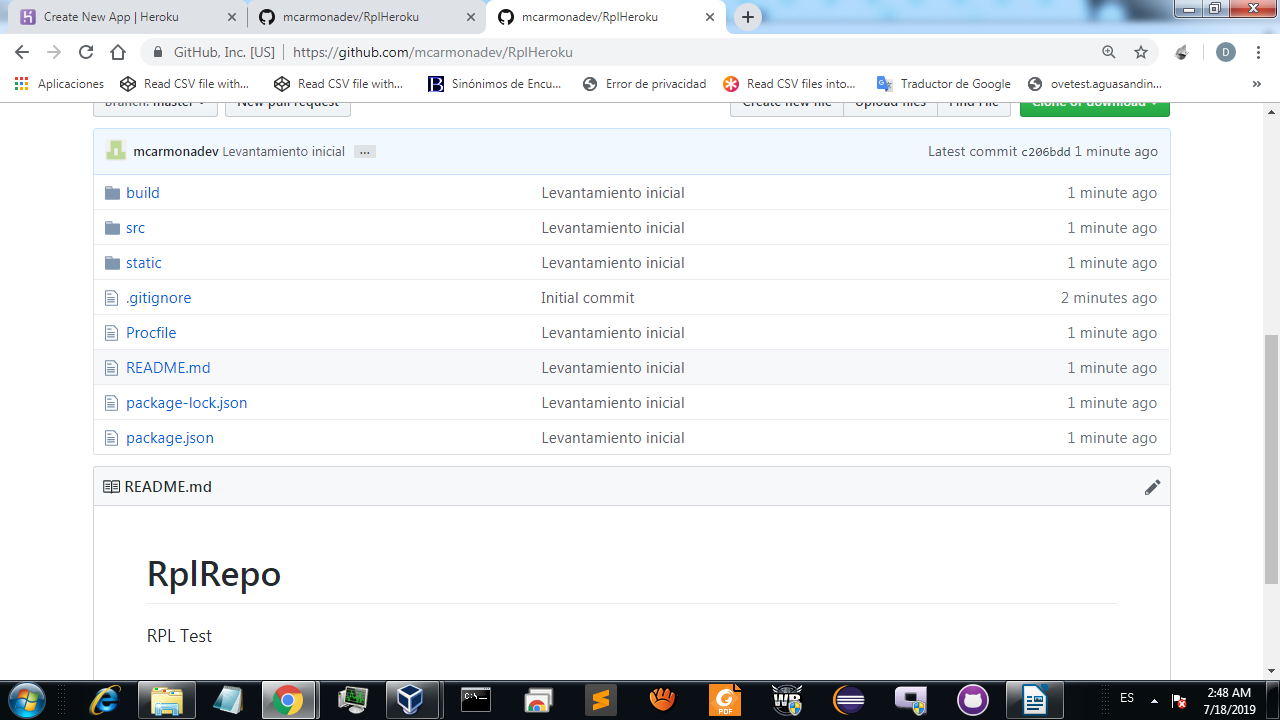
Bienvenido!!

El repo, del aplicativo backend, que incluye un bundle estatico hosteado del script de front… está en el siguiente repo:

<https://github.com/mcarmonadev/RplHeroku>



El aplicativo está desplegado en heroku disponible en:

