

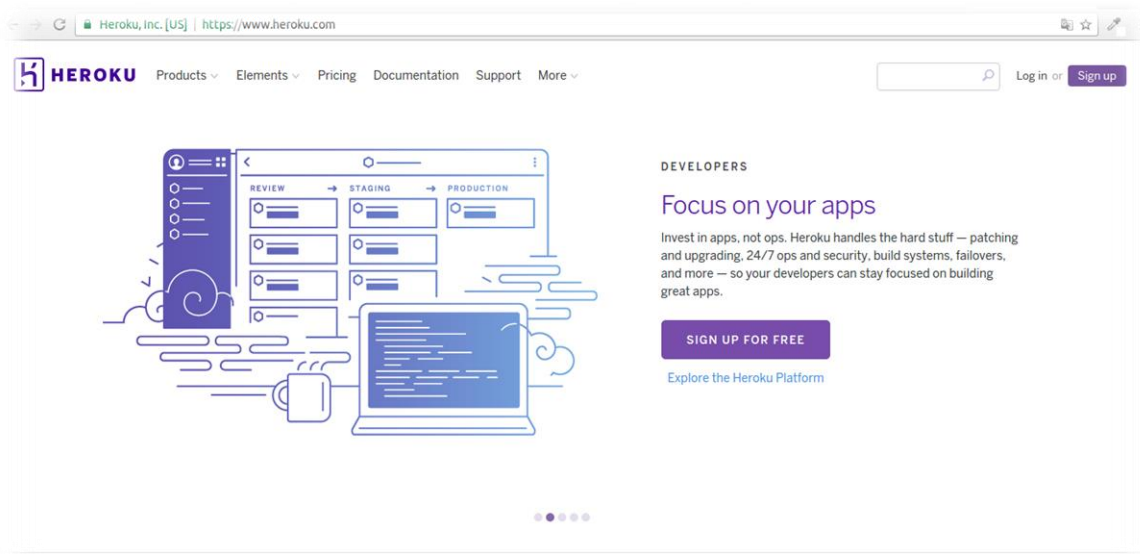
COMO PUBLICAR UN SITIO WEB CON HEROKU

Heroku es un servicio en la nube que nos permite desarrollar y alojar nuestras aplicaciones, acá encontraremos disponibles los siguientes lenguajes de programación: Ruby, Java, Node.js, Scala, Clojure y Python y PHP.

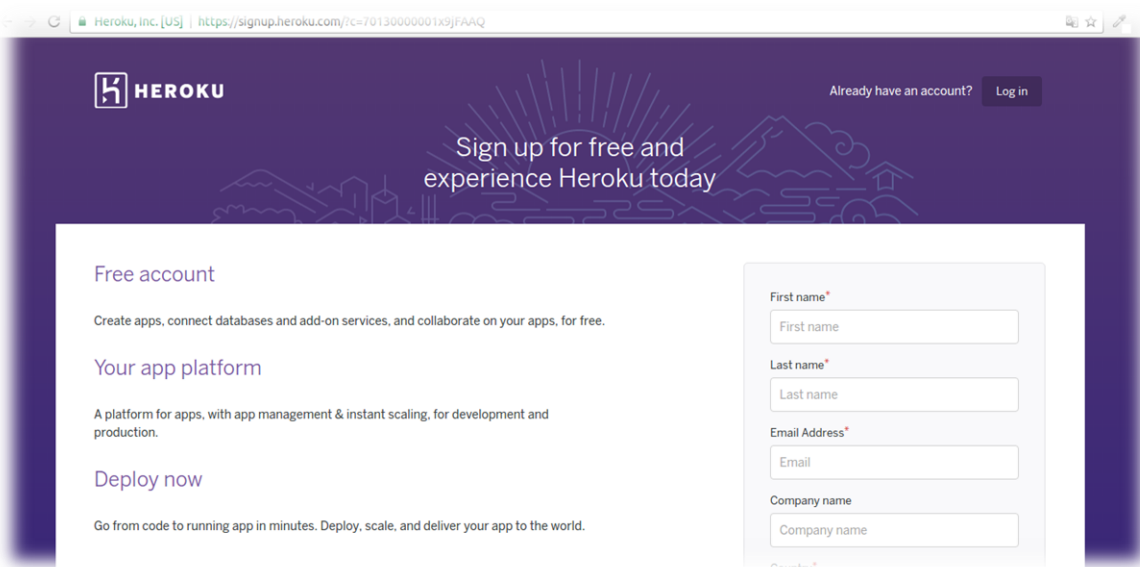
Hoy se indicará como publicar un sitio web con Node.js, subido a GitHub y Ubuntu como SO, para ello se debe realizar uno a uno los siguientes pasos:

Configuración de Heroku

Paso 1: luego de creado localmente el sitio Web, vamos a dirigirnos a <https://heroku.com>.



Paso 2: a continuación creamos una cuenta con nuestros datos y confirmamos la cuenta a través nuestro correo electrónico.



