

# **MANUAL DE INSTALACIÓN**



## **DESARROLLO DE SISTEMA WEB DE GESTIÓN DE ASISTENCIAS EN LA UNIDAD EDUCATIVA JOSÉ MARÍA VELASCO IBARRA - QUITO**

**JAZMIN ANABELLE VILLAMARIN SARANGO**

jazmin.villamarin@epn.edu.ec

**VICTOR ALONSO PACHA AYO**

victor.pacha@epn.edu.ec

# ÍNDICE DE CONTENIDOS

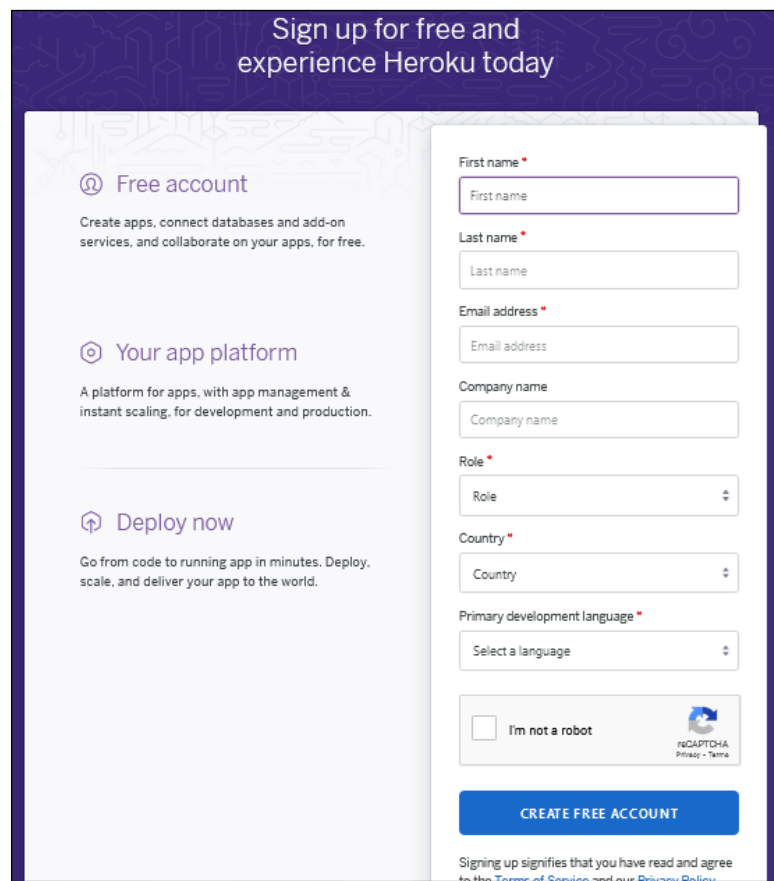
1.	Despliegue del sistema web .....	1
1.1	Configuración inicial .....	1
1.2	Inicialización del repositorio en GitHub.....	3
3.	Conexión a la base de datos .....	8
4.	Credenciales de acceso al sistema web.....	10
5.	Repositorio del código fuente del sistema web.....	11

# 1. Despliegue del sistema web

Culminada la etapa de pruebas del sistema web, se detallan los pasos que se han realizado para el despliegue del sistema web en Heroku, la base de datos en alwaysdata y la conexión respectiva.