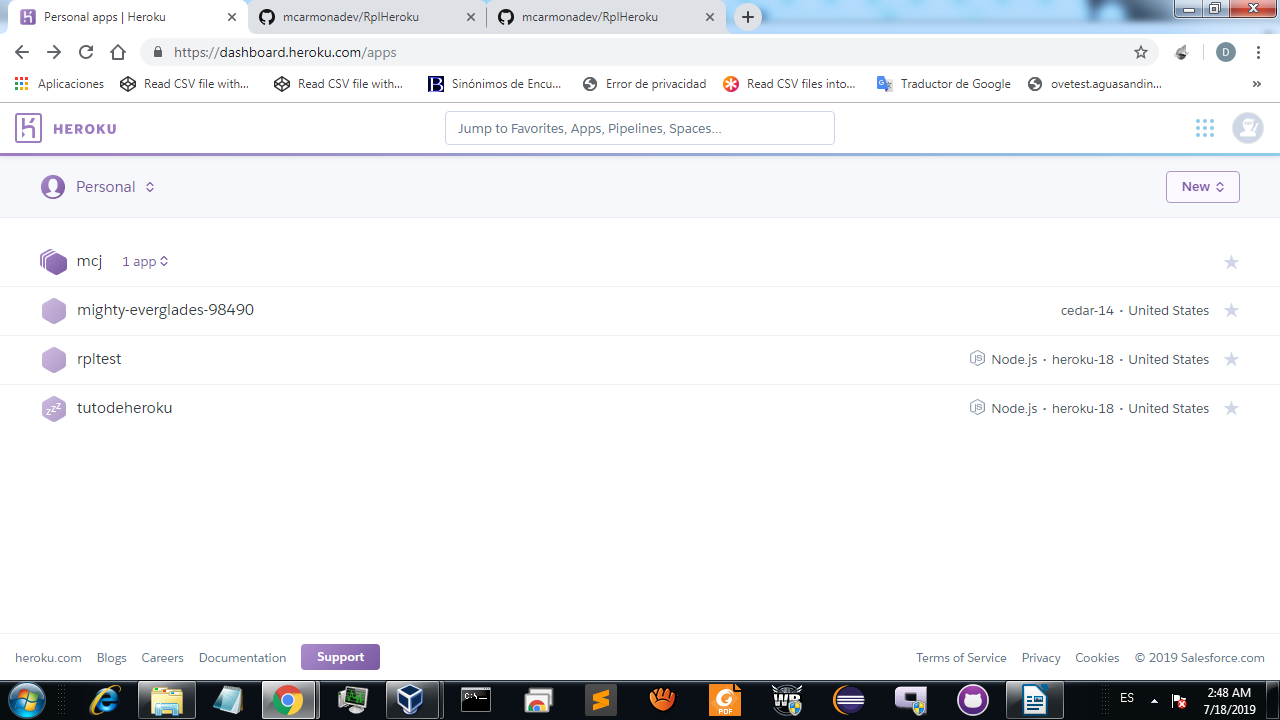
<https://rpltest.herokuapp.com/>

Guia de pauta para despliegue en heroku:

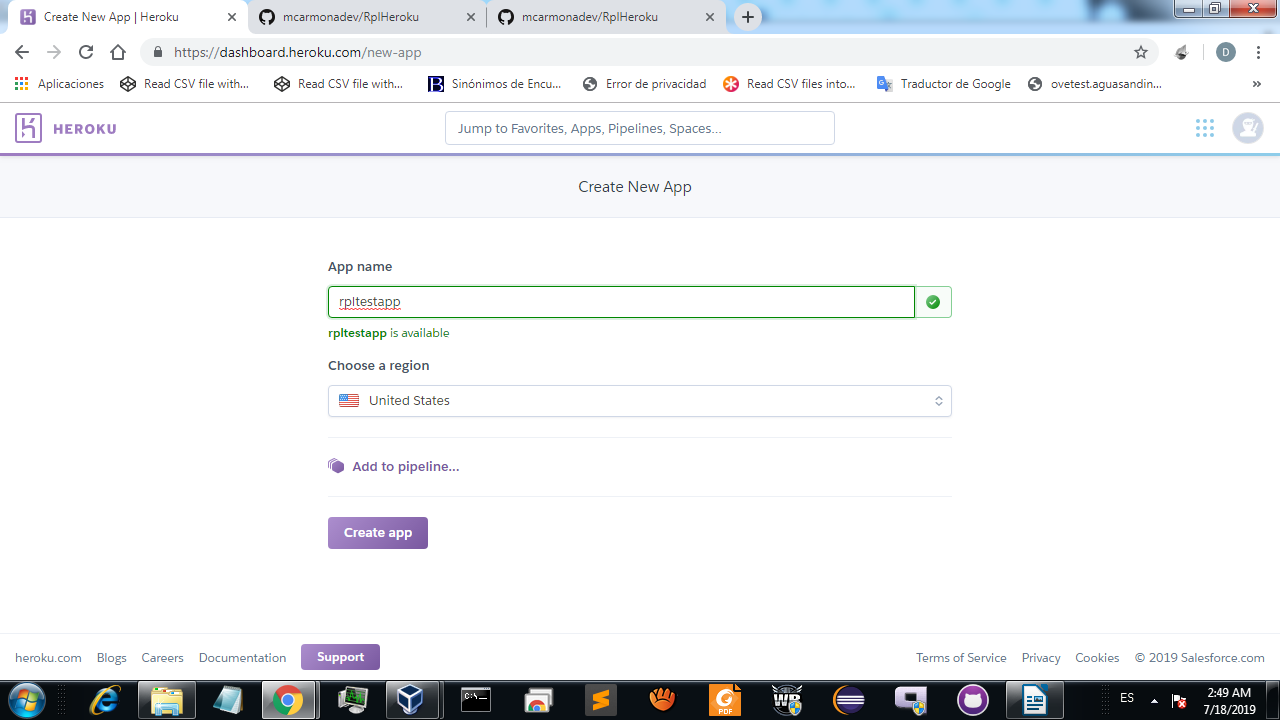
Primero tienes que clonar el repo de github antes mencionado, en tu propio repo para poder autorizar a heroku sobre utilización del origen remoto