Paso 3: posteriormente pasaremos a instalar Heroku Command Line Interface (CLI) mejor conocido como Heroku CLI en Ubuntu (si tienes otro SO sigue esta pagina <https://devcenter.heroku.com/articles/getting-started-with-nodejs#set-up>)

- 3.1** Como requisito previo debemos tener instalado Ruby, Node, Npm y Git, si no lo tenemos instalado nos dirigimos al terminal e ingresamos:

```
$ sudo apt install programa
```

- 3.2** Después de haber instalado los requisitos previos ingresamos el siguiente comando e iniciara la instalación:

```
$ wget -O-https://toolbelt.heroku.com/install-ubuntu.sh | sh
```

```
Yenifer@ubuntu:~/Documents/Node/09_production_ready$ wget -O- https://toolbelt.heroku.com/install-ubuntu.sh | sh
2016-11-07 18:10:56-- https://toolbelt.heroku.com/install-ubuntu.sh
Resolviendo toolbelt.heroku.com (toolbelt.heroku.com)... 54.243.84.197, 107.20.254.176, 50.16.189.161
Conectando con toolbelt.heroku.com (toolbelt.heroku.com)[54.243.84.197]:443... conectado.
Envío HTTP enviado, esperando respuesta... 200 OK
Longitud: 719 [text/plain]
Guardando a: "STDOUT"

100%[=====] 719 --KB/s in 0s

2016-11-07 18:10:57 (55,8 MB/s) - escritos a stdout [719/719]

This script requires superuser access to install apt packages.
You will be prompted for your password by sudo.
[sudo] password for yeniferr:
2016-11-07 18:11:02-- https://toolbelt.heroku.com/apt/release.key
Resolviendo toolbelt.heroku.com (toolbelt.heroku.com)... 54.243.84.197, 107.20.254.176, 50.16.189.161
Conectando con toolbelt.heroku.com (toolbelt.heroku.com)[54.243.84.197]:443... conectado.
Envío HTTP enviado, esperando respuesta... 200 OK
Longitud: 1737 (1,7K) [application/octet-stream]
Guardando a: "STDOUT"

100%[=====] 1,70K --KB/s in 0s

2016-11-07 18:11:04 (146 MB/s) - escritos a stdout [1737/1737]

$
$
$1 http://downloads-distroweb.org/repo/ubuntu-upstart dist InRelease
$2 http://security.ubuntu.com/ubuntu xenial-security InRelease
$3 http://archive.ubuntu.com/ubuntu xenial InRelease
$4 http://ve.archive.ubuntu.com/ubuntu xenial InRelease
$5 http://downloads-distroweb.org/repo/ubuntu-upstart dist InRelease
$6 http://ve.archive.ubuntu.com/ubuntu xenial-updates InRelease [95,7 kB]
$7 http://deb.nodesource.com/node_6.x xenial InRelease
$8 http://toolbelt.heroku.com/ubuntu ./ InRelease
$9 http://toolbelt.heroku.com/ubuntu ./ Release [1.609 B]
$10 http://ve.archive.ubuntu.com/ubuntu xenial-backports InRelease [92,2 kB]
$11 http://toolbelt.heroku.com/ubuntu ./ Release.gpg [473 B]
$12 http://toolbelt.heroku.com/ubuntu ./ Packages [725 B]
$13 http://toolbelt.heroku.com/ubuntu ./ Packages [725 B]
$14 [Esperando las cabeceras]
```

- 3.3** Ahora ingresaremos en la consola el siguiente comando y cuando no los solicite ingresamos el correo con que nos registramos en heroku y la contraseña:

```
$ heroku login
```

Paso 4: a continuación procedemos a clonar nuestra aplicación alojada en nuestro repositorio de GitHub, para este ejemplo usaremos un repositorio de prueba brindado por Heroku, que contiene una sencilla aplicación con node, luego ingresaremos a la carpeta clonada:

```
$ git clone https://github.com/heroku/node-js-getting-started.git
$ cd node-js-getting-started
```

Paso 5: luego de acceder a nuestro repositorio localmente vamos a crear nuestra aplicación heroku con el siguiente comando:

```
$ heroku create
```

```
comprobando la conectividad... hecho.  
yeniferr@yenifer:~/Documentos/Node/heroku$ cd node-js-getting-started  
yeniferr@yenifer:~/Documentos/Node/heroku/node-js-getting-started$ heroku create  
creating app... done, fierce-gorge-83083  
https://fierce-gorge-83083.herokuapp.com/ | https://git.heroku.com/fierce-gorge-83083.git  
yeniferr@yenifer:~/Documentos/Node/heroku/node-js-getting-started$  
yeniferr@yenifer:~/Documentos/Node/heroku/node-js-getting-started$
```

Cuando se crea una aplicación, un control remoto git (llamado también heroku) se crea y se asocia con nuestro repositorio git local.

Como se puede observar en la imagen Heroku genera un nombre aleatorio (en este caso fierce-gorge-83083) para nuestra aplicación (lo podemos cambiar según sea nuestro agrado).

