
UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

FACULTAD DE PRODUCCIÓN Y SERVICIOS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



Configuración de ambiente de trabajo - LAB07

Presentado por:

Integrantes	Turno Teoría	Turno Laboratorio
Choquehuanca Peraltilla, Angel Yvan	A	B
Leon Mamani, Rolando	B	B
Maraza Itomacedo, Diego	A	B
Ortiz Mamani, Waldir Flavio	A	B
Oviedo Yauri Diego Kevin	B	B
Quispe Humalla, Virgilia	A	B

AREQUIPA – PERÚ

2020

Escuela de Postgrado

Manual para Clonar, Ejecutar y Desplegar el Proyecto Frontend

ÍNDICE

INTRODUCCIÓN	2
Ejecutar el programa	5

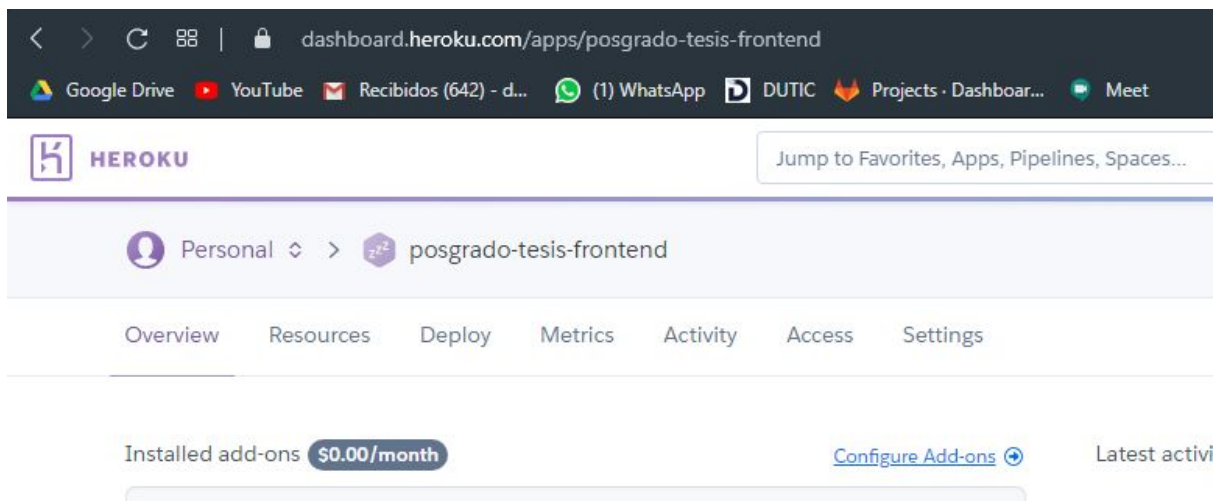
INTRODUCCIÓN

Este documento es elaborado con el objetivo de que los demás colaboradores de puedan apoyar en el desarrollo de proyecto teniendo como referencia este documento

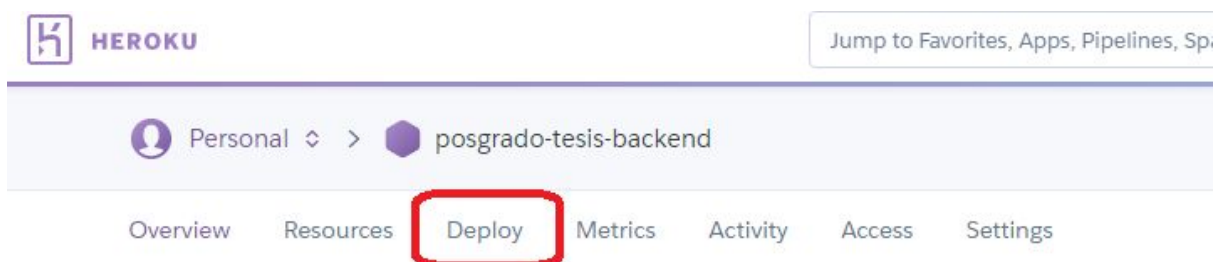
Clonar el Proyecto Frontend

Primeramente debemos ir a la plataforma Heroku en donde el proyecto se encuentra alojado

link: <https://dashboard.heroku.com/apps/posgrado-tesis-frontend>



Nos dirigimos a la pestaña “Deploy” que se encuentra en la barra de navegación



Nos vamos a la sección “Deploy using Heroku Cli” en el cual copiamos la siguiente línea

```
>> $ heroku git:clone -a posgrado-tesis-frontend
```

Deploy using Heroku Git

Use git in the command line or a GUI tool to deploy this app.