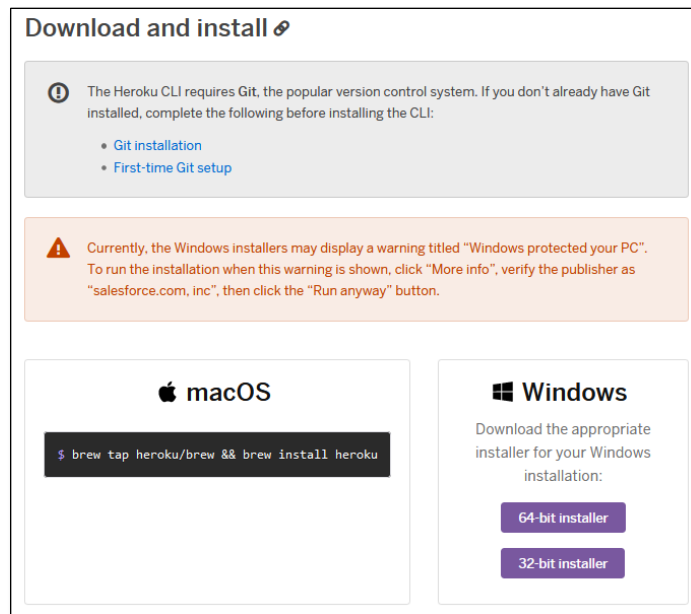
## 1.1 Configuración inicial

Como un primer paso se debe crear una cuenta en Heroku, como se ilustra en la **Fig. 1**.

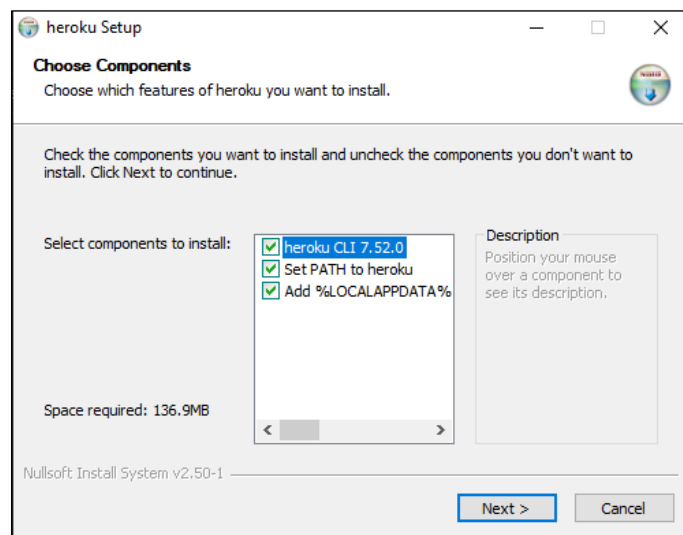
The image shows the Heroku sign-up page. At the top, it says "Sign up for free and experience Heroku today". On the left, there are three sections: "Free account" (with a @ icon), "Your app platform" (with a gear icon), and "Deploy now" (with a push icon). On the right, there is a form with fields for "First name", "Last name", "Email address", "Company name", "Role" (a dropdown menu), "Country" (a dropdown menu), and "Primary development language" (a dropdown menu). Below these fields is a checkbox for "I'm not a robot" with a reCAPTCHA logo. At the bottom of the form is a blue button that says "CREATE FREE ACCOUNT". Below the button, there is a small line of text: "Signing up signifies that you have read and agree to the Terms of Service and our Privacy Policy."

**Fig. 1.** Formulario para crear una cuenta en Heroku.

Luego, se procede a descargar e instalar la interfaz de comandos llamada Heroku CLI del siguiente enlace <https://devcenter.heroku.com/articles/heroku-cli>, el cual permite crear y administrar todas las aplicaciones de Heroku desde una terminal de comandos, como se ilustran en las **Fig. 2** y **Fig. 3**.

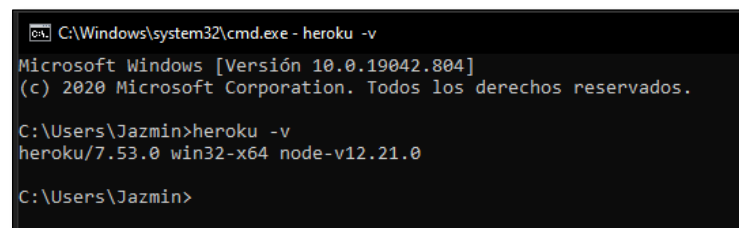


**Fig. 2.** Descargar Heroku CLI.



**Fig. 3.** Instalación de Heroku CLI.

Luego se procede a abrir una terminal de *Windows* y se ejecuta el siguiente comando *heroku -v* para verificar que Heroku CLI está instalado, como se ilustra en la **Fig. 4**.



