

Diplomatura en programación web full stack con React JS

Módulo 6:

Administración, API Rest y despliegue

Unidad 4:

Despliegue



Presentación

En esta unidad finalmente desplegamos nuestro sitio en la plataforma Heroku y de esta forma podremos compartir nuestra aplicación con el mundo.



Objetivos

Que los participantes logren...

- Conocer qué es y para qué sirve Heroku.
- Hacer un despliegue exitoso de una aplicación escrita en Node.js.
- Hacer un despliegue exitoso de una aplicación hecha con React.



Bloques temáticos

1. Despliegue del sitio en Heroku.

1. Despliegue del sitio en Heroku

Heroku es uno de los PaaS (Platform as a Service) más utilizados en la actualidad en entornos empresariales por su fuerte enfoque en resolver el despliegue de una aplicación. Además te permite manejar los servidores y sus configuraciones, escalamiento y la administración.

Podemos ver su sitio, servicios y costos en <https://www.heroku.com/>. En nuestro caso utilizaremos todas opciones gratuitas, pero en caso de necesitar más almacenamiento, poder de procesamiento u otros servicios, siempre podemos escalar dentro de las opciones pagas.

Cuales son los requisitos para poder subir los archivos:

- Tener una **cuenta**
- Asegurarse de que el proyecto este en un **repositorio git local**.
- **Descargar la siguiente herramienta:**
<https://devcenter.heroku.com/articles/heroku-cli#download-and-install>.

Despliegue del backend

Paso 1

Escribir `heroku login` en la consola. En este paso no importa el path o ruta donde escribimos el comando, ya que el proceso de login será válido en toda nuestra computadora.

Paso 2

En el directorio de la aplicación escribimos `heroku create`. Este comando registra nuestra aplicación en el servicio de Heroku y le asigna un nombre al azar. Además agrega como remote de git el servidor que corresponda a nuestra aplicación.

Paso 3

Para subir nuestra aplicación, debemos asegurarnos de tener todo commiteado localmente y luego hacer el push al repositorio que Heroku nos agregó. Para eso usamos el comando `git push heroku master`.

Al finalizar el proceso, en caso de ser exitoso, nos mostrará un link a nuestra aplicación

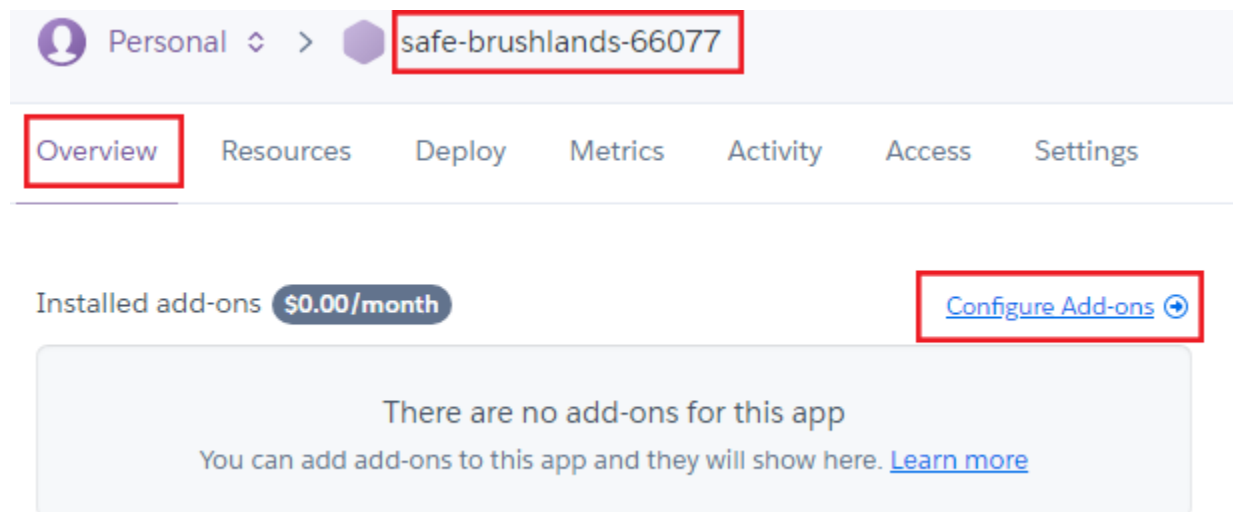
Pasos para trabajar con la BD

Para implementar bases de datos **MySQL** en Heroku vamos a utilizar uno de los tantos plugins que ofrece. Para agregar cualquiera de ellos a nuestro servicio será necesario tener validada una tarjeta de crédito dentro de los datos de perfil.