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

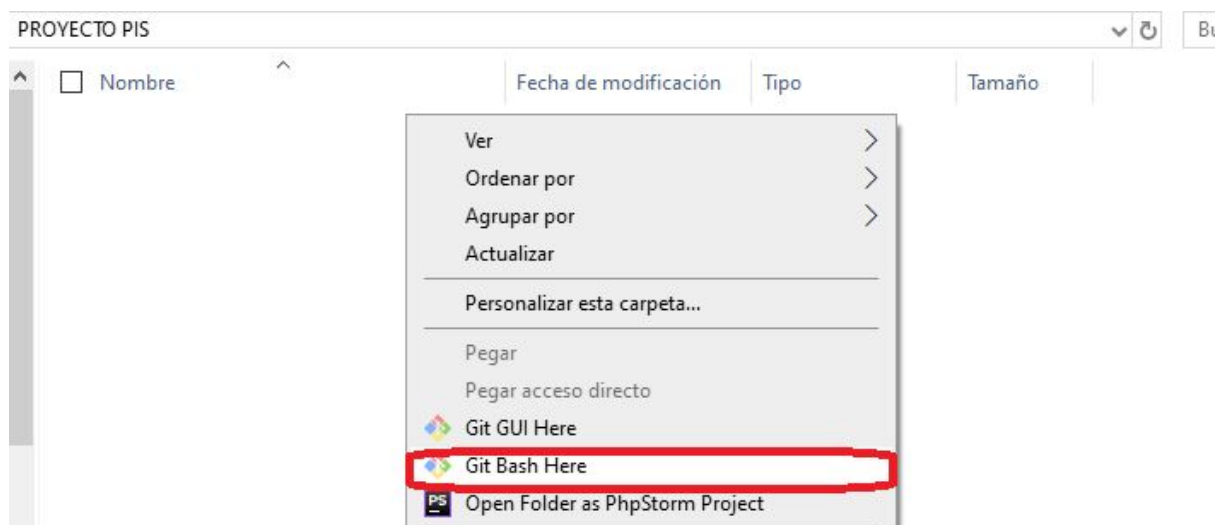
```
$ heroku login
```

Clone the repository

Use Git to clone posgrado-tesis-frontend's source code to your local machine.

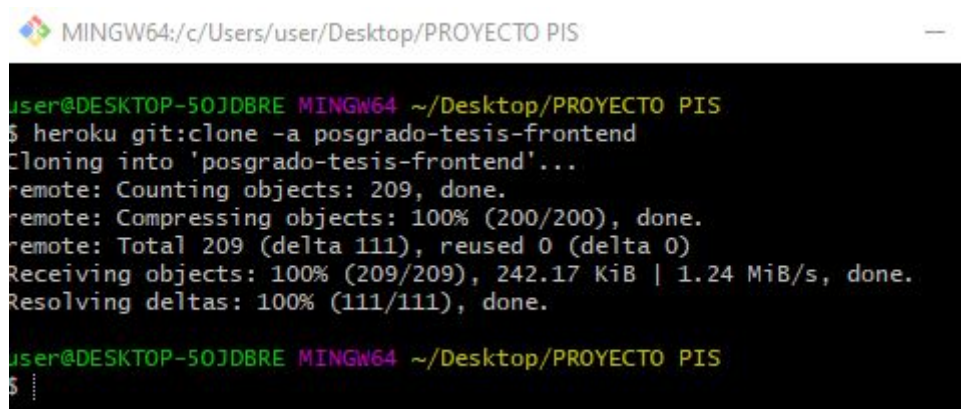
```
$ heroku git:clone -a posgrado-tesis-frontend  
$ cd posgrado-tesis-frontend
```

Una vez copiado, vamos a la carpeta en el cual se va a continuar desarrollando en el proyecto, para ello se debe tener instalado Angular, damos click derecho y escogemos “Git Bash Here”, cabe resalta que se debe tener instalado Git





















Haciendo click se nos abre la consola de Git en el cual pegamos el código copiado anteriormente de la plataforma Heroku para su clonación

```
heroku git:clone -a posgrado-tesis-frontend
```



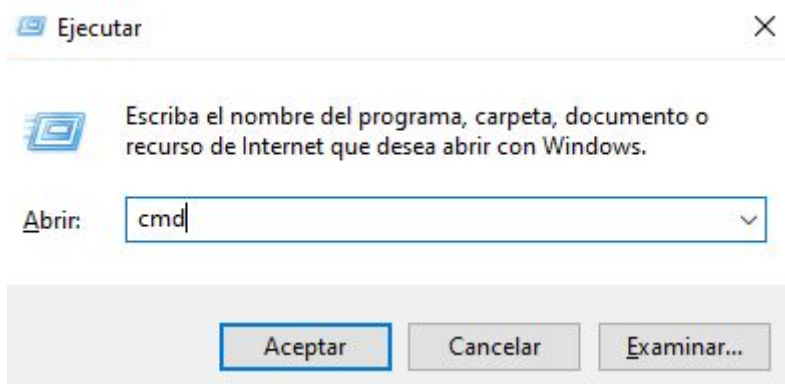
Verificamos

PROYECTO PIS				
<input type="checkbox"/> Nombre	Fecha de modificación	Tipo	Tamaño	
 posgrado-tesis-frontend	22/06/2020 02:01	Carpeta de archivos		

PROYECTO PIS > posgrado-tesis-frontend				
<input type="checkbox"/> Nombre	Fecha de modificación	Tipo	Tamaño	
 .git	22/06/2020 02:01	Carpeta de archivos		
 e2e	22/06/2020 02:01	Carpeta de archivos		
 src	22/06/2020 02:01	Carpeta de archivos		
 .angulardoc	22/06/2020 02:01	Archivo JSON	1 KB	
 .editorconfig	22/06/2020 02:01	Archivo EDITORC...	1 KB	
 .gitignore	22/06/2020 02:01	Documento de te...	1 KB	
 angular	22/06/2020 02:01	Archivo JSON	4 KB	
 browserslist	22/06/2020 02:01	Archivo	1 KB	
 karma.conf	22/06/2020 02:01	JetBrains PhpStorm	2 KB	
 package	22/06/2020 02:01	Archivo JSON	2 KB	
 package-lock	22/06/2020 02:01	Archivo JSON	526 KB	
 README.md	22/06/2020 02:01	Archivo MD	2 KB	
 server	22/06/2020 02:01	JetBrains PhpStorm	1 KB	
 tsconfig.app	22/06/2020 02:01	Archivo JSON	1 KB	
 tsconfig	22/06/2020 02:01	Archivo JSON	1 KB	
 tsconfig.spec	22/06/2020 02:01	Archivo JSON	1 KB	
 tslint	22/06/2020 02:01	Archivo JSON	4 KB	

