

ANEXO XX

Despliegue de aplicación web Java en Heroku

Paso 1: Se deberá crear una cuenta en la página de Heroku accediendo al siguiente link: <https://signup.heroku.com/dc>

Paso 2: Luego se deberá descargar e instalar la interfaz de línea de comandos de Heroku (Heroku CLI). En este caso se descargará e instalará la opción disponible para Windows disponible en el siguiente link: <https://devcenter.heroku.com/articles/getting-started-with-java#set-up>

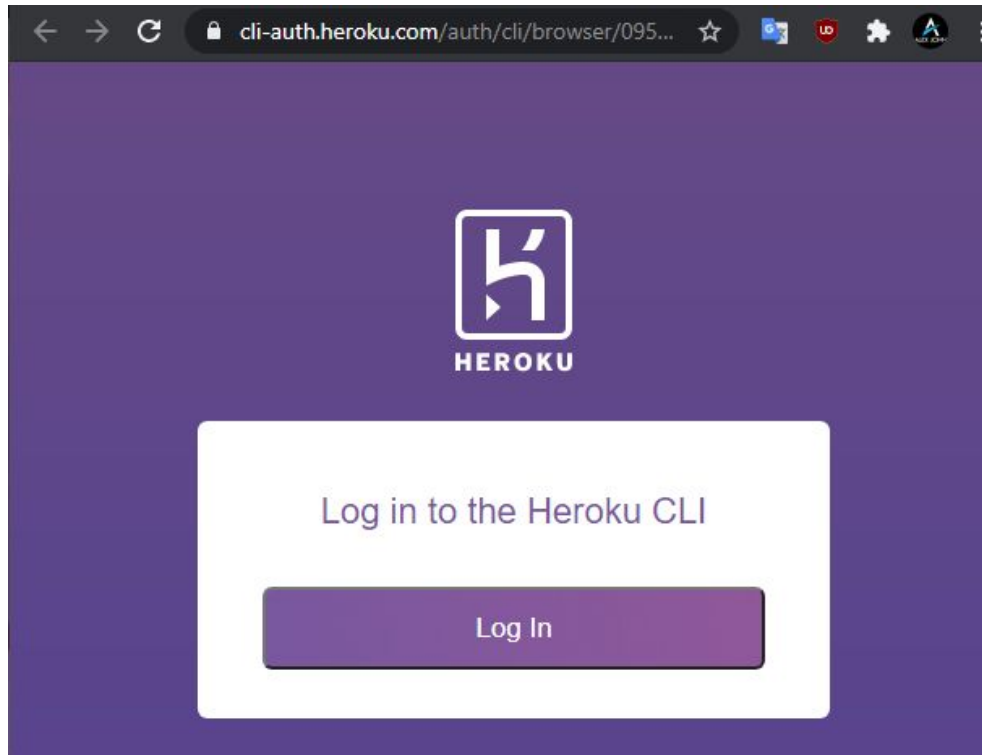


Fig.

Paso 3: Cuando se complete la instalación, puede usar los comandos de Heroku desde su terminal. En Windows, inicie el símbolo del sistema (cmd.exe) o Powershell para acceder al shell de comandos. Utilice el siguiente comando para iniciar sesión en la CLI de Heroku:

```
$ heroku login
heroku: Press any key to open up the browser to login or q to exit
> Warning: If browser does not open, visit
> https://cli-auth.heroku.com/auth/browser/**
heroku: Waiting for login...
Logging in... done
Logged in as me@example.com
```

Este comando abre su navegador web en la página de inicio de sesión de Heroku, simplemente haga clic en el botón Iniciar sesión que se muestra en dicha página.



Paso 4: Ahora nos dirigimos al directorio donde está nuestro proyecto y ejecutamos los siguientes comandos en caso de no haberlos ejecutado aún.

```
$ git add .
```

```
$ git commit -m "Demo"
```

Paso 5: Utilice el siguiente comando para instalar el complemento *heroku-cli-deploy*:

```
$ heroku plugins:install heroku-cli-deploy
```

Paso 6: En este paso, implementará la aplicación web en Heroku. Primero, creamos una aplicación en Heroku, la cual prepara a Heroku para recibir nuestro código fuente:

```
$ heroku create
Creating app... done, blooming-citadel-85625
https://blooming-citadel-85625.herokuapp.com/ | https://git.heroku.com/blooming-
```

Cuando se crea una aplicación, Heroku crea automáticamente un control remoto de Git (nombrado como heroku) y se asocia con nuestro repositorio de Git local. De forma predeterminada, Heroku genera un nombre aleatorio para su aplicación y su respectiva URL.