No debemos preocuparnos, ya que mientras nos mantengamos dentro de los límites que el plan gratis nos define no recibiremos cargo alguno en esta tarjeta.

Paso 1

Dentro del panel de administración de Heroku buscamos nuestra aplicación creada anteriormente, en este caso (safe-brushlands-66077), y hacemos click en la opción **configure Add-ons**.



En el buscador escribimos mysql y seleccionamos el servicio de **ClearDB MySQL**.

 >  safe-brushlands-66077  [Open app](#) [More](#)


[Overview](#) [Resources](#) [Deploy](#) [Metrics](#) [Activity](#) [Access](#) [Settings](#)


Dynos


This app has no process types yet

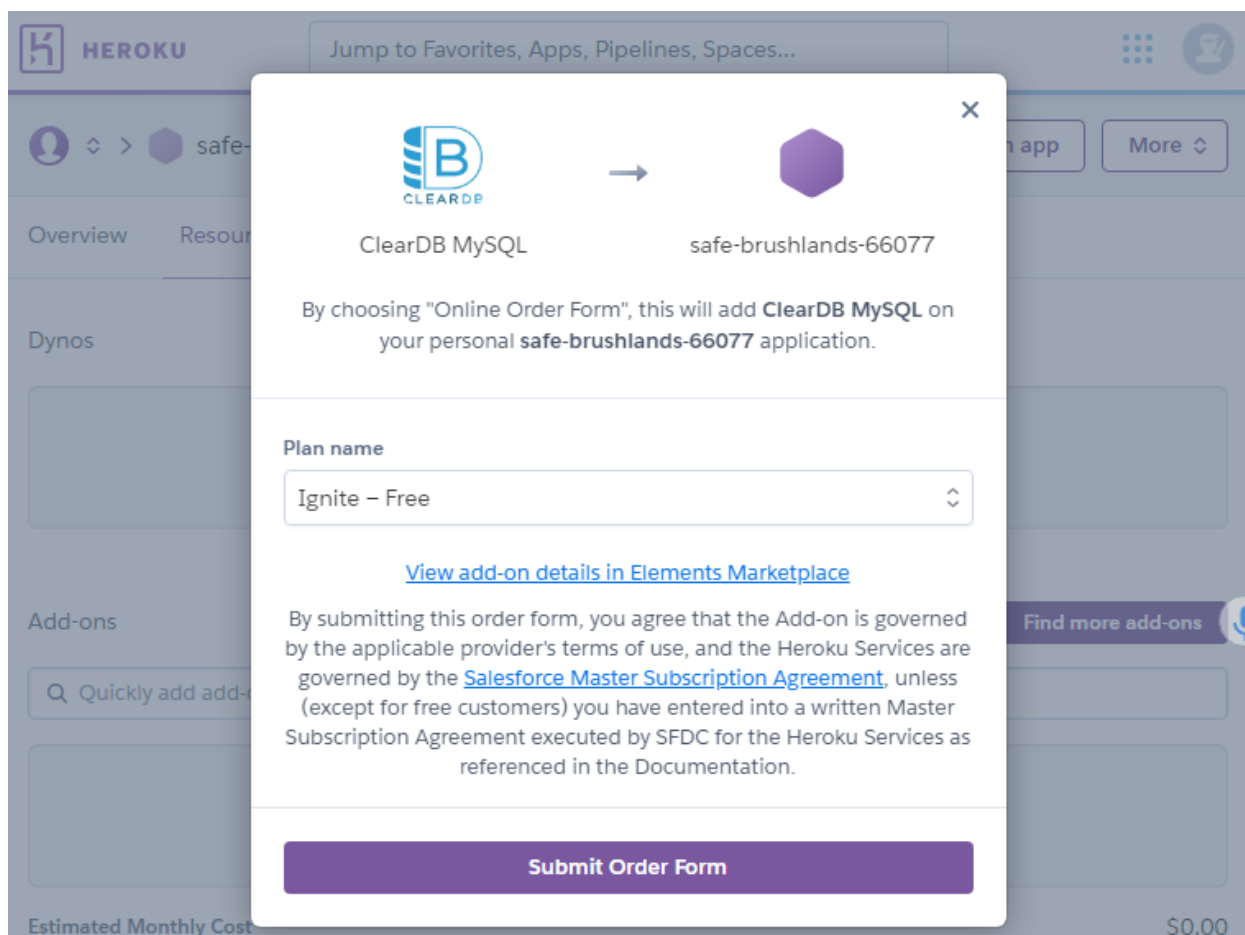
Add a Procfile to your app in order to define its process types. [Learn more](#)

