

Deploy Heroku

Documentación para subir nuestra aplicación al servidor de heroku.

1. Crear cuentas

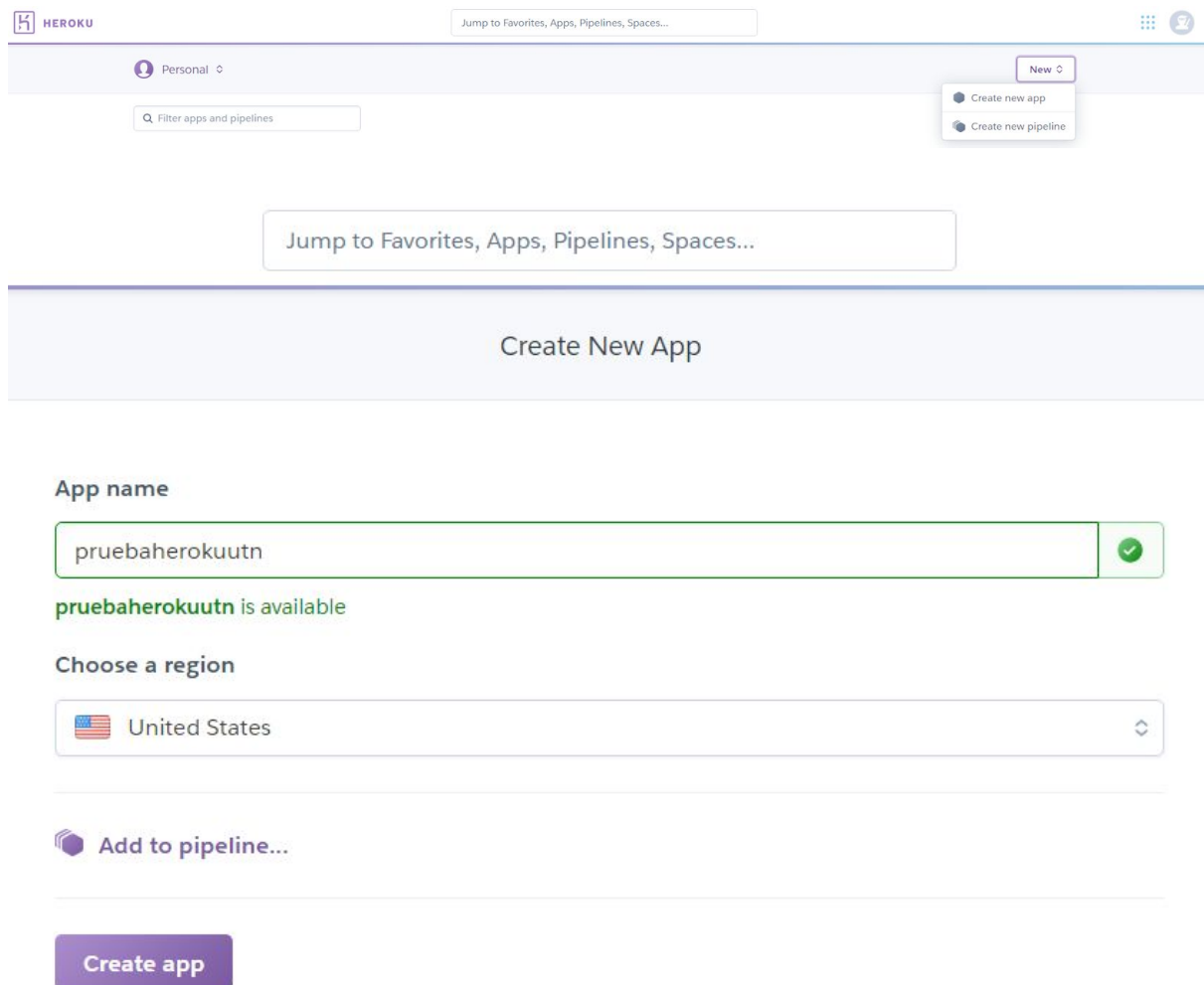
Debemos tener una cuenta en Heroku y github

