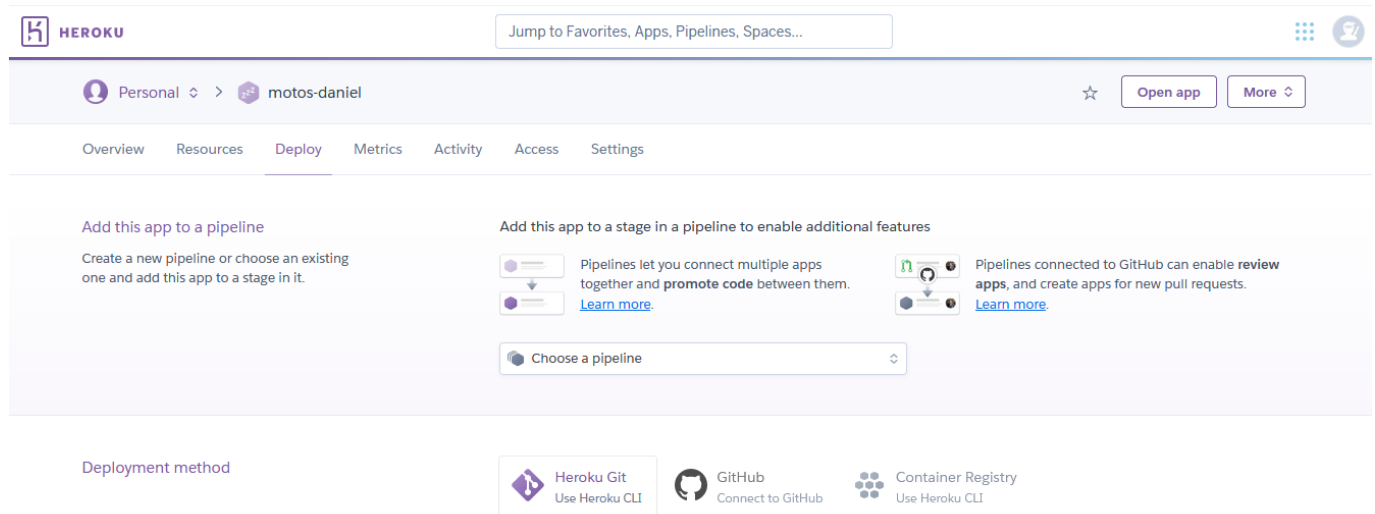


## Paso 3

Una vez creada nuestra App, debemos ingresar a ella, accionando clic sobre la aplicación creada, en este proyecto le llamados a la App: **motos-daniel**



Luego, tendremos que **conectar** nuestra **cuenta** y elegir el repositorio donde queremos publicar nuestro proyecto, para ello debe accionar clic sobre la opción **Deploy**, y luego debemos seleccionar la opción :







## Paso 4

Ahora debemos proceder a instalar en nuestro equipo, [Heroku Git](#).

The screenshot shows the Heroku deployment interface. At the top, there's a navigation bar with the Heroku logo and a search bar. Below the navigation bar, there's a section titled "Deployment method" with three options: "Heroku Git", "GitHub", and "Container Registry". The "Heroku Git" option is selected, and it shows a sub-section "Deploy using Heroku Git" with the instruction "Use git in the command line or a GUI tool to deploy this app." To the right, there's a section titled "Install the Heroku CLI" with the instruction "Download and install the [Heroku CLI](#)". Below this, there's a code block with the command `$ heroku login`. Further down, there's a section titled "Create a new Git repository" with the instruction "Initialize a git repository in a new or existing directory". Below this, there's a code block with the commands `$ cd my-project/`, `$ git init`, and `$ heroku git:remote -a dh-fullstack-app`. Next, there's a section titled "Deploy your application" with the instruction "Commit your code to the repository and deploy it to Heroku using Git." Below this, there's a code block with the commands `$ git add .`, `$ git commit -am "make it better"`, and `$ git push heroku master`. At the bottom, there's a note that says "You can now change your main deploy branch from 'master' to 'main' for both manual and automatic deploys, please follow the instructions [here](#)." Below this, there's a section titled "Existing Git repository" with the instruction "For existing repositories, simply add the `heroku` remote". Below this, there's a code block with the command `$ heroku git:remote -a dh-fullstack-app`.

Debemos accionar clic sobre la opción: **Download and install the [Heroku CLI](#)**. y allí veremos la siguiente pantalla e instalamos el producto en nuestro equipo, tomando en cuenta el tipo de sistema operativo:


## Install the Heroku CLI

### Pre-requisites

The Heroku CLI requires **Git**, the popular version control system. If you don't already have Git installed, complete the following:

- [Git installation](#)
- [First-time Git setup](#)

### Install with an Installer

 The Windows installers display a warning titled "Windows protected your PC" to some users. To run the installation when this warning shows, click "More info", verify the publisher as "salesforce.com, inc", then click the "Run anyway" button.