Paso 6: ahora desde nuestra terminal cargamos y verificamos nuestro código con el siguiente comando:

```
$ git push heroku master
```

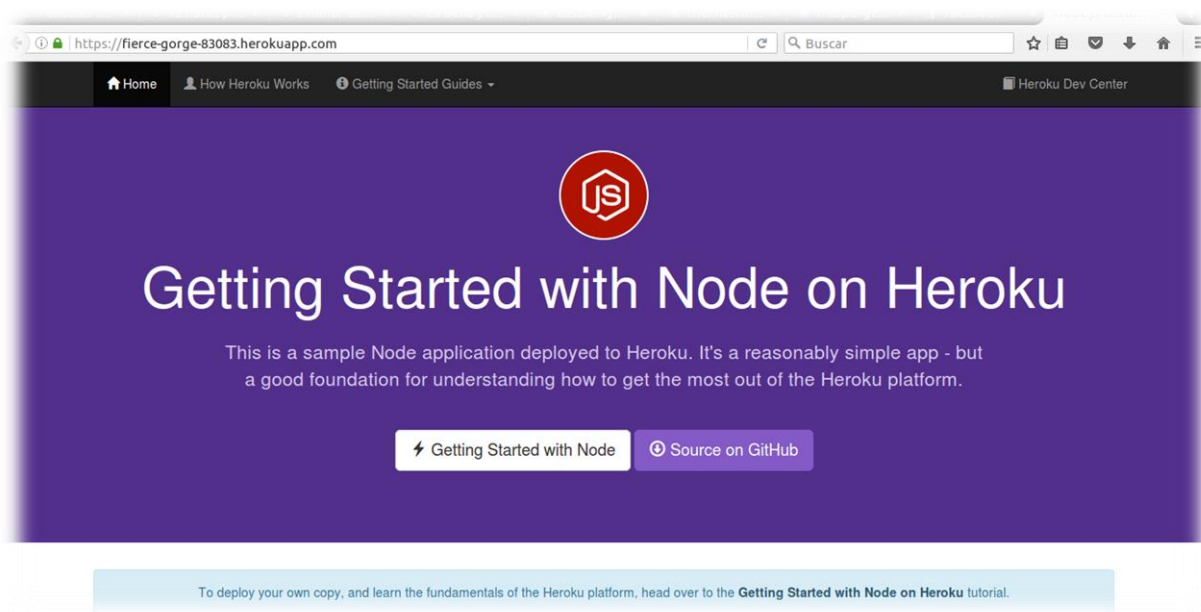
```
yeniferr@yenifer:~/Documentos/Node/heroku/node-js-getting-started$ git push heroku master  
Counting objects: 456, done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (353/353), done.  
Writing objects: 100% (456/456), 226.29 KiB | 0 bytes/s, done.  
Total 456 (delta 69), reused 456 (delta 69)  
remote: compressing source files... done.  
remote: Building source:  
remote:  
remote: ----> Node.js app detected  
remote:  
remote: ----> Creating runtime environment  
remote:  
remote:       NPM_CONFIG_LOGLEVEL=error  
remote:       NPM_CONFIG_PRODUCTION=true  
remote:       NODE_ENV=production  
remote:       NODE_MODULES_CACHE=true  
remote:  
remote: ----> Installing binaries  
remote:       engines.node (package.json): 5.9.1  
remote:       engines.npm (package.json): unspecified (use default)  
remote:  
remote:       Downloading and installing node 5.9.1...  
remote:       Using default npm version: 3.7.3  
remote:  
remote: ----> Restoring cache  
remote:       Skipping cache restore (new runtime signature)  
remote:  
remote: ----> Building dependencies  
remote:       Installing node modules (package.json)  
remote:       node-js-getting-started@0.2.5 /tmp/build_7c7b002d98de90bc4bf9337c80ce4acc  
remote:       ├── ejs@2.4.1  
remote:       ├── express@4.13.3  
remote:       ├── accepts@1.2.13  
remote:       │   ├── mime-types@2.1.12  
remote:       │   └── mime-db@1.24.0  
remote:       ├── negotiator@0.5.3  
remote:       ├── array-flatten@1.1.1  
remote:       ├── content-disposition@0.5.0  
remote:       └── content-type@1.0.2
```

Nota: después de que finalice correctamente el paso 6, debemos asegurarnos de que al menos una instancia de la aplicación se está ejecutando, tecleando el siguiente comando en consola:

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

Paso 7: ahora visitamos la aplicación en la URL generada por el nombre de nuestra aplicación. De manera sencilla, podemos abrir la página web de la siguiente manera:

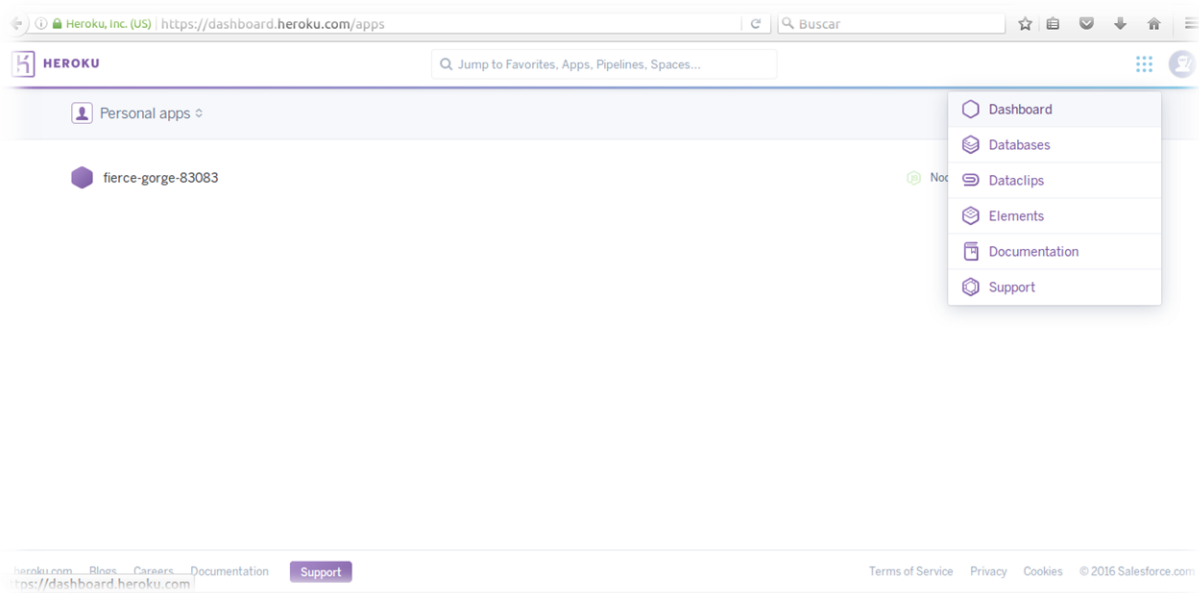
```
$ Heroku open
```



Con esto podremos visualizar nuestro sitio web desde cualquier navegador a través del url generado, ya alojado en heroku.

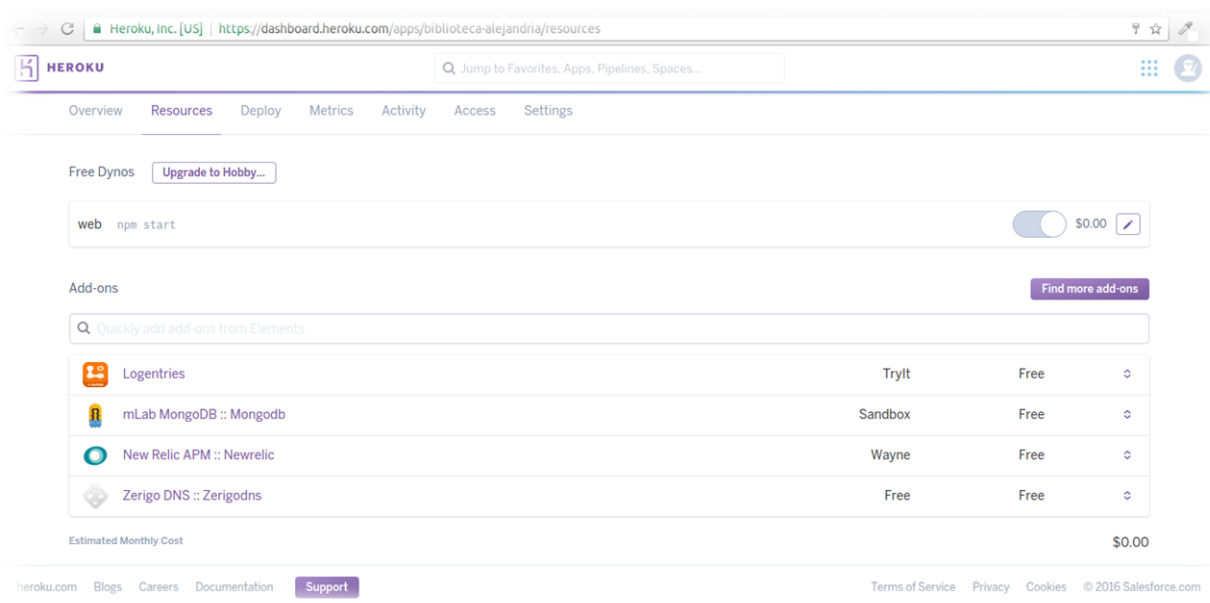
Configuraciones extras de Heroku

Paso1: ahora manejaremos todo desde nuestros dashboard de nuestra cuenta web en heroku, donde se mostraran los sitios web activos.