<https://github.com/mcarmonadev/RplHeroku>

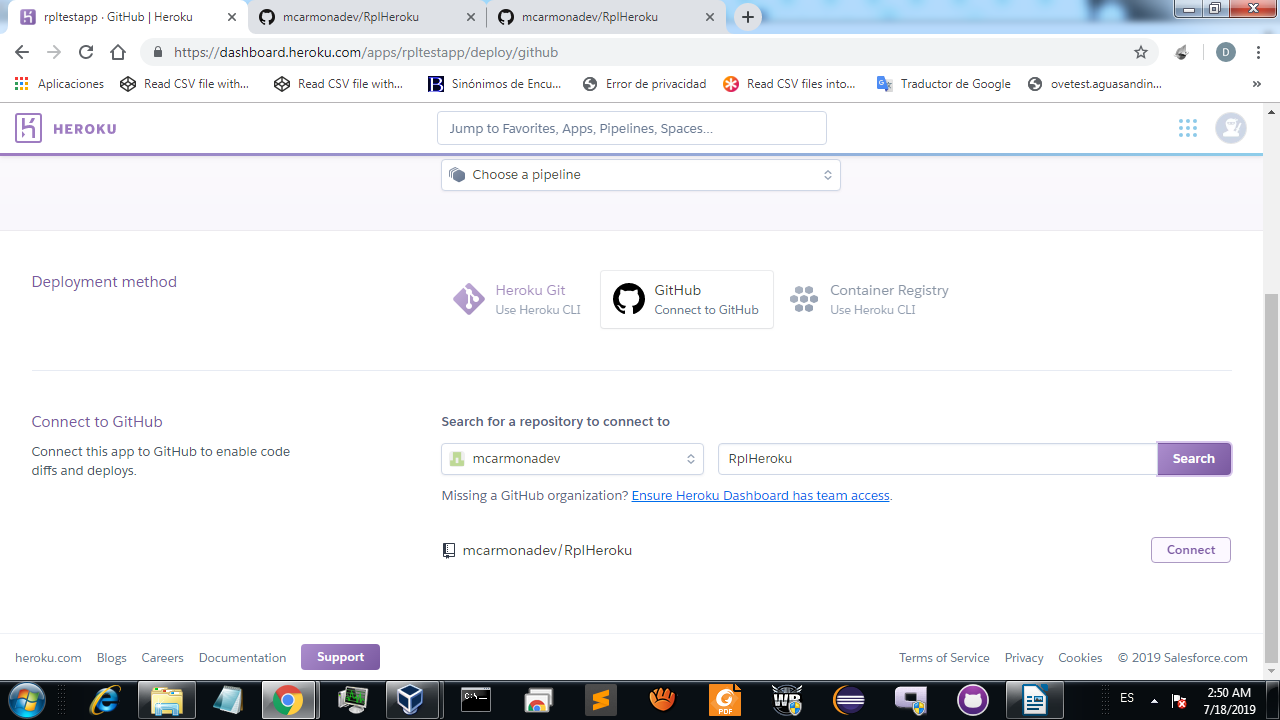
En segunda, tienes que crear una aplicación en heroku (en este caso visualmente aunque está la opción de utilizar comandos del cli)



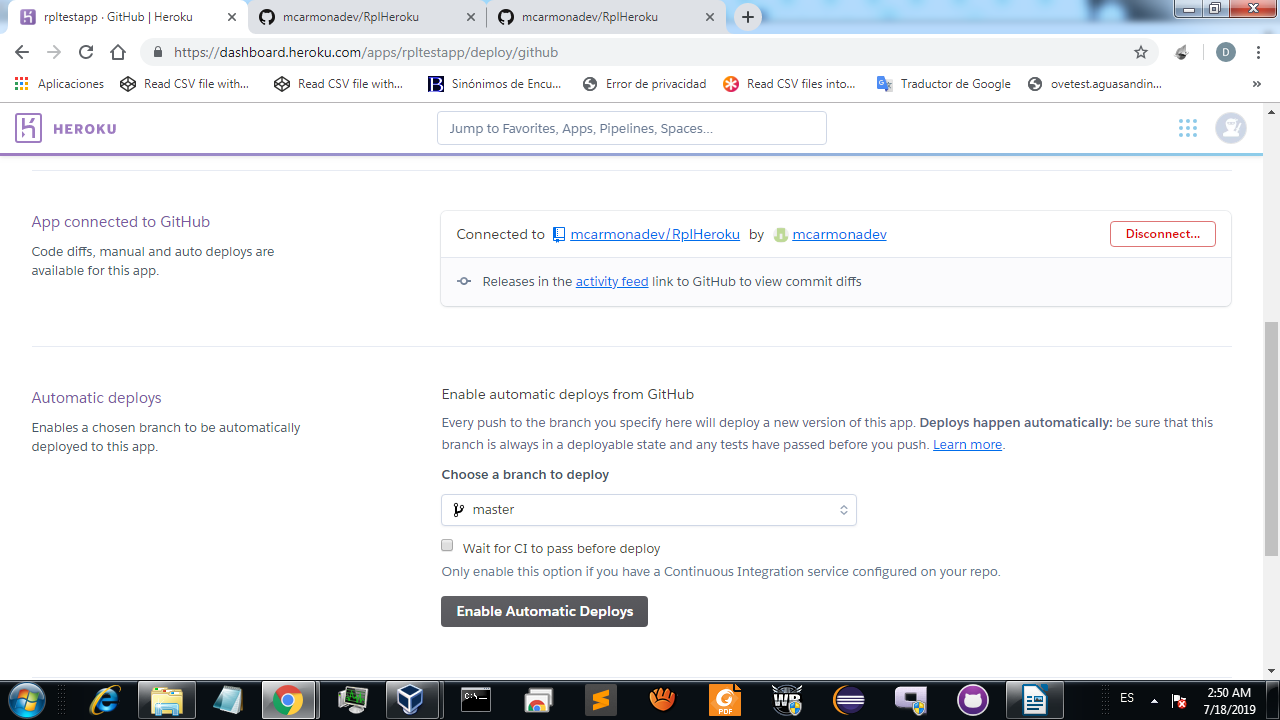
Crear nuevo aplicativo, en este caso mi aplicativo utilizó el nombre ‘rpltestapp’