#### 🍏 macOS

```
$ brew tap heroku/brew && brew install heroku
```

#### 🪟 Windows

Download the appropriate installer for your Windows installation:

[64-bit installer](#)

[32-bit installer](#)

#### 🐧 Ubuntu 16+

Run the following from your terminal:

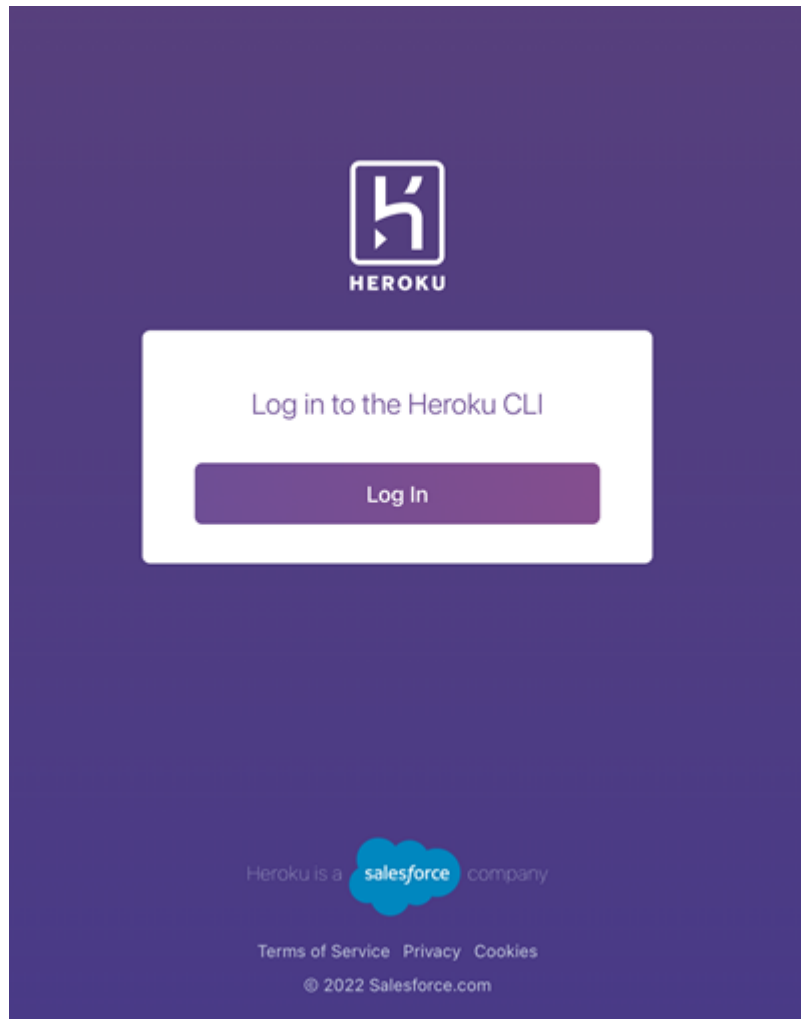
```
$ sudo snap install --classic heroku
```

[Snap is available on other Linux OS's as well.](#)

Luego de haber instalado **Heroku Cli**, ahora debemos loguearnos, a la plataforma desde la carpeta de nuestro proyecto, en nuestra PC local, para ello ejecutamos el comando **heroku login** y luego accionamos clic:

```
dfuentes@AR-BE-FS-05 MINGW64 ~/Desktop/backup daniel fuentes/DIGITAL-HOUSE-2022/nodejs-html-css
$ heroku login
» Warning: heroku update available from 7.42.13 to 7.60.1.
heroku: Press any key to open up the browser to login or q to exit: █
```

Aparece la siguiente ventana para loguearnos:



Luego de colocar nuestro usuario y contraseña nuestra cuenta en Heroku, podemos si queremos cerrar la ventana que aparece y debemos regresar a la consola de nuestro proyecto:

```
✓ TERMINAL BASH +
$ heroku login
» Warning: heroku update available from 7.42.13 to 7.60.1.
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/628472da-3f36-420a-bef2-b0d9117b81f1?re
r=SFMyNTY.g2gDbQAAAA0xODEuOTMuMzYuMjM0bgYACP-paIABYgABUYA.dRrLkqSC0vwuAoO_T3gXSvRvX7jr-6SuFBcn5VLdG1U
Logging in... done
Logged in as cedavilu@gmail.com

dfuentes@AR-BE-FS-05 MINGW64 ~/Desktop/backup daniel fuentes/DIGITAL-HOUSE-2022/nodejs-html-css
$
```

A continuación, debemos ejecutar el comando: **git init**, y de esta forma inicializamos nuestro

proyecto como un repositorio GIT.

```
▼ TERMINAL BASH + ▾

dfuentes@AR-BE-FS-05 MINGW64 ~/Desktop/backup daniel fuentes/DIGITAL-HOUSE-2022/nodejs-html-css
$ git init
Initialized empty Git repository in C:/Users/dfuentes/Desktop/backup daniel fuentes/DIGITAL-HOUSE-2022/nodejs-html-css/.git/

dfuentes@AR-BE-FS-05 MINGW64 ~/Desktop/backup daniel fuentes/DIGITAL-HOUSE-2022/nodejs-html-css (master)
$
```