Ejecutar el programa

Para ejecutar el programa abrimos el simbolo de sistema



```
C:\> C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.18362.900]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\user>
```

Nos dirigimos a la carpeta donde se ha clonado el proyecto

```
C:\> C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.18362.900]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\user>cd Desktop
C:\Users\user\Desktop>cd "PROYECTO PIS"
C:\Users\user\Desktop\PROYECTO PIS>cd posgrado-tesis-frontend
C:\Users\user\Desktop\PROYECTO PIS\posgrado-tesis-frontend>
```

Ahora debemos escribir el siguiente comando

```
>> npm install --save-dev @angular-devkit/build-angular
```

```
C:\Users\user\Desktop\PROYECTO PIS\posgrado-tesis-frontend>npm install --save-dev @angular-devkit/build-angular
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node v14+. Upgrade to chokidar 3 with 15x less dependencies.
npm WARN deprecated fsevents@1.2.13: fsevents 1 will break on node v14+ and could be using insecure binaries. Upgrade to fse
```

Esperamos a que termine de cargar

Finalmente para ejecutar el proyecto, escribimos el siguiente comando

>> ng serve

```
C:\Users\user\Desktop\PROYECTO PIS\posgrado-tesis-frontend>ng serve
Compiling @angular/core : es2015 as esm2015
Compiling @angular/animations : es2015 as esm2015
Compiling @angular/compiler/testing : es2015 as esm2015
Compiling @angular/core/testing : es2015 as esm2015
Compiling @angular/common : es2015 as esm2015
Compiling @angular/animations/browser : es2015 as esm2015
Compiling @angular/animations/browser/testing : es2015 as esm2015
Compiling @angular/platform-browser : es2015 as esm2015
Compiling @angular/common/http : es2015 as esm2015
Compiling @angular/common/testing : es2015 as esm2015
Compiling @angular/platform-browser-dynamic : es2015 as esm2015
Compiling @angular/platform-browser/testing : es2015 as esm2015
Compiling @angular/router : es2015 as esm2015
Compiling @angular/common/http/testing : es2015 as esm2015
Compiling @angular/forms : es2015 as esm2015
Compiling @angular/platform-browser/animations : es2015 as esm2015
Compiling @angular/platform-browser-dynamic/testing : es2015 as esm2015
Compiling @angular/router/testing : es2015 as esm2015

chunk {main} main.js, main.js.map (main) 84.7 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry]
chunk {styles} styles.js, styles.js.map (styles) 928 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3.41 MB [initial] [rendered]
Date: 2020-06-22T07:38:42.454Z - Hash: 353534307a546a958f48 - Time: 10.1s
** Angular Live Development Server is listening on localhost:4200, C
: Compiled successfully.
```

Una vez compilador nos dirigimos a un navegador y escribimos

>> <http://localhost:4200>

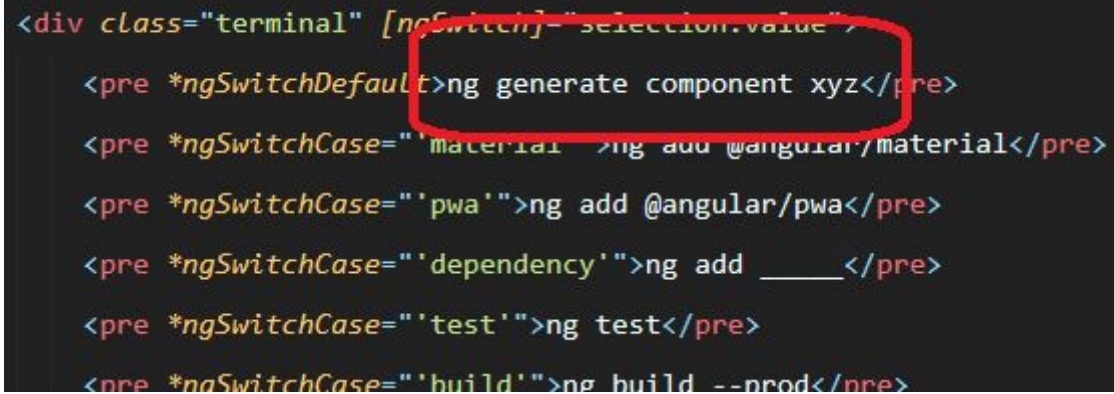


Desplegar Proyecto

Una vez clonado y desarrollado nuevos avances el siguiente paso es deployar la nueva version que hemos originado.