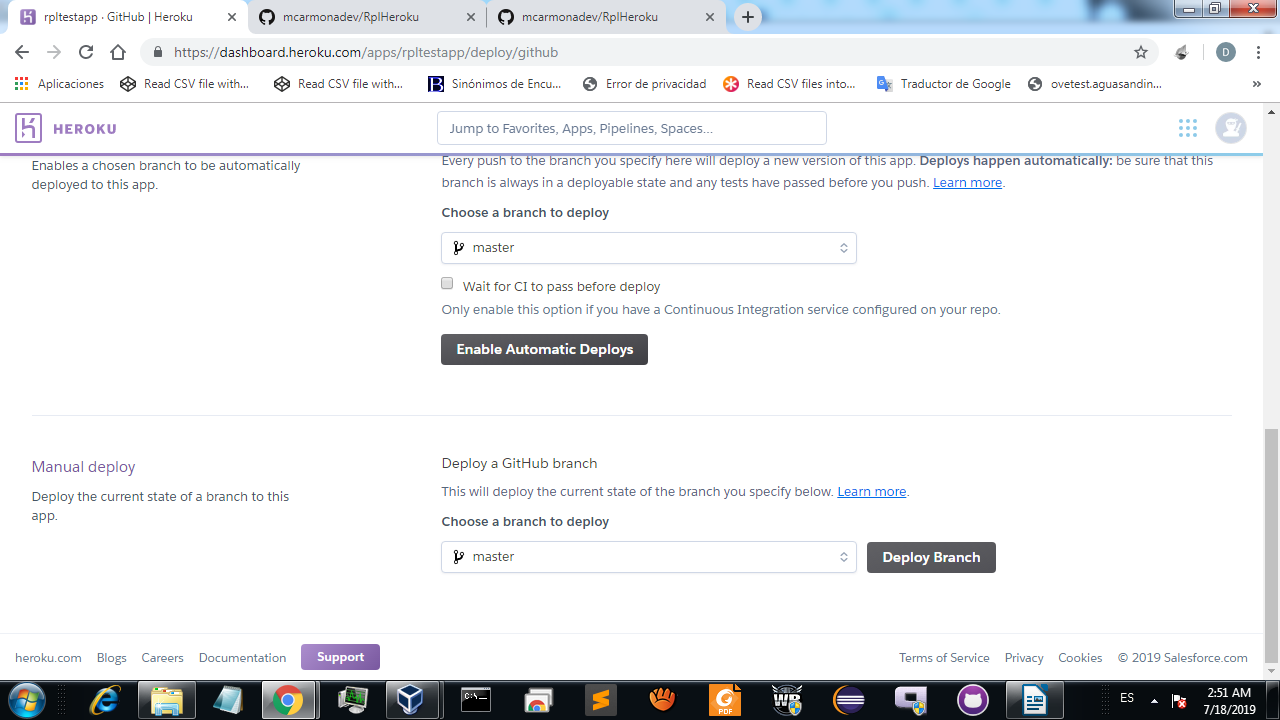
A continuación la pantalla te llevará a definir origen de fuentes para control de versiones. Previamente debes



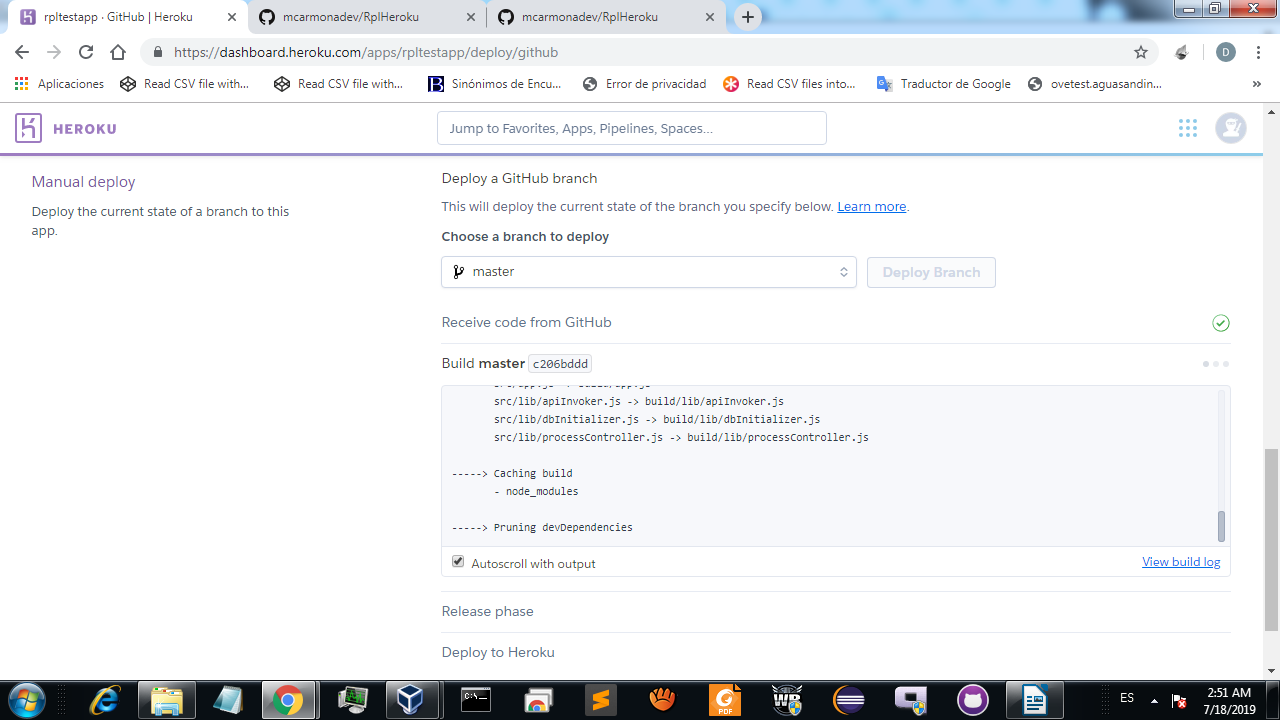
Despues de configura el repositorio, el botón connect, canbiara de opción a Desconnect