Una vez que hacemos esto, debemos sincronizar nuestro repositorio local con el repositorio remoto en **Heroku**, ejecutando el siguiente comando: **Heroku git:remote -a nombreDelProyecto**, en nuestro ejemplo: **heroku git:remote -a motos-daniel**

```
$ heroku git:remote -a motos-daniel
» Warning: heroku update available from 7.42.13 to 7.60.1.
set git remote heroku to https://git.heroku.com/motos-daniel.git

dfuentes@AR-BE-FS-05 MINGW64 ~/Desktop/backup daniel fuentes/DIGITAL-HOUSE-2022/nodejs-html-css (master)
$
```

Procedemos a ejecutar las siguientes acciones para entonces lograr subir nuestro proyecto a **Heroku**.

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

```
$ git add .
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in index.js.
The file will have its original line endings in your working directory
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.
```

```
$ git commit -am "Subiendo el proyecto a Keroku"
[master (root-commit) 5673b84] Subiendo el proyecto a Keroku
28 files changed, 3650 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md
create mode 100644 index.js
create mode 100644 package-lock.json
create mode 100644 package.json
create mode 100644 public/css/master.css
create mode 100644 public/css/registro.css
create mode 100644 public/img/banner.jpg
```

```
$ git push heroku master
Enumerating objects: 34, done.
Counting objects: 100% (34/34), done.
Delta compression using up to 4 threads
Compressing objects: 100% (33/33), done.
Writing objects: 100% (34/34), 1.86 MiB | 506.00 KiB/s, done.
Total 34 (delta 1), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Using buildpack: heroku/nodejs
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:       NPM_CONFIG_LOGLEVEL=error
remote:       NODE_VERBOSE=false
remote:       NODE_ENV=production
remote:       NODE_MODULES_CACHE=true
```

Finalmente, para comprobar que nuestro proyecto se encuentra en Heroku, ingresamos a la plataforma y en la pantalla de nuestra App, accionamos clic sobre la opción **Overview** y allí

notaremos que nuestro proyecto se subió a la nube:

Salesforce Platform

HEROKU Jump to Favorites, Apps, Pipelines, Spaces...

Personal > motos-daniel ☆ Open app More

Overview Resources Deploy Metrics Activity Access Settings

Get a complete visualization of your app in a team-based continuous delivery environment with [Heroku Pipelines](#). Hide Create a Heroku Pipeline

Installed add-ons \$0.00/month Configure Add-ons

There are no add-ons for this app  
You can add add-ons to this app and they will show here. [Learn more](#)

Dyno formation \$0.00/month Configure Dynos

This app is using free dynos

Latest activity All Activity

cedavilu@gmail.com: Deployed 5673b846  
Today at 11:15 PM · v5

cedavilu@gmail.com: Build succeeded  
Today at 11:15 PM · [View build log](#)

cedavilu@gmail.com: Deployed 158b61be  
Apr 19, 2021 at 5:37 PM · v4



## Paso 5

Ahora, en la sección **Settings**, debemos ir a **Buildpacks** y presionar **Add Buildpack**.

Add buildpack

En la ventana que se abre elegir **Node.js** y guardar los cambios.

Add Buildpack ×

Enter Buildpack URL

heroku/nodejs

Or select from our officially supported buildpacks

nodejs

python

php

ruby

java

go

gradle

scala

clojure

Save changes

### Buildpacks

Buildpacks are scripts that are run when your app is deployed. They are used to install dependencies for your app and configure your environment. [Find new buildpacks on Heroku Elements](#)

Add buildpack

heroku/nodejs

×



## Paso 6

Luego en la opción de setting o configuración podemos observar la URL de nuestro proyecto subido a la plataforma **Heroku**:

Salesforce Platform

**HEROKU** Jump to Favorites, Apps, Pipelines, Spaces...

Personal > motos-daniel ☆ Open app More

Overview Resources Deploy Metrics Activity Access **Settings**

### App Information

App Name	motos-daniel
Region	United States
Stack	heroku-20
Framework	Node.js
Slug size	33.9 MiB of 500 MiB
Heroku git URL	<a href="https://git.heroku.com/motos-daniel.git">https://git.heroku.com/motos-daniel.git</a>

available for applications running on paid dynos to automate your SSL security.

Certificates acquired elsewhere may be configured using the [Manual Certificate](#) option.

### Domains

You can add custom domains to any Heroku app, then visit [Configuring DNS](#) to setup your DNS target.

Your app can be found at <https://motos-daniel.herokuapp.com/> Add domain

Filter domains

Custom domains will appear here

Custom domains allow you to access your app via one or more non-Heroku domain names (for example, [www.yourcustomdomain.com](#))

En base al ejemplo en cuestión, si usted toma esa URL y la dispone en su Browser de preferencia, logrará observar que el proyecto está en la nube y puede acceder:

<https://motos-daniel.herokuapp.com/>



**¡Happy Deploying!**