www.heroku.com

www.github.com

2. Crear nueva App en Heroku

Una vez creadas las cuentas debemos ingresar a Heroku y 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 'New' button with a dropdown menu containing 'Create new app' and 'Create new pipeline'. Below the navigation bar is a large 'Create New App' button. Underneath, the 'App name' field is filled with 'pruebaherokuutn' and has a green checkmark icon. Below the name field, a message states 'pruebaherokuutn is available'. The 'Choose a region' dropdown menu is set to 'United States'. At the bottom, there is an 'Add to pipeline...' link and a prominent purple 'Create app' button.

3. Modificar Package

Debemos realizar algunos cambios en el package.json para que heroku realice las instalaciones y el deploy de nuestra app en angular de forma automática.

Ir al package.json y en la sección "devDependencies" y tomar las dependencias, "**@angular/cli**": "**~8.1.1**" y "**@angular/compiler-cli**": "**~8.1.1**", copiarlas a "dependencies". También copiar "**typescript**": "**~3.4.3**" a "dependencies".

Ahora en la sección “scripts” debemos realizar las siguientes modificaciones:

- agregar “**postinstall**”: “**ng build --output-path angularapp --aot --prod**”

Por último debemos generar una nueva sección:

```
“engines”: {  
  “node”: “v10.16.0”,  
  “npm”: “6.9.0”  
}
```

Tener en cuenta de utilizar las versiones que tengan instaladas en sus máquinas, para saber cuales son ejecutar en la consola de comandos:

- node -v
- npm -v

El package.json quedará de la siguiente manera:

```
{  
  "name": "prueba-heroku",  
  "version": "0.0.0",  
  "scripts": {  
    "ng": "ng",  
    "start": "ng serve",  
    "build": "ng build",  
    "test": "ng test",  
    "lint": "ng lint",  
    "e2e": "ng e2e",  
    "postinstall": "ng build --aot --prod"  
  },  
  "private": true,  
  "dependencies": {  
    "@angular/animations": "~8.1.1",  
    "@angular/common": "~8.1.1",  
    "@angular/compiler": "~8.1.1",  
    "@angular/cli": "~8.1.1",  
    "@angular/compiler-cli": "~8.1.1",  
    "@angular/core": "~8.1.1",  
    "@angular/forms": "~8.1.1",  
    "@angular/platform-browser": "~8.1.1",  
    "@angular/platform-browser-dynamic": "~8.1.1",  
    "@angular/router": "~8.1.1",  
    "rxjs": "~6.4.0",  
    "tslib": "^1.9.0",  
  },  
}
```

```

    "zone.js": "~0.9.1",
    "typescript": "~3.4.3"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.801.1",
    "@angular/language-service": "~8.1.1",
    "@types/node": "~8.9.4",
    "@types/jasmine": "~3.3.8",
    "@types/jasminewd2": "~2.0.3",
    "codelyzer": "^5.0.0",
    "jasmine-core": "~3.4.0",
    "jasmine-spec-reporter": "~4.2.1",
    "karma": "~4.1.0",
    "karma-chrome-launcher": "~2.2.0",
    "karma-coverage-istanbul-reporter": "~2.0.1",
    "karma-jasmine": "~2.0.1",
    "karma-jasmine-html-reporter": "^1.4.0",
    "protractor": "~5.4.0",
    "ts-node": "~7.0.0",
    "tslint": "~5.15.0"
  },
  "engines": {
    "node": "10.16.0",
    "npm": "6.9.0"
  }
}

```

4. Instalar express

Debemos instalar express para que sea la librería que realice la compilación una vez que realicemos el deploy en heroku.

En la consola de comandos situados en la carpeta raíz de nuestra aplicación ejecutar, **npm install --save express path**.