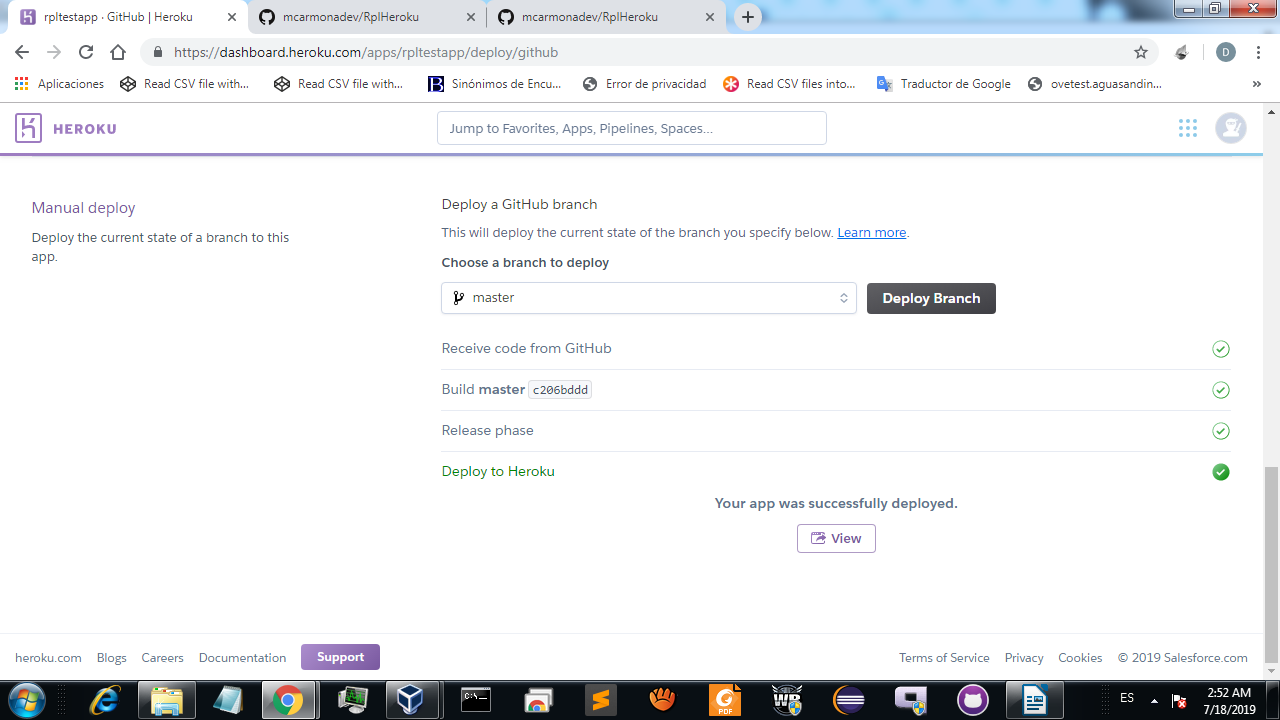
Luego, de aquello, presiona en Deploy Branch, sin necesidad de cambiar rama, porque existirá solo Master



En la imagen se grafica la reacción esperada en el sitio de heroku, pues se mostrará la actividad verbosa de la linea de comando en el panel en movimiento:



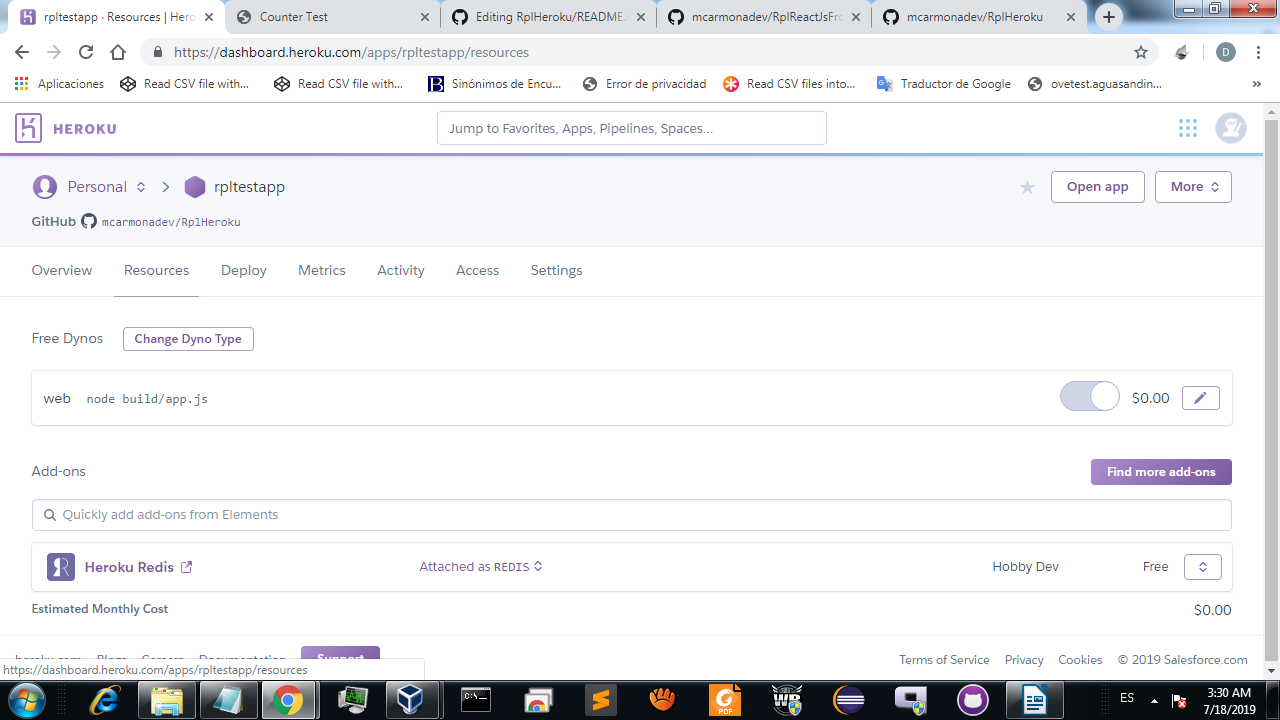
Terminando el despliegue, debe indicarse en color verde “Deploy in Heroku’