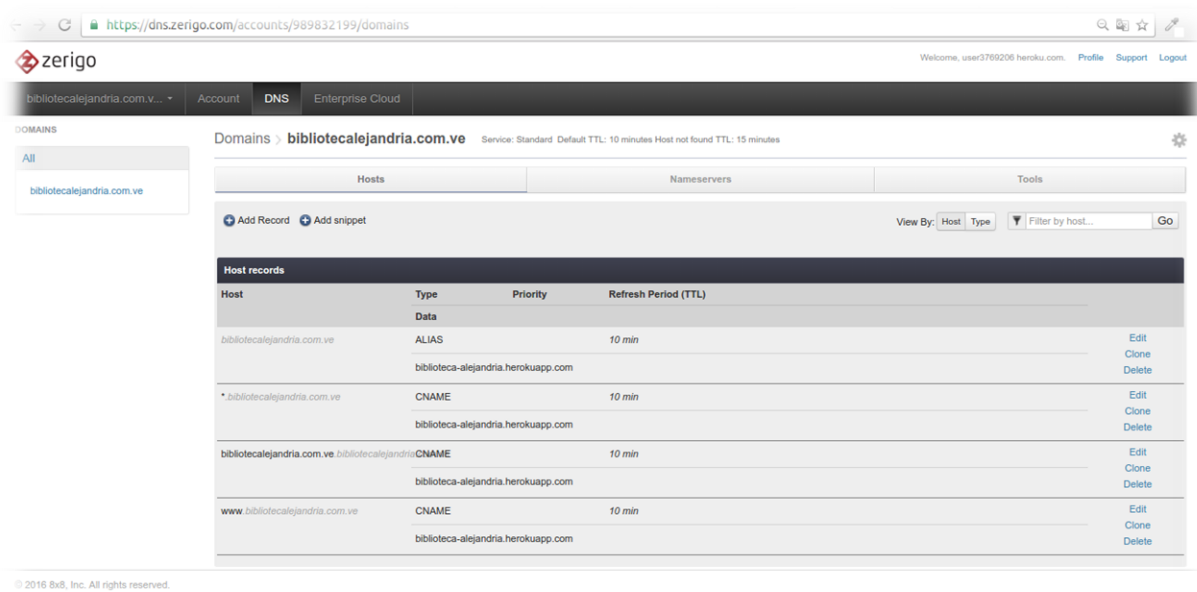


Paso 2: seleccionamos nuestra aplicación y a continuación nos dirigimos a la pestaña de Resources, agregaremos los siguientes recursos (es solo una recomendación):

- 2.1 *Logentries*: analiza sus datos de registro.
- 2.2 *mLab MongoDB*: recurso que permite importar una base de datos de mongo a nuestro sitio web.
- 2.3 *newrelic*: chequea nuestra estructura web y funcionamiento
- 2.4 *Zerigo DNS :: Zerigodns*: genera los DNS para las configuraciones de los dominios.



Paso 3: luego de agregarlos vamos a ingresar al recurso de Zerigo DNS y configuraremos los host y nameservers como se indican en las imágenes:



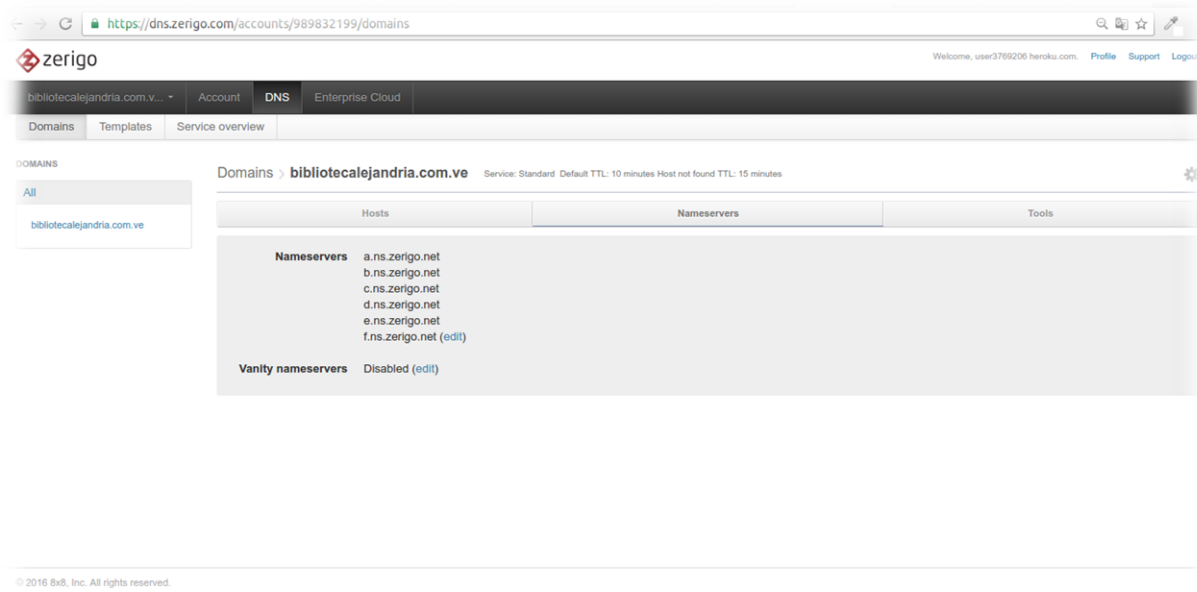
Domains > **bibliotecalejandria.com.ve** Service: Standard Default TTL: 10 minutes Host not found TTL: 15 minutes

Hosts Nameservers Tools

+ Add Record + Add snippet View By: Host Type Filter by host... Go

Host	Type	Priority	Refresh Period (TTL)	
Host records				
Data				
bibliotecalejandria.com.ve	ALIAS		10 min	Edit Clone Delete
biblioteca-alejandria.herokuapp.com				
*.bibliotecalejandria.com.ve	CNAME		10 min	Edit Clone Delete
biblioteca-alejandria.herokuapp.com				
bibliotecalejandria.com.ve bibliotecalejandria	CNAME		10 min	Edit Clone Delete
biblioteca-alejandria.herokuapp.com				
www.bibliotecalejandria.com.ve	CNAME		10 min	Edit Clone Delete
biblioteca-alejandria.herokuapp.com				

© 2016 Bx8, Inc. All rights reserved.