Paso 7: Ahora implementamos nuestro código fuente:

```
$ git push heroku master
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Java app detected
remote: -----> Installing JDK 1.8... done
remote: -----> Executing Maven
...
remote:      [INFO] -----
remote:      [INFO] BUILD SUCCESS
```

Paso 8: Crea el archivo WAR dando clic derecho en nuestro proyecto en Netbeans y luego damos clic en Build. Puede utilizar cualquier otro método para generar un archivo WAR. Puede usar Maven o simplemente exportar su aplicación desde su IDE como un archivo WAR. El único requisito es que el archivo WAR sea una aplicación web Java estándar y se adhiera a la estructura y las convenciones de la aplicación web estándar.

```
$ heroku war:deploy <path_to_war_file> --app <app_name>
Uploading my-app.war....
---> Packaging application...
- app: my-app
- including: webapp-runner.jar
- including: my-app.war
---> Creating build...
- file: slug.tgz
- size: 1MB
---> Uploading build...
- success
---> Deploying...
```

Paso 9: En caso de que el paso anterior presente un error utilice este comando y ejecute el comando del paso anterior, caso contrario omita este paso continúe con el siguiente.

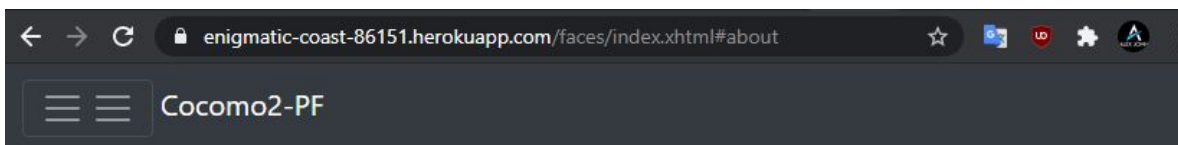
```
$ heroku buildpacks:clear
```

Paso 10: Luego de que la aplicación se implemente, asegúrese de que se esté ejecutando al menos una instancia de la aplicación con el siguiente comando:

```
$ heroku ps:scale web=1
```

Paso 11: Ahora visite su aplicación web alojada en la nube mediante la URL generada anteriormente utilizando un navegador web. Como un atajo útil, puede abrir el sitio web con este comando:

```
$ heroku open
```



COCOMO II: Puntos de función.

La presente aplicación web
realiza el cálculo de los puntos de función,
Para ello hay que llenar los datos solicitados en los pasos 1 y 2.

PASO 1:					
Parámetros de medición	Simple	Promedio	Complejo	Total	
Cantidad de entradas del usuario	<input type="text"/> x 3 +	<input type="text"/> x 4 +	<input type="text"/> x 6 =	<input type="text"/>	6

Paso 12: Si se desea ver los registros de nuestra aplicación desplegada ordenados por tiempo se deberá ejecutar el siguiente comando:

```
$ heroku logs --tail
2020-05-20T09:02:00.923447+00:00 app[web.1]: 2020-05-20 09:02:00.923 INFO 4
2020-05-20T09:02:00.923662+00:00 app[web.1]: 2020-05-20 09:02:00.923 INFO 4
2020-05-20T09:02:01.994084+00:00 app[web.1]: 2020-05-20 09:02:01.991 INFO 4
2020-05-20T09:02:02.726829+00:00 app[web.1]: 2020-05-20 09:02:02.726 INFO 4
2020-05-20T09:02:03.093455+00:00 app[web.1]: 2020-05-20 09:02:03.093 INFO 4
2020-05-20T09:02:03.578841+00:00 app[web.1]: 2020-05-20 09:02:03.578 INFO 4
```

Visite su aplicación en un navegador web para generar otro mensaje de registro.

Adicionalmente, cada vez que se desee subir cambios de nuestro código fuente a Heroku simplemente se deberá ejecutar únicamente los siguientes comandos:

```
git add .
git commit -am "Update"
git push heroku master
heroku buildpacks:clear
heroku war:deploy <file.war> --app <name-app-heroku>
heroku open
```