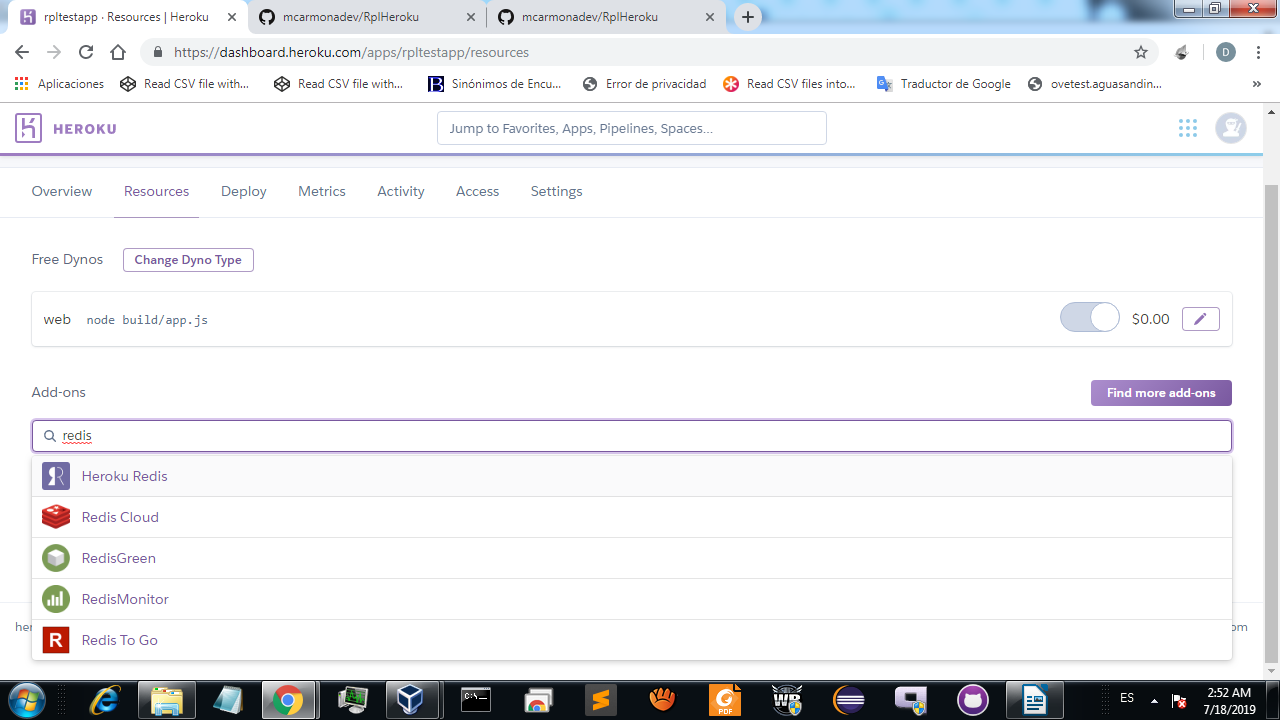
Despues de esto, tienes la url en ‘Settings’ >> ‘Domains and certificates’

Pero primero antes de visitar la url, debes antes provisionar al aplicativo con Redis