Add-ons

[Find more add-ons](#) 

 **ClearDB MySQL**

 Draxlr for Postgres / MySQL



Paso 2

Vamos a configurar las variables de entorno de nuestra aplicación para que se conecte a la base de datos que acabamos de dar de alta. Para esto vamos a la opción **Settings** y clickeando en Reveal Config Vars veremos la ruta a la base de datos, la cual deberemos descomponer en valores que igualen a los que usabamos en nuestro archivo .env local.

Overview

Resources

Deploy

Metrics

Activity

Access

Settings

App Information

App Name

safe-brushlands-66077

Region  United States

Stack **heroku-18** [Upgrade Stack](#)

Framework No framework detected

Slug size No slug detected

Heroku git URL <https://git.heroku.com/safe-brushlands-66077.git>

Config Vars








[Reveal Config Vars](#)

El valor que nos mostrará tendrá el siguiente formato y deberemos ir extrayendo cada uno de los valores que necesitamos y asignarlo a una de las variables de entorno que nuestra aplicación espera.

`mysql://bbad738eoc:b5a50d@us-cdbr-east-02.cleardb.com/heroku_a32a4f251fec?reconnect=true`

`mysql://user:password@host/database`

Config Vars Hide Config Vars

CLEARDB_DATABASE_URL	mysql://bbad738e0c3103:b5a500ad@us-cdbr-e	 
MYSQL_DB_NAME	heroku_a32a4f251fec52d	 
MYSQL_HOST	us-cdbr-east-02.cleardb.com	 
MYSQL_PASSWORD	b5a50dao	 
MYSQL_USER	bbad738e0c3130	 

Add

Paso 3

Con el siguiente código vamos a poder migrar los datos de nuestra base de datos local a la base de datos que acabamos de agregar a Heroku. El código lo podemos bajar desde esta URL:

<https://medium.com/@michaeltendossemwanga/import-mysql-database-to-heroku-with-one-command-import-db-sql-a932d720c82b>

Copiamos el código y lo guardamos en un archivo con el nombre: **import.bat** al mismo nivel donde está el archivo app.js. Cabe destacar que esta opción sirve solo para Windows.



```
@echo off
heroku config | findstr CLEARDB > config.txt
set /p url=<config.txt

set "string=%url:?" & set "x=%"
set "x=%string:/" & set "dbname=%"
echo DB name: %dbname%
echo DB name: %dbname% >> config.txt

set "x=%string:@=" & set "substring=%"
set "host=%substring:/" & set "x=%"
echo Host: %host%
echo Host: %host% >> config.txt

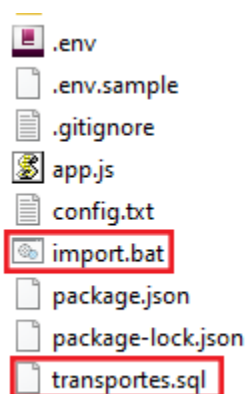
set "x=%string:==" & set "substring=%"
set "password=%substring:@=" & set "x=%"
echo Password: %password%
echo Password: %password% >> config.txt

set "x=%string:://" & set "substring=%"
set "user=%substring:==" & set "x=%"
echo User: %user%
echo User: %user% >> config.txt

mysql -u %user% -p%password% -h %host% -D %dbname% < %1
```

Paso 4

Exportamos la base de datos desde phpMyAdmin, y la guardamos como **transportes.sql** junto con el archivo **import.bat**. Finalmente en la consola escribimos el siguiente comando: **import transportes.sql**. Una vez terminado, todos los datos que teníamos en nuestra base de datos local habrán sido copiados al servidor MySQL remoto.