Domains > **bibliotecalejandria.com.ve** Service: Standard Default TTL: 10 minutes Host not found TTL: 15 minutes

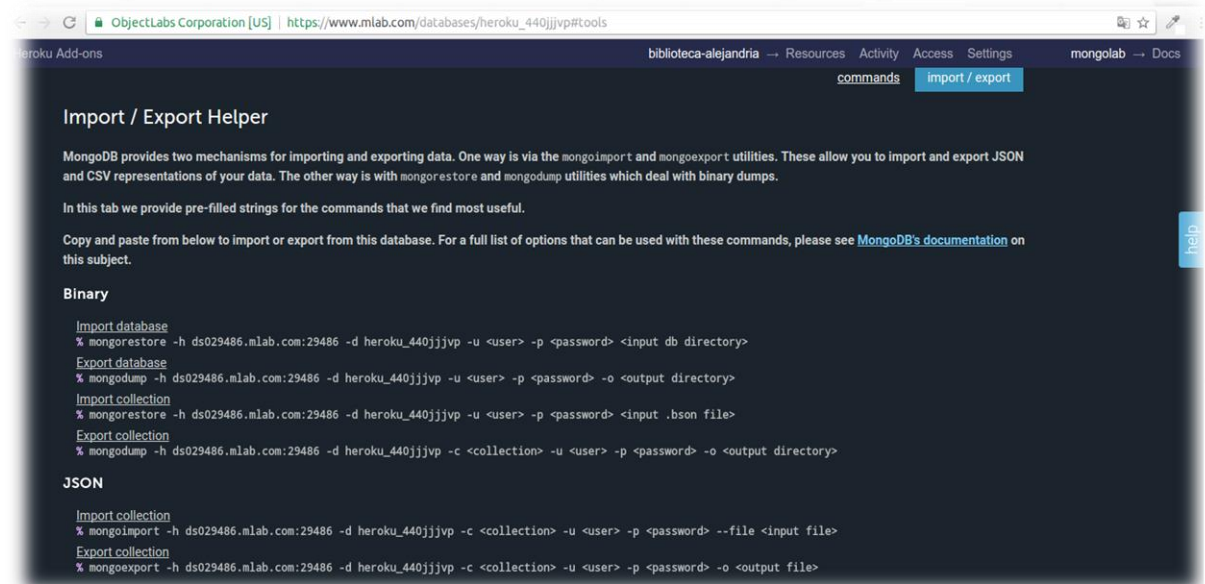
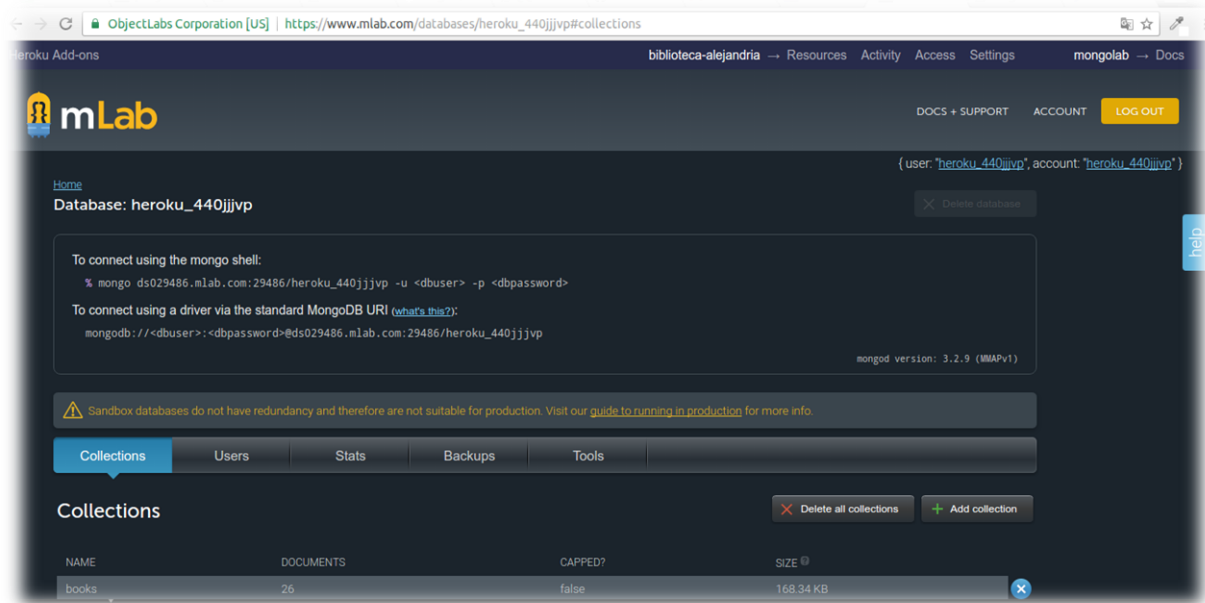
Hosts Nameservers Tools

