Deployment

Heroku: Nos va a permitir deployar nuestra app a un servidor y que esté online. Esta herramienta funciona muy bien con github y nos va a dar toda la infraestructura para llevar nuestra aplicación a **producción**.

La idea es:

- Tengo un repo en Github
- Me armo una cuenta en Heroku y las vinculo
- Cada vez que hago un cambio, puedo pushear a dos lugares => Github por un lado y Heroku por otro

Cada vez que hacemos un push a Heroku, se actualiza el código que publicamos.

En nuestro proyecto teníamos dos “aplicaciones”

1. Website
2. API

Como vimos, existe un gran beneficio en que estas dos estén separadas, es por eso que la API va a ser una aplicación, mientras que el website va a ser otro. Ambas aplicaciones deployadas a Heroku

Heroku

Existe una versión gratuita. Tenemos que registrarnos. Una vez que nos registremos vamos a poder crear aplicaciones que necesitemos.

CLI

Heroku nos provee con una CLI para que podamos conectarnos de manera directa a través de la terminal. La descargamos desde https://devcenter.heroku.com/articles/heroku-cli

Para macOS users, primero van a tener que instalar brew: https://brew.sh/index_es

Una vez instalado

Chequeo

1. Reiniciamos la terminal (Ya sea dentro del VSCode o la terminal de nuestro OS)

2. Corremos el comando **heroku -v** para asegurarnos que quedó instalada nuestra versión.

Logueo

1. Corremos el comando **heroku login**. Esto nos va a pedir que apretemos alguna tecla para hacer un logueo desde el browser.
2. Nos logueamos
3. Cerramos la ventana del navegador, y deberíamos estar logrados.

Agregar claves

1. Corremos el comando **heroku keys:add**

Crear una aplicación en Heroku

1. Corremos el comando **heroku create nombre-de-app** en el root de nuestro proyecto.
2. Abrimos la ruta y chequeamos que la app haya quedado instalada de manera correcta.

Configurar nuestra app

1. Tenemos que indicarle a Heroku que use el **node app.js** similar a como venimos haciendo nosotros.
2. En el package.json, en la parte de scripts, le damos la key **start** y el comando a ejecutar **node app.js**
3. Corremos de manera local **npm run start** para chequear que todo sigue funcionando de manera correcta.
4. Cambiamos el puerto. Hasta ahora estuvimos trabajando con el puerto 3000, pero eso es solamente para trabajar en modo desarrollo. Heroku y muchos otros servicios tienen lo que se llaman variables de entorno que son simplemente variables en el entorno, a las cuales podemos acceder.
    1. Creamos una constante de puerto

```
const port = process.env.PORT || 3000;
```

    2. Reemplazamos el puerto 3000 por nuestra constante

Pushear a Heroku

1. Pusheamos nuestro código a git (para tenerlo actualizado, en la última versión)

2. Corremos el código **git push heroku**

Actualizar código a Heroku

1. Agregar nueva información en nuestro sitio
2. Agregamos los cambios a Git
3. Volvemos a correr el comando **git push heroku master**

Extra - Variables de entorno

Agregar

heroku config:set key=value

Eliminar

heroku config:unset key

Listar

heroku config

Extra - Claves SSH

Es una alternativa más segura a las contraseñas. Tenemos dos:

- Clave privada
- Clave pública

La privada queda en nuestra computadora.

La pública la podemos agregar donde querramos. Se copia en distintas ubicaciones remotas.

Cuando se establece una conexión, se establece una comparación para ver si ambas claves están relacionadas, y si lo están, se permite el acceso.

Generar claves:

Para generar una clave, corremos el comando:

ssh-keygen -t rsa

Mantengamos la ubicación por default dado que nos va a traer muchos beneficios.

Copiar clave SSH

cat ~/.ssh/id_rsa.pub

Agregar claves a Heroku

heroku keys:add

https://docs.oracle.com/en/cloud/cloud-at-customer/occ-get-started/generate-ssh-key-pair.html

https://www.digitalocean.com/community/questions/copy-ssh-key-to-clipboard