**Fig. 4.** Comando para ver la versión de Heroku instalada.

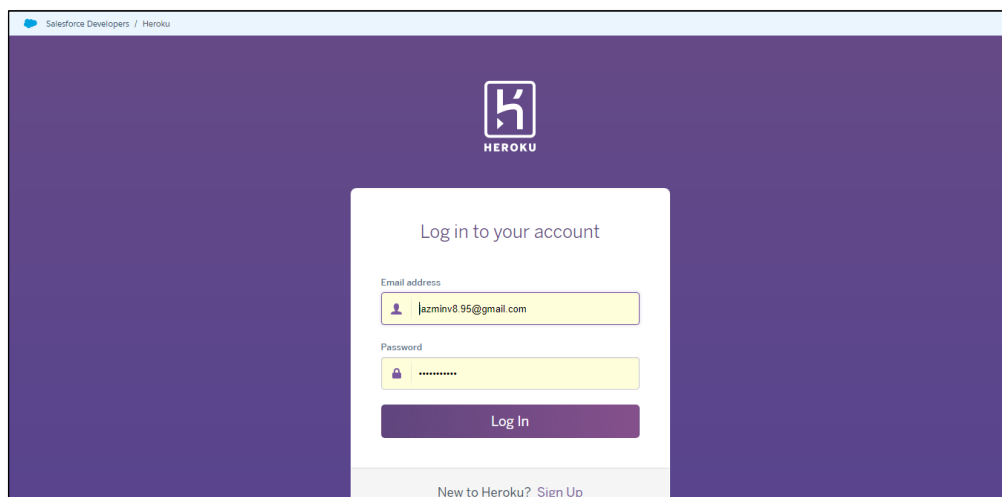
## 1.2 Inicialización del repositorio en GitHub

Ahora se procede a iniciar sesión en Heroku desde la terminal ejecutando el siguiente comando *heroku login*, como se ilustra en la **Fig. 5**.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19042.804]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.
C:\Users\Jazmin>heroku login
```

**Fig. 5.** Comando para iniciar sesión en Heroku.

Se procede a iniciar sesión en Heroku desde un navegador, como se ilustran en las **Fig. 6** y **Fig. 7**.



**Fig. 6.** Página de login en Heroku.

```
C:\Users\Jazmin>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/242750f8-e771-44a6-a9d7-5bbbcfef43f?requestor=5FMyNTY.g
2gDbQAAAA4xODEuMTk2LjcyLjE3MW4GAibag015AWIAAVGA.ojjzK11QoNCdHz19kD6bYCWj4KKr0SeFaq-Ctz7b6cE
Logging in... done
Logged in as jazminv8.95@gmail.com
C:\Users\Jazmin>
```

**Fig. 7.** Sesión iniciada desde la terminal.

Desde la terminal se debe dirigir al directorio donde se encuentra el proyecto y se debe ejecutar el siguiente comando *git init*, como se ilustra en la **Fig. 8**.

```
C:\Users\Jazmin>cd C:\Users\Jazmin\Documents\Tesis\2Repositorio\bk\20210303\tesis2
C:\Users\Jazmin\Documents\Tesis\2Repositorio\bk\20210303\tesis2>git init
Reinitialized existing Git repository in C:/Users/Jazmin/Documents/Tesis/2Repositorio/bk/20210303/tesis2/.git/
```

**Fig. 8.** Comando para inicializar el repositorio en GitHub.

Luego se debe agregar los archivos al repositorio con el siguiente comando *git add*, posterior a ello ejecutar el siguiente comando *git commit -m*, para agregar un comentario, como se ilustran en la **Fig. 9** y **Fig. 10**.

```
C:\Users\Jazmin\Documents\Tesis\2Repositorio\bk\20210303\tesis2>git add .
warning: LF will be replaced by CRLF in app/Http/Controllers/JornadaController.php.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in app/Http/Controllers/PermisoController.php.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in app/Http/Controllers/UsersController.php.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in app/Http/Requests/SaveUserRequest.php.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in resources/views/permisos/index.blade.php.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in resources/views/permisos/justificar.blade.php.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in resources/views/users/create.blade.php.
The file will have its original line endings in your working directory
```

**Fig. 9.** Comando para agregar los archivos del proyecto al repositorio.