Nameservers a.ns.zerigo.net
b.ns.zerigo.net
c.ns.zerigo.net
d.ns.zerigo.net
e.ns.zerigo.net
f.ns.zerigo.net (edit)

Vanity nameservers Disabled (edit)

© 2016 Bx8, Inc. All rights reserved.

Paso 4: luego iremos al recurso de MongoDB y crearemos nuestra base de datos o importaremos según sea el gusto:

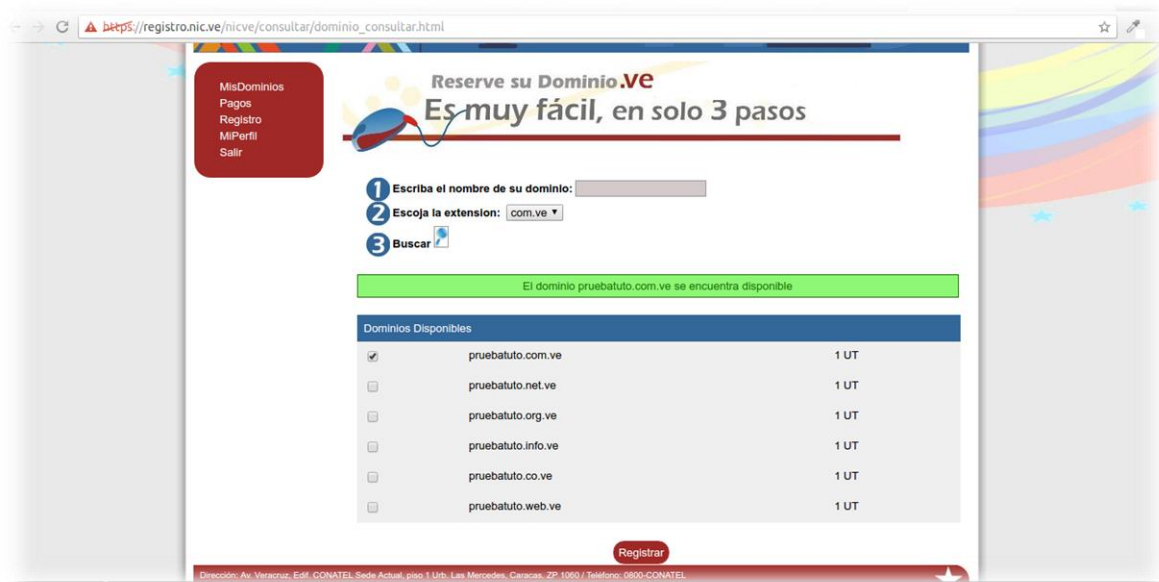


Reserva de Dominio y configuración de Heroku

Si deseamos configurar nuestra aplicación con un Dominio reservado debemos hacer lo siguiente:

Paso1: creamos una cuenta en la reservación de dominios de nuestro gusto para este tutorial usaremos <https://registro.nic.ve>

Paso 2: al registrarnos e ingresar en nuestra cuenta nos iremos a registros y buscaremos y registraremos el dominio deseado:



Paso 3: iremos a la pestaña de pago y realizaremos nuestro pago del dominio:



Nota: por cada proceso que realicemos, registro.nic nos enviara un correo para la confirmación.

Paso 4: luego de procesado nuestro pago veremos en la pestaña de MisDominios el dominio registrado, lo seleccionamos y nos vamos a Configurar DNS.

https://registro.nic.ve/nicve/consultar/lista_dominios.html

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para la Comunicación y la Información | Comisión Nacional de Telecomunicaciones

Centro de Información de Red

Lista de Dominios

Tiene un total de **3** dominios registrados, si no es la información correcta haga click aquí para actualizar esta lista.
Obtenga aquí la lista en formato CSV

Paginar resultados... No paginar...

Nombre de Dominio	Fecha de Registro	Fecha de Vencimiento
✓	07/09/2016	07/09/2017
⚠	24/10/2016	24/10/2017
✓	09/06/2016	09/06/2017

Dirección: Av. Venezuela, Edif. CONATEL Sede Actual, piso 1 Urb. Las Mercedes, Caracas, ZP 1060 / Teléfono: 0800-CONATEL
Correo de soporte técnico: soporte@conatel.gob.ve

https://registro.nic.ve/nicve/consultar/detalle_dominio.html?dom_id=4798906&id_usu=79838

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para la Comunicación y la Información | Comisión Nacional de Telecomunicaciones

Centro de Información de Red

Información de Dominio: bibliotecalejandria.com.ve

Contatos

Ingrese aquí para cambiar la información de contatos asociada al dominio "bibliotecalejandria.com.ve".