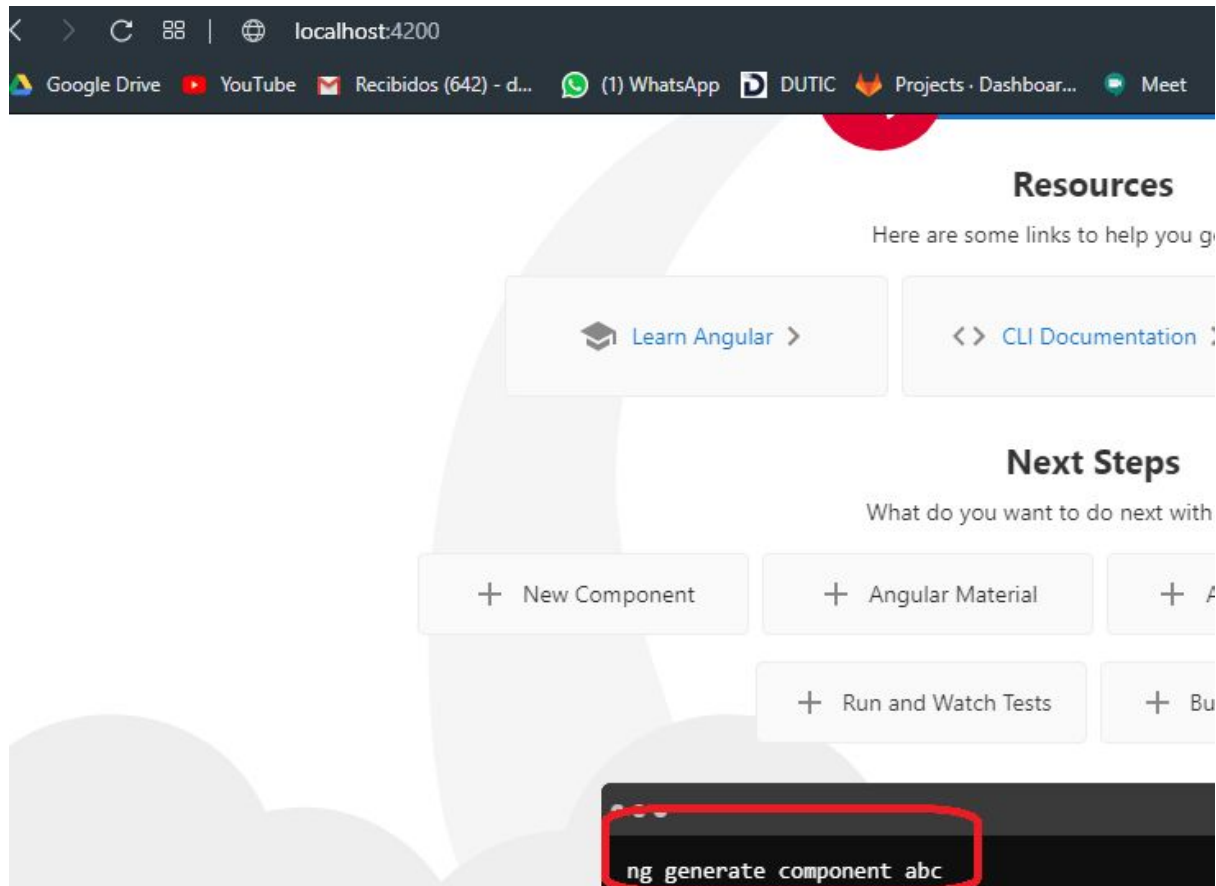
Ejemplo modificamos

“ng generate component xyz” por “ng generate component abc”



```
<div class="terminal" [ngSwitch]="selection.value">
  <pre *ngSwitchDefault>ng generate component xyz</pre>
  <pre *ngSwitchCase="'material'">ng add @angular/material</pre>
  <pre *ngSwitchCase="'pwa'">ng add @angular/pwa</pre>
  <pre *ngSwitchCase="'dependency'">ng add ____</pre>
  <pre *ngSwitchCase="'test'">ng test</pre>
  <pre *ngSwitchCase="'build'">ng build --prod</pre>
```

Verificamos



Una vez modificado se ha generado una nueva versión entonces se procede a su despliegue