```
C:\Users\Jazmin\Documents\Tesis\2Repositorio\bk\20210303\tesis2>git commit -m "subiendo proyecto"
[master 26f1ebc] subiendo proyecto
17 files changed, 27 insertions(+), 68 deletions(-)
delete mode 100644 resources/views/permiso_profesors/_form.blade.php
C:\Users\Jazmin\Documents\Tesis\2Repositorio\bk\20210303\tesis2>
```

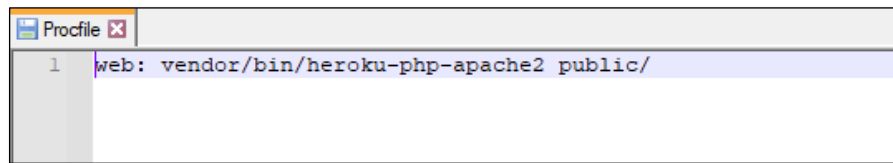
**Fig. 10.** Comando para realizar un comentario a la acción anterior.

Cuando se crea proyectos en Laravel, se debe crear un archivo llamado “Procfile” (sin ninguna extensión) en la raíz del proyecto, como se ilustra en la **Fig. 11**.

Nombre	Fecha de modificación	Tipo
.gitignore	26/04/2021 23:22	Documento de te...
.styleci.yml	26/04/2021 23:22	Archivo YML
artisan	26/04/2021 23:22	Archivo
composer.json	26/04/2021 23:22	Archivo JSON
composer.lock	27/04/2021 0:26	Archivo LOCK
package.json	26/04/2021 23:22	Archivo JSON
package-lock.json	26/04/2021 23:22	Archivo JSON
phpunit.xml	26/04/2021 23:22	Documento XML
Procfile	26/04/2021 23:22	Archivo
README.md	26/04/2021 23:22	Archivo MD
server.php	26/04/2021 23:22	JetBrains PhpStorm
webpack.mix.js	26/04/2021 23:22	JetBrains PhpStorm

**Fig. 11.** Creación del archivo “Procfile” en la raíz del proyecto.

Abrir el archivo “Procfile” en un editor de texto y agregar la siguiente línea, como se ilustra en la **Fig. 12**.



**Fig. 12.** Ruta public en archivo Procfile.

Luego se debe añadir el archivo creado con el comando *git add Procfile*, luego agregar un commit con un mensaje de lo que se está realizando, como se ilustra en la **Fig. 13**.

```
Jazmin@DESKTOP-J5G4QN2 MINGW64 ~/Documents/Tesis/2Repositorio/bk/20210426/tesis2
(master)
$ git add Procfile

Jazmin@DESKTOP-J5G4QN2 MINGW64 ~/Documents/Tesis/2Repositorio/bk/20210426/tesis2
(master)
$ git commit -m "Añadir archivo Procfile"
```

**Fig. 13.** Comandos para agregar el archivo Procfile en el repositorio.

Ahora se procede a crear una nueva aplicación en Heroku con el siguiente comando *heroku create* seguido del nombre de la aplicación, como se ilustra en la **Fig. 14**.

```
C:\Users\Jazmin\Documents\Tesis\2Repositorio\bk\20210426\tesis2>heroku create tesisjv
Creating @ tesisjv... !
```

**Fig. 14.** Comando para crear la aplicación en Heroku.

Posterior a ello, subir el código del repositorio al proyecto creado en Heroku con el comando *git push heroku master*, como se ilustra en la **Fig. 15**.