Despliegue del frontend

Paso 1

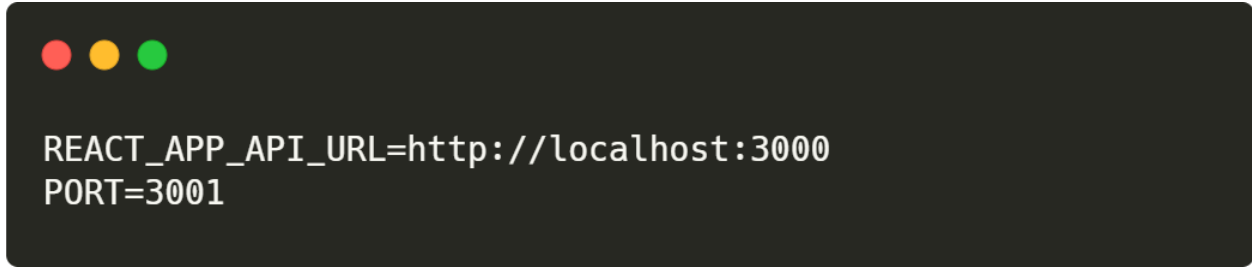
Instalamos **env-cmd** como dependencia de desarrollo con el comando "npm i -D env-cmd". Esta librería nos permitirá usar variables de entorno al momento de compilar nuestra aplicación.



```
npm i -D env-cmd
```

Paso 2

Crear archivo **.env** en el raíz de la aplicación con este contenido.

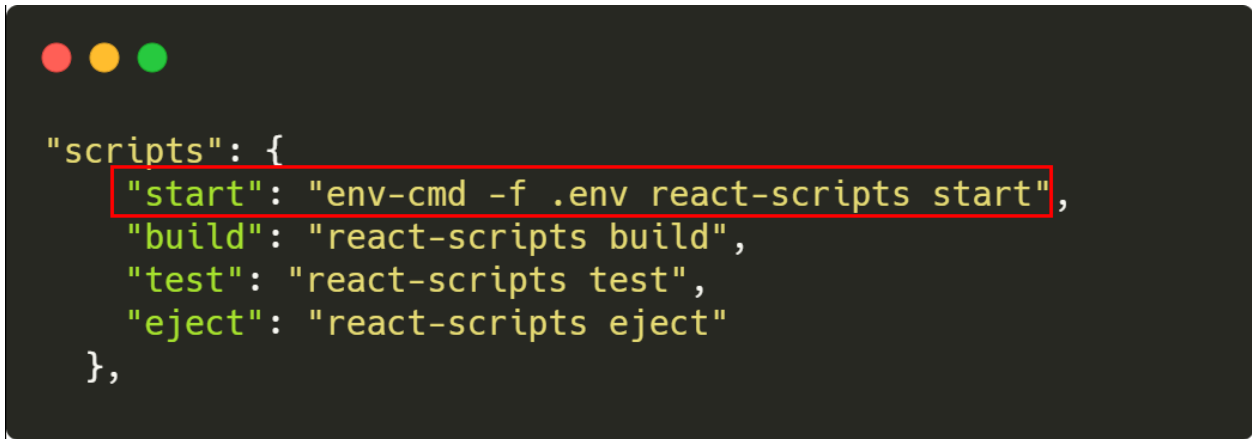


```
REACT_APP_API_URL=http://localhost:3000  
PORT=3001
```

La variable **REACT_APP_API_URL** contendrá la ruta donde se encuentre alojada la API que vamos a consumir. La variable **PORT** nos permite asegurarnos que nuestra aplicación se ejecute siempre en el puerto **3001**. Cabe destacar que estas variables en este archivo en particular solo tiene impacto en la computadora donde estamos desarrollando la aplicación.