Para ello se usa los comandos en el git,

```
>>git add .
```

```
>>git commit -m "describir lo desarrollado"
```

```
>>git push Heroku master
```

```
user@DESKTOP-5OJDBRE MINGW64 ~/Desktop/PROYECTO PIS/marazangular (master)
$ git status
On branch master
Your branch is up to date with 'heroku/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   package-lock.json
        modified:   package.json
        modified:   src/app/app.component.html

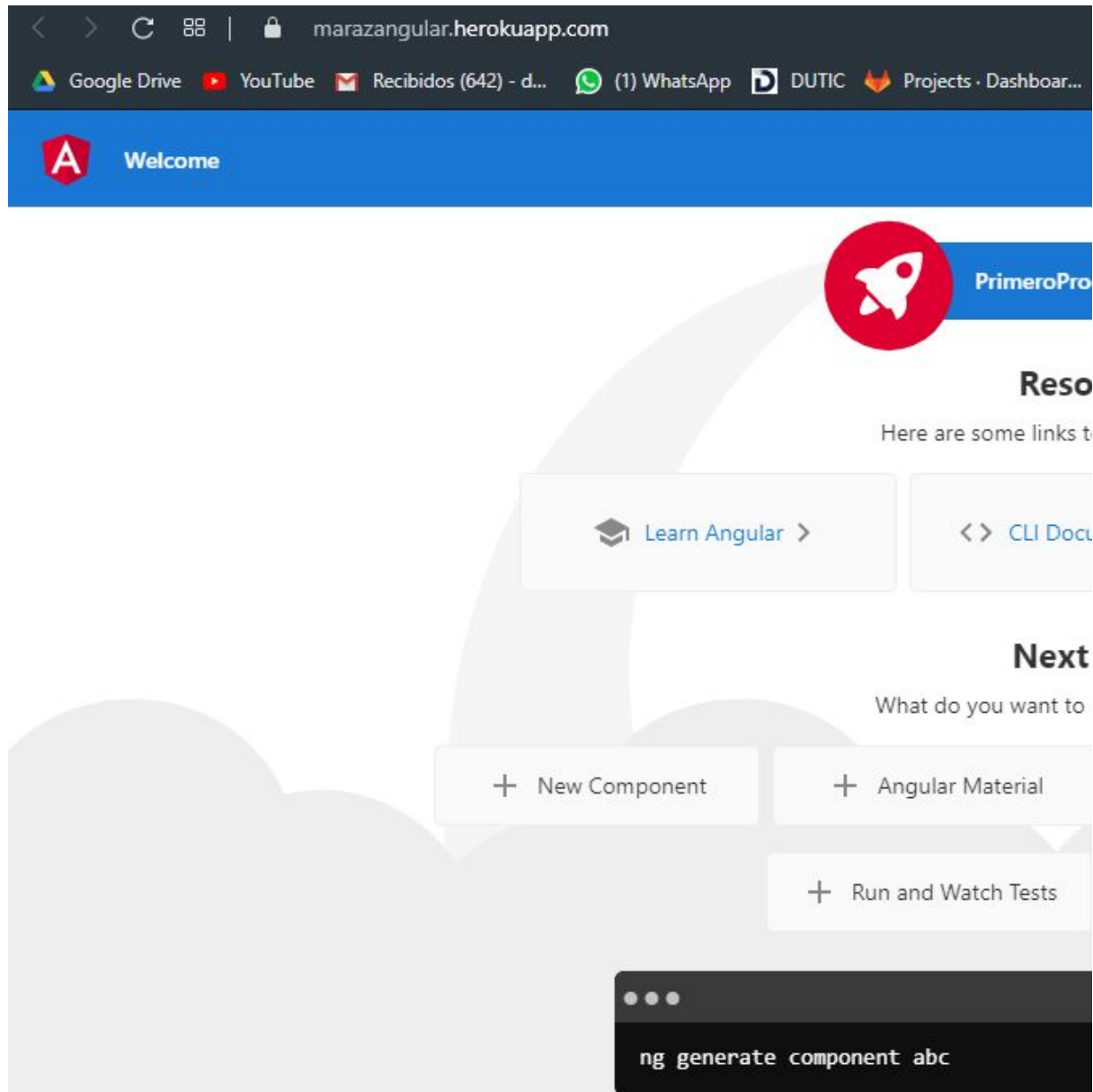
no changes added to commit (use "git add" and/or "git commit -a")

user@DESKTOP-5OJDBRE MINGW64 ~/Desktop/PROYECTO PIS/marazangular (master)
$ git add .

user@DESKTOP-5OJDBRE MINGW64 ~/Desktop/PROYECTO PIS/marazangular (master)
$ git commit -m "cambiar de xvy a abc"
[master 3032dcb] cambiar de xvy a abc
 3 files changed, 924 insertions(+), 199 deletions(-)

user@DESKTOP-5OJDBRE MINGW64 ~/Desktop/PROYECTO PIS/marazangular (master)
$ git push heroku master
Enumerating objects: 13, done.
Counting objects: 100% (13/13) done
```

Verificando



Escuela de Postgrado

Manual para Clonar, Ejecutar y Desplegar el Proyecto Backend

INTRODUCCIÓN	3
Clonar Proyecto Backend	3
Ejecutar proyecto	6
Deployar Proyecto	8

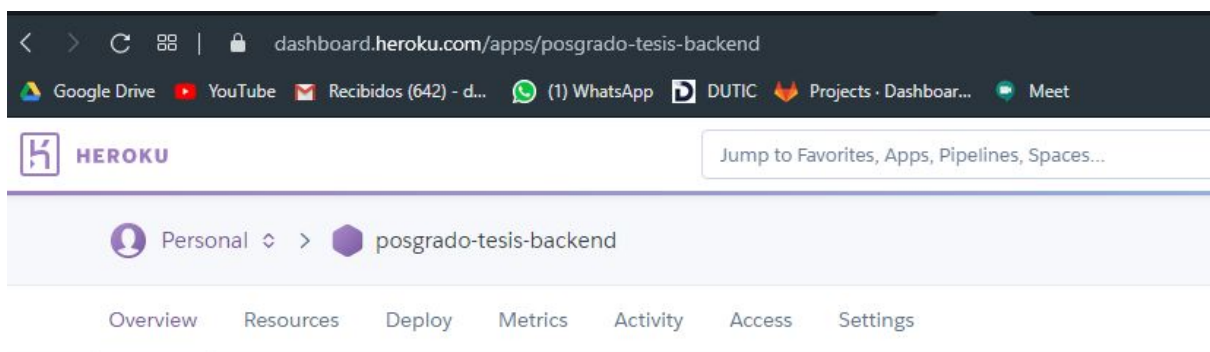
INTRODUCCIÓN

Este documento es elaborado con el objetivo de que los demás colaboradores de puedan apoyar en el desarrollo de proyecto teniendo como referencia este documento