```
C:\Users\Jazmin\Documents\Tesis\2Repositorio\bk\20210426\tesis2>git push heroku master
```

**Fig. 15.** Comando para subir el código al proyecto en Heroku

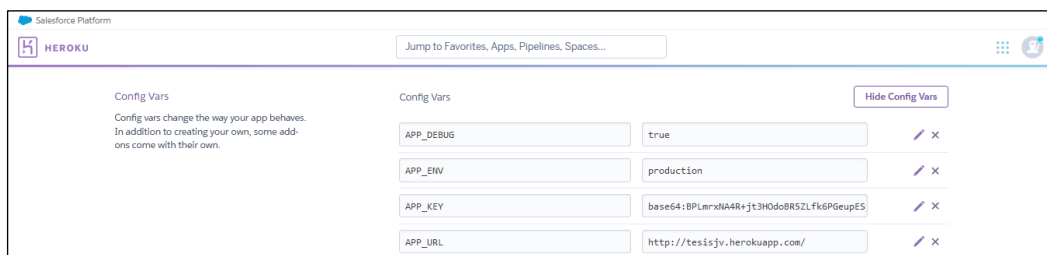
Se ingresa al proyecto en Heroku desde un navegador, posteriormente en la sección “*Config Vars*” ubicado en el apartado “*Settings*”, se coloca las variables obtenidas desde el archivo *.env* del proyecto, como se ilustra en las **Fig. 16** y **Fig. 17**.

```
.env: Bloc de notas
Archivo Edición Formato Ver Ayuda
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:BPLmrXNA4R+jt3HOdoBR5ZLfk6PGeupESy0yI41kg/4=
APP_DEBUG=true
APP_URL=http://tesis2.test