5. Crear el archivo server.js

Debemos crear el archivo server.js en la raíz de nuestra aplicación este será el que ejecute los comandos de angular para compilar la aplicación en el deploy a heroku.

En el archivo vamos a poner el siguiente código:

```

const express = require('express');
const app = express();
const path = require('path');

```

```
app.use(express.static(__dirname + '/angularapp'));

app.listen(process.env.PORT || 8080);

app.get('/', function(req, res) {
    res.sendFile(path.join(__dirname + '/angularapp/index.html'));
});

console.log('Build successful!!');
```

Una vez realizada esta modificación, debemos cambiar en el package.json el tag “start” agregando que ejecute el archivo que nosotros creamos previamente.

Reemplazar el tag actual por:

"start": "node server.js"

6. Generar un repositorio en GitHub

Podemos crear uno nuevo o utilizar uno ya existente.

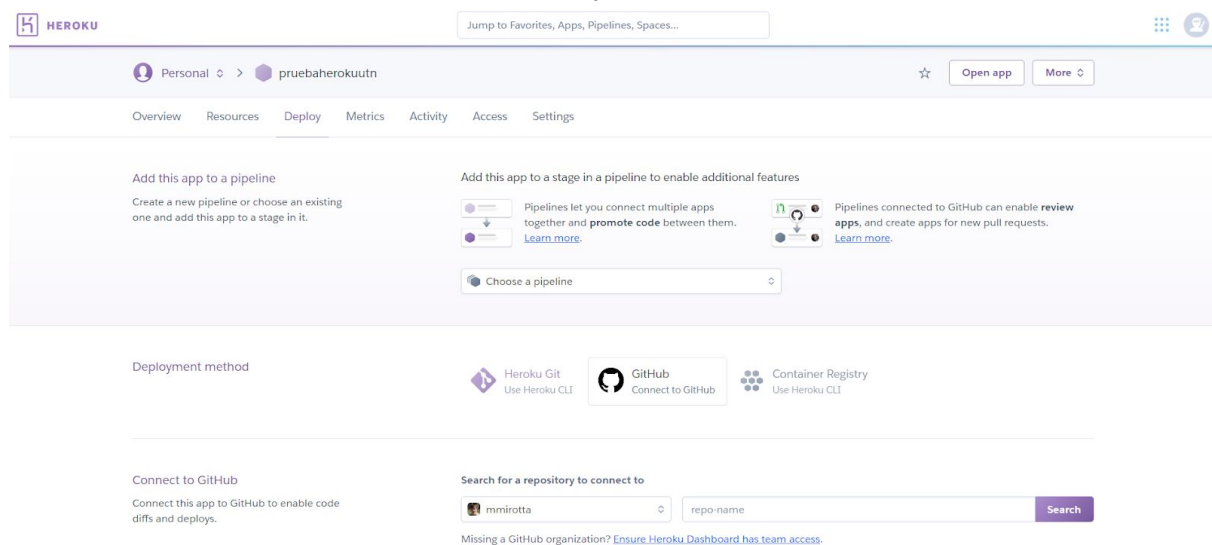
Una vez identificado el repositorio, en caso de que no lo tengamos asociado a nuestra aplicación, vamos a eliminar la carpeta .git de la misma (Esta se generó al realizar el ng new).

Luego, vamos a ejecutar los siguientes comandos:

```
git init
git add .
git commit -m "Primer commit. Modificaciones para deploy heroku"
git remote add origin https://github.com/mmirotta/pruebaherokuutn.git
```

7. Asociar App a repositorio Github

Debemos seleccionar en la sección de Deployment Method GitHub.



Una vez asociado a nuestro GitHub debemos buscar el repositorio creado previamente y conectarnos a él.