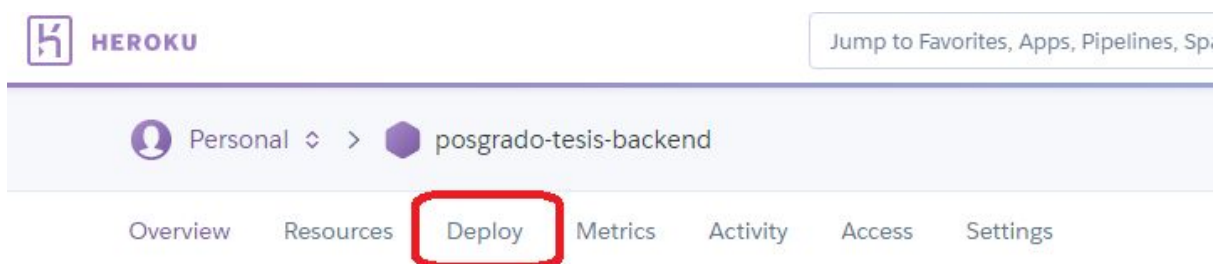
Clonar Proyecto Backend

Primeramente debemos ir al proyecto que se encuentra alojado en la plataforma heroku

link: <https://dashboard.heroku.com/apps/posgrado-tesis-backend>



Nos dirigimos a la pestaña “Deploy” que se encuentra en la barra de navegación



Nos vamos a la sección “Deploy using Heroku CLI” en el cual copiamos la siguiente línea

```
>> $ heroku git:clone -a posgrado-tesis-backend
```


Deploy using Heroku Git

Use git in the command line or a GUI tool to deploy this app.

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

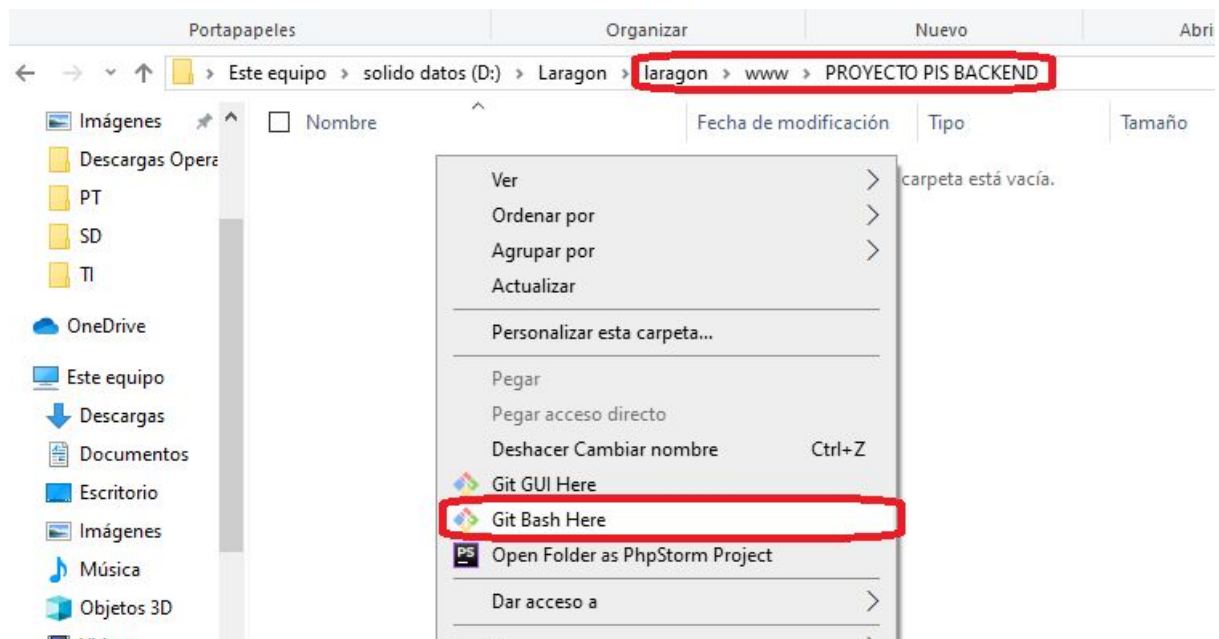
```
$ heroku login
```

Clone the repository

Use Git to clone posgrado-tesis-backend's source code to your local machine.

```
$ heroku git:clone -a posgrado-tesis-backend
```

Una vez copiado, vamos a la carpeta en el cual se va a continuar desarrollando en el proyecto, para ello se debe tener instalado laravel, esta carpeta se debe encontrar dentro de "www", damos click derecho y escogemos "Git Bash Here", cabe resalta que se debe tener instalado Git

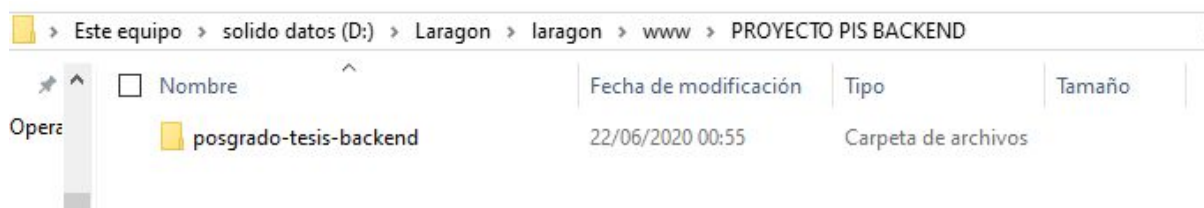


Haciendo click se nos abre la consola de Git en el cual pegamos el código copiado anteriormente de la plataforma Heroku para su clonación