Contatos

DNS

Si su(s) servidor(es) de DNS pertenece(n) al dominio "bibliotecalejandria.com.ve" utilice esta opción para crearlos y mantenerlos.

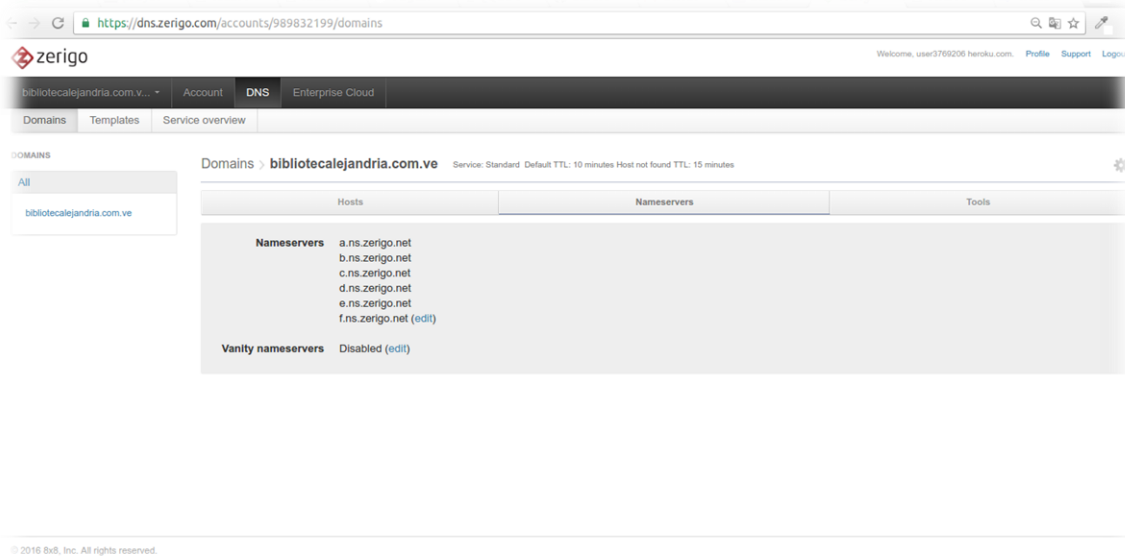
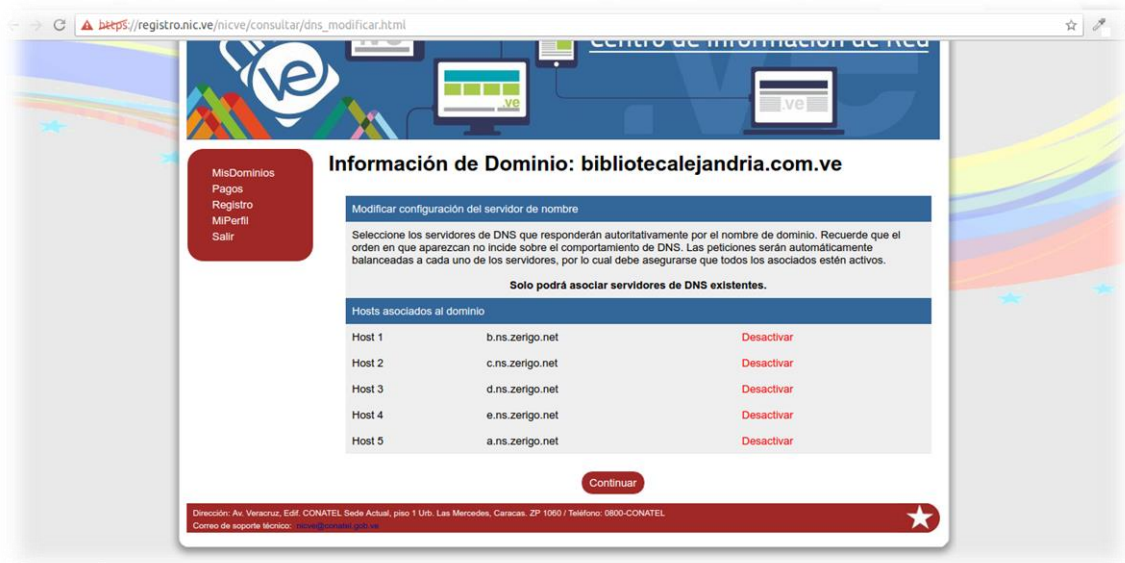
[Registrar/Editar servidores de nombre](#)

Haga clic en esta opción para especificar que servidores responderán autoritativamente para el dominio "bibliotecalejandria.com.ve".

[Configurar DNS](#)

Dirección: Av. Venezuela, Edif. CONATEL Sede Actual, piso 1 Urb. Las Mercedes, Caracas, ZP 1060 / Teléfono: 0800-CONATEL
Correo de soporte técnico: soporte@conatel.gob.ve

Paso 5: acá agregaremos tal cual nos indican en ZerigoDns nuestros nameservers:



Si todos los pasos fueron exitosos podemos acceder a nuestro sitio web en línea a través de nuestro dominio.