Paso 3

Modificar el script start de **package.json** con el siguiente contenido para que mediante el uso de **env-cmd** incorpore los valores del archivo **.env** al momento de compilar.



```
"scripts": {  
  "start": "env-cmd -f .env react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject"  
},
```

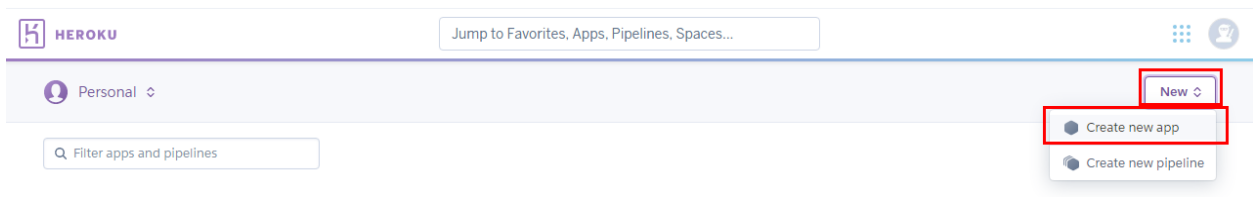
Paso 4

Ahora que tenemos nuestra variable disponible la utilizamos como parte de la url en **NovedadesPage.js**.

```
const cargarNovedades = async () => {  
  setLoading(true);  
  const response = await axios.get(`${process.env.REACT_APP_API_URL}/api/novedades`);  
  setNovedades(response.data);  
  setLoading(false);  
};
```

Paso 5

Creamos una nueva app en Heroku. Esta vez lo vamos a hacer desde el dashboard en vez de utilizar las herramientas de consola.



The screenshot shows the Heroku dashboard. At the top, there's a navigation bar with the Heroku logo, a search bar, and a 'New' button. Below the navigation bar, there's a 'Personal' section with a search bar and a list of apps. A red box highlights the 'New' button, and another red box highlights the 'Create new app' option in the dropdown menu.

Create New App

App name

transportesreact

transportesreact is available

Paso 6

Heroku no está preparado de forma nativa para soportar aplicaciones creadas mediante Create React App. Para suplir esta falencia podemos utilizar un buildpack personalizado que se encargue de compilar y desplegar nuestra aplicación.

Los buildpacks en Heroku son los encargados de instalar y configurar las aplicaciones que subimos. Podemos confiar en que Heroku detecte correctamente nuestra aplicación como en la mayoría de los casos o utilizar un buildpack de terceros como en este caso.

Para incorporar el buildpack necesario vamos al dashboard de Heroku y en Settings -> Buildpacks borramos lo que haya y agregamos el siguiente.

<https://github.com/mars/create-react-app-buildpack>

Buildpacks

Buildpacks are scripts that are run when your app is deployed. They are used to install dependencies for your app and configure your environment.

[Find new buildpacks on Heroku Elements](#)

Add buildpack

Buildpacks will appear here


Buildpacks are used to install dependencies for your app and configure your environment.


Add Buildpack


Enter Buildpack URL


`https://github.com/mars/create-react-app-buildpack`


Or select from our officially supported buildpacks



nodejs



python



php



ruby


java


go


gradle


scala


clojure

Save changes

Your new buildpack configuration will be used when this app is next deployed.



Add buildpack

Paso 7

Necesitamos tener disponible en Heroku la variable que le indique a la aplicación la URL de la API tal cual tenemos en nuestro archivo **.env**, para esto vamos a :

Dashboard -> Settings -> Config Vars y creamos la variable **REACT_APP_API_URL** con la URL del backend sin la barra final.

Config Vars Hide Config Vars

REACT_APP_API_URL	https://sitiofullexperto.herokuapp.com	 
KEY	VALUE	Add

Paso 7

Para configurar el repositorio de nuestra nueva app en la computadora nos dirigimos con la consola al directorio donde creamos el frontend y ejecutamos este comando reemplazando "transportesreact" por el nombre que hayamos definido o Heroku nos haya asignado.

```
heroku git:remote -a transportesreact
```

Paso 8

Nos aseguramos de tener todo commiteado localmente y hacemos el push a heroku con el siguiente comando: `git push heroku master`. Si todo sale bien nuestra aplicación estará desplegada y lista para ser compartida con el mundo!



Bibliografía utilizada y sugerida

Artículos de revista en formato electrónico:

Heroku. Disponible desde la URL: <https://www.heroku.com/>

npmjs. Disponible desde la URL: <https://www.npmjs.com/>