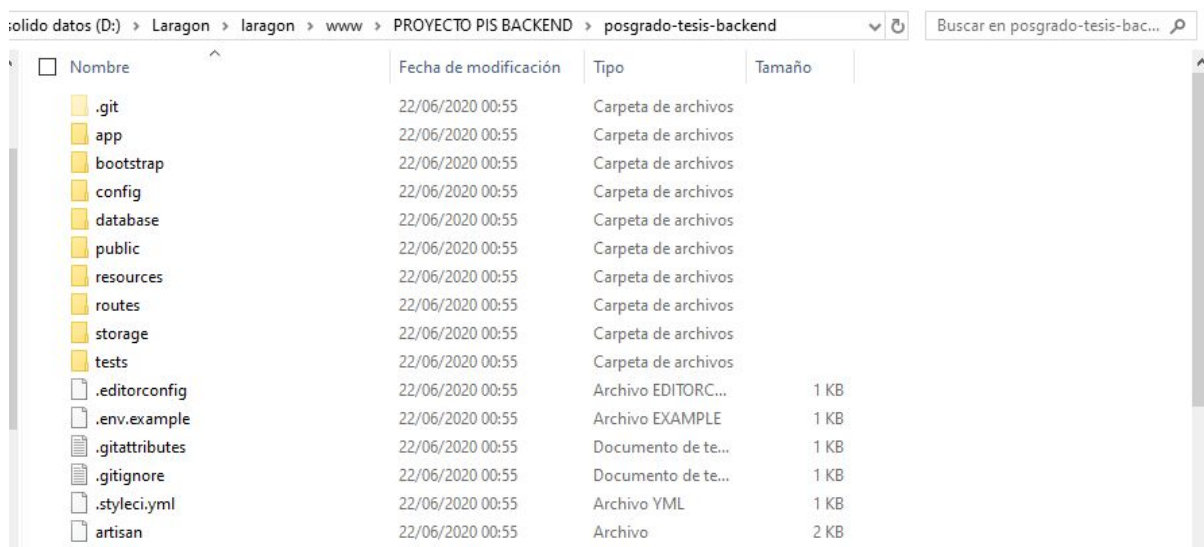
```
$ heroku git:clone -a posgrado-tesis-backend
```

```
MINGW64:/d/Laragon/laragon/www/PROYECTO PIS BACKEND
user@DESKTOP-5OJDBRE MINGW64 /d/Laragon/laragon/www/PROYECTO PIS BACKEND
$ heroku git:clone -a posgrado-tesis-backend
Cloning into 'posgrado-tesis-backend'...
remote: Counting objects: 348, done.
remote: Compressing objects: 100% (316/316), done.
remote: Total 348 (delta 154), reused 106 (delta 10)
Receiving objects: 100% (348/348), 183.80 KiB | 1.19 MiB/s, done.
Resolving deltas: 100% (154/154), done.
user@DESKTOP-5OJDBRE MINGW64 /d/Laragon/laragon/www/PROYECTO PIS BACKEND
$
```

Verificamos en la carpeta



Entramos a la carpeta “posgrado-tesis-backend”



Una vez clonado el proyecto se debe solicitar al equipo de back dos archivos “vendor” y “.env” en el cual son variables de entorno para su funcionamiento, estos archivos no se clonan por políticas de seguridad.

Este equipo > solido datos (D:) > Descargas Opera > posgrado-tesis-backend				
<input type="checkbox"/> Nombre	Fecha de modificación	Tipo	Tamaño	
vendor	13/06/2020 23:03	Carpeta de archivos		
.env	18/06/2020 17:43	Archivo ENV	2 KB	


Se copia estos dos archivos al proyecto que se ha clonado

solido datos (D:) > Laragon > laragon > www > PROYECTO PIS BACKEND > posgrado-tesis-backend >				
<input type="checkbox"/> Nombre	Fecha de modificación	Tipo	Tamaño	
.git	22/06/2020 00:55	Carpeta de archivos		
app	22/06/2020 00:55	Carpeta de archivos		
bootstrap	22/06/2020 00:55	Carpeta de archivos		
config	22/06/2020 00:55	Carpeta de archivos		
database	22/06/2020 00:55	Carpeta de archivos		
public	22/06/2020 00:55	Carpeta de archivos		
resources	22/06/2020 00:55	Carpeta de archivos		
routes	22/06/2020 00:55	Carpeta de archivos		
storage	22/06/2020 00:55	Carpeta de archivos		
tests	22/06/2020 00:55	Carpeta de archivos		
<input checked="" type="checkbox"/> vendor	22/06/2020 01:07	Carpeta de archivos		
.editorconfig	22/06/2020 00:55	Archivo EDITORC...	1 KB	
<input checked="" type="checkbox"/> .env	18/06/2020 17:43	Archivo ENV	2 KB	
.env.example	22/06/2020 00:55	Archivo EXAMPLE	1 KB	

Ejecutar proyecto

Una vez copiado, utilizamos Laragon para ejecutar su funcionamiento de manera local

Laragon Full 4.0.16 190914 php-7.4.6-Win32-vc15-x64 [TS] 192.168.56.1



Menú

h ? ⚙

© Leo K

The journey of a thousand miles begins with one step.