The screenshot shows the 'Connect to GitHub' section of the Heroku dashboard. On the left, it says 'Connect this app to GitHub to enable code diffs and deploys.' On the right, there's a search bar with the text 'Search for a repository to connect to'. Below the search bar, there's a dropdown menu showing 'mmirotta' and a text input field containing 'prueba'. A 'Search' button is to the right of the input field. Below the search results, there's a link: 'Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)'. At the bottom, there are two repository entries: 'mmirotta / pruebaherokuutn' and 'mmirotta / ArquitecturaYSO_2C', each with a 'Connect' button.

Una vez conectados al repositorio debemos realizar el deploy en heroku a través del comando `git push -u origin master`

8. Deploy Heroku

El deploy puede ser manual o automático, para eso debemos habilitarlo desde nuestro dashboard.

The screenshot shows the 'Automatic deploys' section of the Heroku dashboard. On the left, it says 'Automatic deploys' and 'Enables a chosen branch to be automatically deployed to this app.' On the right, it says 'Enable automatic deploys from GitHub' and 'Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more.](#)'. Below this, there's a section 'Choose a branch to deploy' with a dropdown menu showing 'master'. There's also a checkbox labeled 'Wait for CI to pass before deploy' with the text 'Only enable this option if you have a Continuous Integration service configured on your repo.' At the bottom, there's a button labeled 'Enable Automatic Deploys'. Below this, there's a section 'Manual deploy' with the text 'Deploy the current state of a branch to this app.' and 'Deploy a GitHub branch'. It says 'This will deploy the current state of the branch you specify below. [Learn more.](#)'. Below this, there's a section 'Choose a branch to deploy' with a dropdown menu showing 'master' and a button labeled 'Deploy Branch'.

Luego de habilitarlo, por ser la primera vez, debemos realizar el deploy manualmente. Para ello hacer click en **Deploy Branch**.

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

master

Deploy Branch

Receive code from GitHub

Build **master** 2ac8f4b3

```
engines.node (package.json): 10.16.0
engines.npm (package.json): 6.9.0

Resolving node version 10.16.0...
Downloading and installing node 10.16.0...
npm 6.9.0 already installed with node

----> Installing dependencies
Installing node modules (package.json + package-lock)

Autoscroll with output View build log
```

Release phase

Deploy to Heroku

Vamos a ir teniendo el log de nuestro deploy y podemos si debemos realizar alguna acción o si genero el deploy de forma correcta. Una vez finalizado el deploy heroku nos dará acceso a nuestra página.

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

master

Deploy Branch

Receive code from GitHub

Build **master** 2ac8f4b3

Release phase

Deploy to Heroku

Your app was successfully deployed.

View

Activity

En la sección activity vamos a poder ver el estado de nuestros deploy y en caso de error ver un log que nos dará información para poder solucionarlo.

Personal > ciclodevida

GitHub mmirotta/ciclodevida master

Overview Resources Deploy Metrics **Activity** Access Settings

Activity Feed

- m.mirota@gmail.com: Build in progress
Just now · [View build progress](#)
- m.mirota@gmail.com: Build failed
Today at 5:18 PM · [View build log](#)
- m.mirota@gmail.com: Deployed a19a8dab
Today at 5:07 PM · v6 · [Compare diff](#)
- m.mirota@gmail.com: Build succeeded
Today at 5:05 PM · [View build log](#)
- m.mirota@gmail.com: Deployed ae8b7e1
Today at 4:58 PM · v5 · [Roll back to here](#) · [Compare diff](#)
- m.mirota@gmail.com: Build succeeded
Today at 4:54 PM · [View build log](#)

Logs

Podemos ver los logs de nuestro sitio web accediendo desde More a View Logs.

Personal > pruebaherokuutn

GitHub mmirotta/pruebaherokuutn master

Overview Resources Deploy Metrics Activity Access Settings

Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and **promote code** between them. [Learn more.](#)

Pipelines connected apps, and create app [Learn more.](#)

Choose a pipeline

More >

- View logs
- View webhooks
- Run console
- Production check
- Add to pipeline...
- Restart all dynos