LOG_CHANNEL=stack
LOG_LEVEL=debug

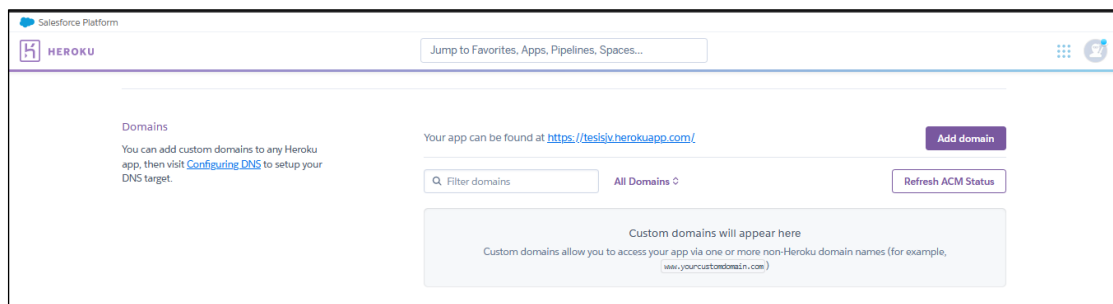
DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=vi_asistencia
DB_USERNAME=root
DB_PASSWORD=12345
```

**Fig. 16.** Archivo .env del proyecto.



**Fig. 17.** Credenciales del Archivo .env en el proyecto de Heroku.

Ahora en la sección de *Domains* se visualiza el enlace al proyecto en producción y finalmente se accede al sistema web por medio una URL, como se ilustran en las **Fig. 18** y **Fig. 19**.



**Fig. 18.** Enlace de acceso al proyecto en Heroku.

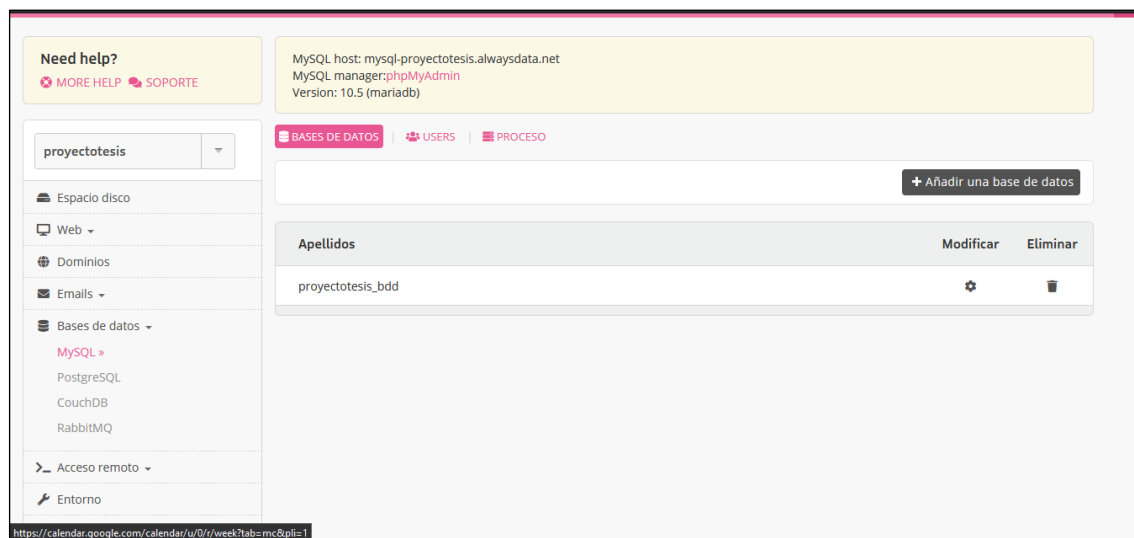




**Fig. 19.** Sistema web en producción.

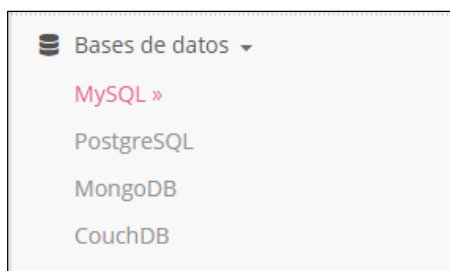
### 3. Conexión a la base de datos

Ingresar al panel de administración de alwaysdata, el cual permite alojar una base de datos, como se ilustra en la **Fig. 20**.



**Fig. 20.** Panel de administración de alwaysdata.

En el menú lateral izquierdo, escoger la opción bases de datos y seleccionar MySQL, como se ilustra en la **Fig. 21**.



**Fig. 21.** Menú de selección para la base de datos.

Hacer clic en el botón “Añadir una base de datos” y a continuación escribir el nombre de la base de datos, así como asignar los permisos que tiene el usuario de la cuenta, como se ilustra en la **Fig. 22**.

**Fig. 22.** Creación de la base de datos y permisos de usuario.

Dirigirse a la cuenta de Heroku en la cual está creada la aplicación llamada “tesisjv”, y luego ubicarse en la pestaña “Settings” apartado “Config Vars” y definir las credenciales de conexión; para esto se define el tipo de conexión de base de datos que se va a utilizar, host, puerto, nombre de la base de datos, usuario y contraseña obtenidos desde la cuenta de Alwaysdata, como se ilustra en la **Fig. 23**.

**Fig. 23.** Credenciales para la conexión de la base de datos.

Finalmente, en la terminal ejecutar el comando “heroku restart” para reiniciar el servidor y esperar hasta que se apliquen los cambios realizados y con ello, se establece la conexión directa entre el sistema web y la base de datos, como se ilustra en la **Fig. 24**.

```
C:\Users\Jazmin\Documents\Tesis\2Repositorio\bk\20210426\tesis2>heroku restart
Restarting dynos on  @ tesisjv... done
```

**Fig. 24.** Comando para reiniciar Heroku.

#### 4. Credenciales de acceso al sistema web

Para acceder al sistema web en producción, ingresar a la siguiente URL:

<http://tesisjv.herokuapp.com/>

- **Credenciales para el usuario Administrador**

Correo: [admin@gmail.com](mailto:admin@gmail.com)

Contraseña: admin

- **Credenciales para el usuario Inspector**

Correo: [inspector1@gmail.com](mailto:inspector1@gmail.com)

Contraseña: inspector-1

- **Credenciales para el usuario Profesor**

Correo: [profesor1@gmail.com](mailto:profesor1@gmail.com)

Contraseña: profesor-1

## 5. Repositorio del código fuente del sistema web

El código fuente de todo el proyecto, así como la documentación se encuentra alojado en un repositorio de GitHub, al cual se accede mediante los siguientes enlaces:

- **sistema web:** <https://github.com/victoralonsopacha/tesis2>
- **Documentación:** <https://github.com/victoralonsopacha/tesis2/tree/documentacion>