Iniciar Todo

Web

Base de Datos

Terminal

Root

Damos clic en “iniciar Todo”

Laragon Full 4.0.16 190914 php-7.4.6-Win32-vc15-x64 [TS] 192.168.56.1



Menú

h ? ⚙

© Leo K

Apache httpd-2.4.35-win64-VC15 started80Recargar

MySQL mysql-5.7.24-winx64 started3306

Detener

Web

Base de Datos

Terminal

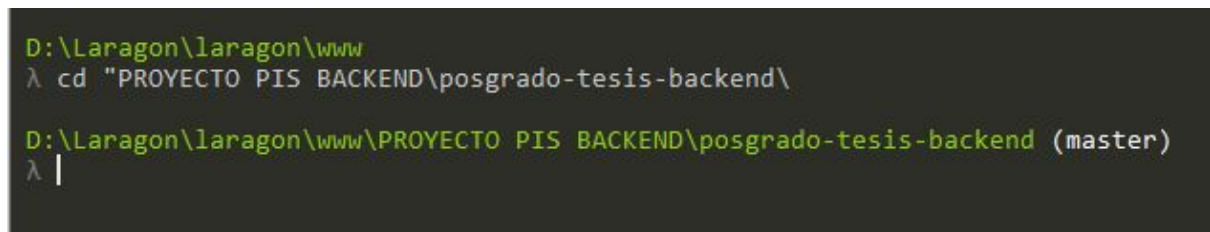
Root

Luego damos en clic en “Terminal”



```
Cmder
1. www
D:\Laragon\laragon\www
λ
```

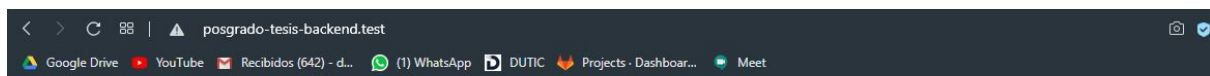
En el terminal entramos a la carpeta donde se ha clonado el proyecto



```
D:\Laragon\laragon\www
λ cd "PROYECTO PIS BACKEND\posgrado-tesis-backend\"
D:\Laragon\laragon\www\PROYECTO PIS BACKEND\posgrado-tesis-backend (master)
λ |
```

Una vez dentro de la carpeta, nos vamos a un navegador y escribimos

>> <http://posgrado-tesis-backend.test>



Deployar Proyecto

Una vez clonado y desarrollado nuevos avances el siguiente paso es deployar la nueva version que hemos originado.

Ejemplo modificamos

“Docs” a “Documents”

```
<div class= content >  
  
  <div class="title m-b-md">  
    Laravel  
  </div>  
  
  <div class="links">  
    <a href="https://laravel.com/docs">Documents</a>  
    <a href="https://laracasts.com">Laracasts</a>  
    <a href="https://laravel-news.com">News</a>
```

Una vez modificado se ha generado una nueva versión entonces se procede a su despliegue

Para ello se usa los comandos en el git,

```
>>git add .
```

```
>>git commit -m "describir lo desarrollado"
```

```
>>git push Heroku master
```



```

user@DESKTOP-5OJDBRE MINGW64 ~/Desktop/Clonando Angular y Laravel/Laravel/marazala
ravel (master)
$ git status
On branch master
Your branch is ahead of 'heroku/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   resources/views/welcome.blade.php

no changes added to commit (use "git add" and/or "git commit -a")

user@DESKTOP-5OJDBRE MINGW64 ~/Desktop/Clonando Angular y Laravel/Laravel/marazala
ravel (master)
$ git add .

user@DESKTOP-5OJDBRE MINGW64 ~/Desktop/Clonando Angular y Laravel/Laravel/marazala
ravel (master)
$ git commit -m "Modificando la palabra Docs a Documents"
[master 34e6d12] Modificando la palabra Docs a Documents
 1 file changed, 1 insertion(+), 1 deletion(-)

user@DESKTOP-5OJDBRE MINGW64 ~/Desktop/Clonando Angular y Laravel/Laravel/marazala
ravel (master)
$ git push heroku master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 587 bytes | 65.00 KiB/s, done.
Total 6 (delta 3), reused 1 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> PHP app detected
remote: -----> Bootstrapping...

```

Verificamos

Laravel

DOCUMENTS

LARACASTS

NEWS

BLOG

NOVA

FORGE

VAPOR

GITHUB

Asignación de Tareas mediante Issues

The screenshot shows the GitLab interface for the 'Graduate school project'. The left sidebar contains navigation links: Project overview, Issues (8), Boards, Labels, Service Desk, Milestones, Merge Requests (0), Requirements, CI / CD, and Security & Compliance. The main content area is titled 'Issue Boards' and shows a board for 'PIS - Semana 14'. The board has a column '14 - En Proceso' with three issues: 'Manual de Usuario V3.0' (#203, Tomorrow), 'PLANTILLA :Plan de Aseguramineto de la Calidad' (#206, Tomorrow), and 'SEGUIMIENTO Y CONTROL : Informe Individual sobre el avance durante el proyecto.' (#208, Tomorrow). The board also shows 'Open' (0) and 'Closed' (200) counts.

Asignación de Hitos

The screenshot shows the GitLab interface for the 'Graduate school project' Milestones. The left sidebar is the same as the previous screenshot. The main content area is titled 'Milestones' and shows a list of milestones. The milestones are: 'Hito 12: Cierre - Sprint 9' (Aug 9, 2020–Aug 13, 2020, 11 Issues, 0 Merge Requests, 27% complete), 'Hito 11: Ejecución - Sprint 8' (Aug 2, 2020–Aug 8, 2020, 27 Issues, 0 Merge Requests, 100% complete), 'Hito 10: Ejecución - Sprint 7' (Jul 29, 2020–Aug 1, 2020, 28 Issues, 0 Merge Requests, 100% complete), and 'Hito 09: Ejecución - Sprint 6' (Jul 18, 2020–Jul 25, 2020, 30 Issues, 0 Merge Requests, 100% complete). Each milestone has a 'Closed' status and a link to the project.

Informe de Desarrollo con Burndown Chart