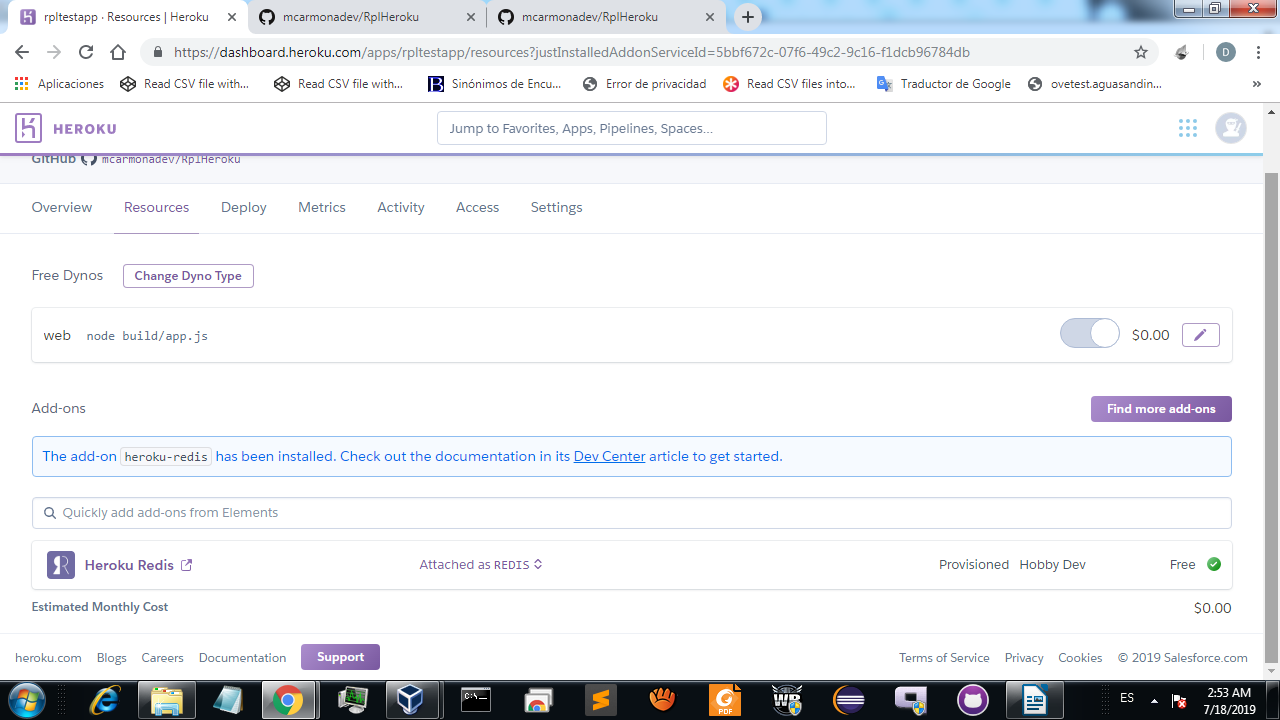
Vaya a la opción Resources



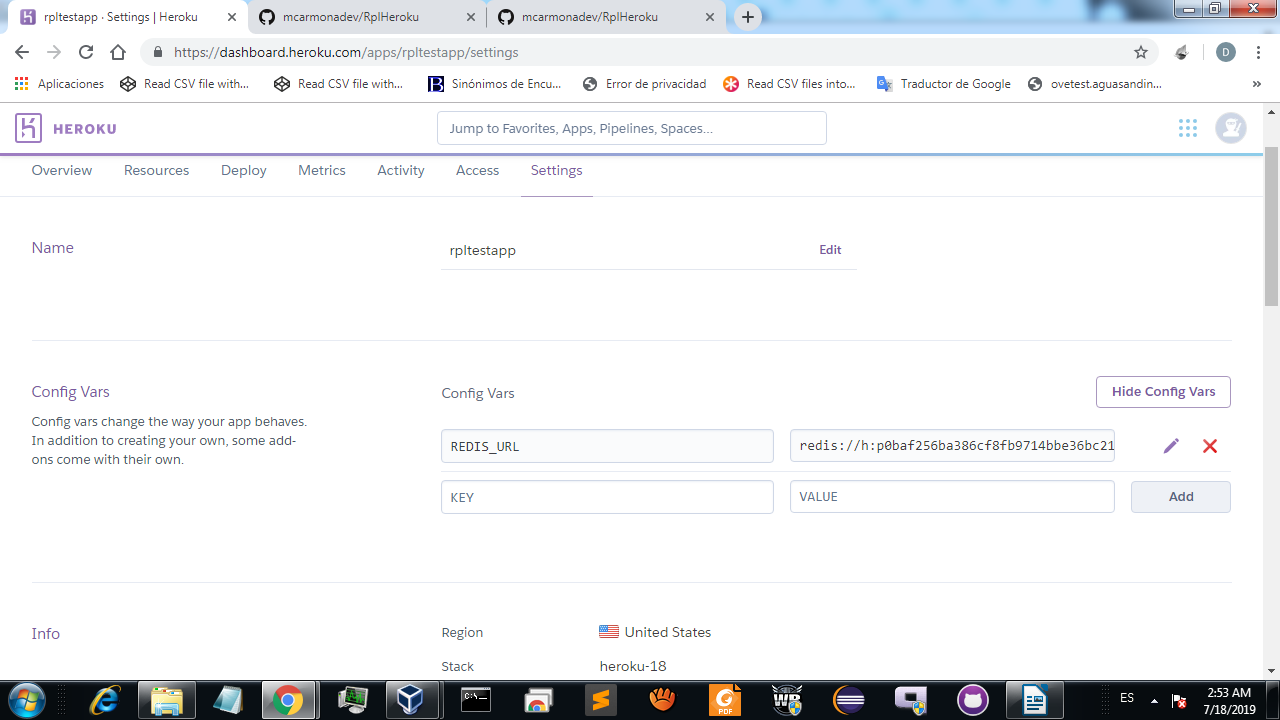
Busque el add on ‘redis’ y selecciones que tiene de nombre ‘Heroku Redis’



Espere el tiempo que toma completar el aprovisionaiento, hasta que se muestre la señal en color verde, como se muestra en la imagen:

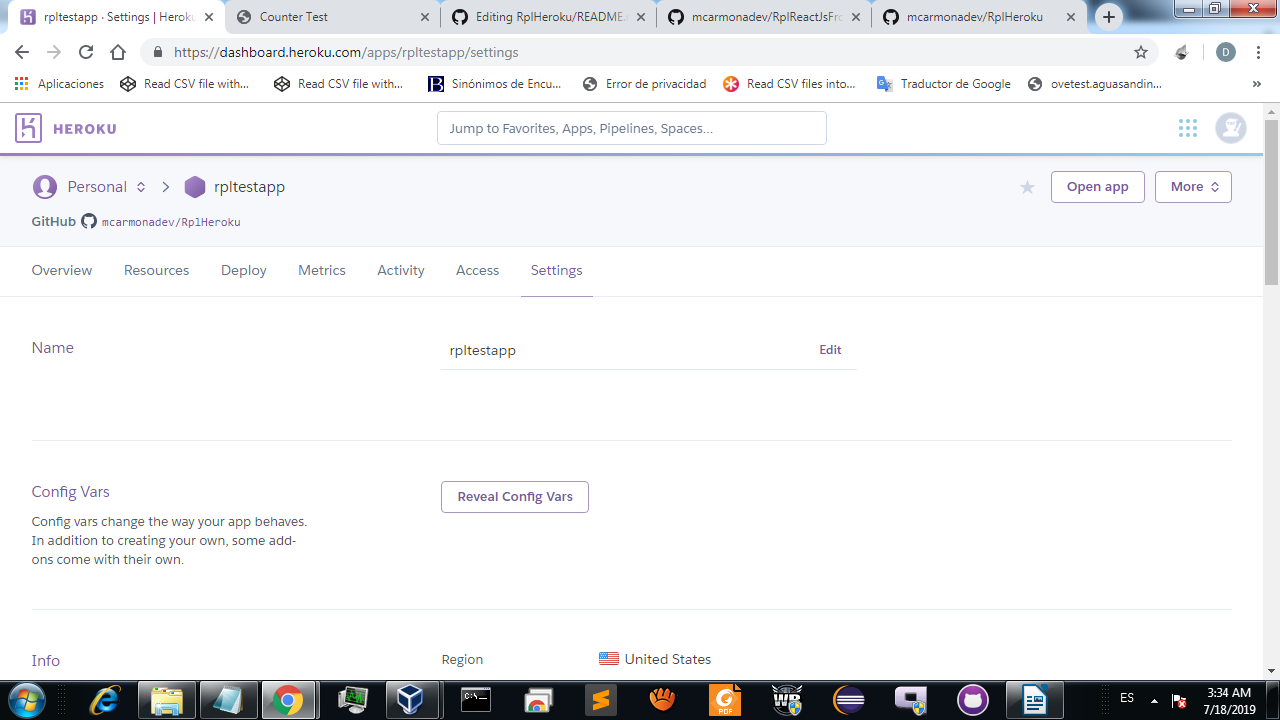


Vale mecionar, que al término de esta acción, automáticamente Heroku habrá creado una variable de ambiente signada REDIS\_URL, con el string de acceso al recurso



Llegando al fin de esta guia, ubicaremos la url para entrar al aplicativo

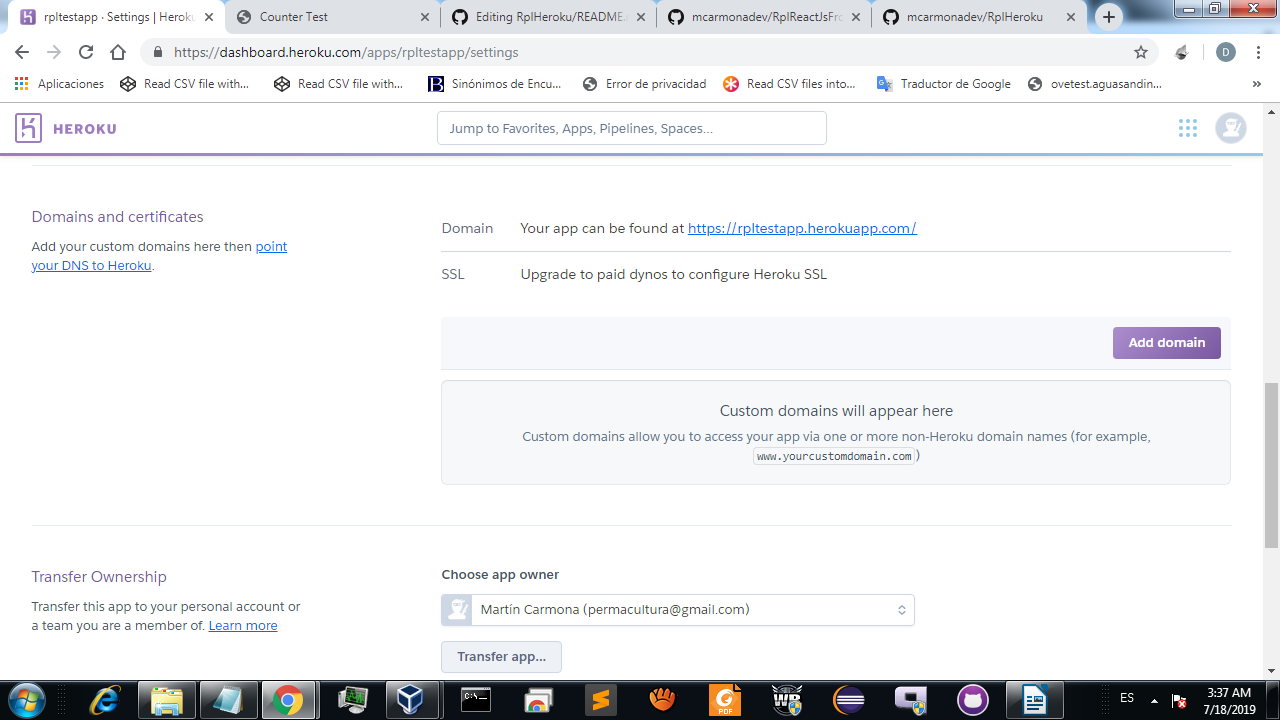
Para esto, pinche en la opción ‘Settings’



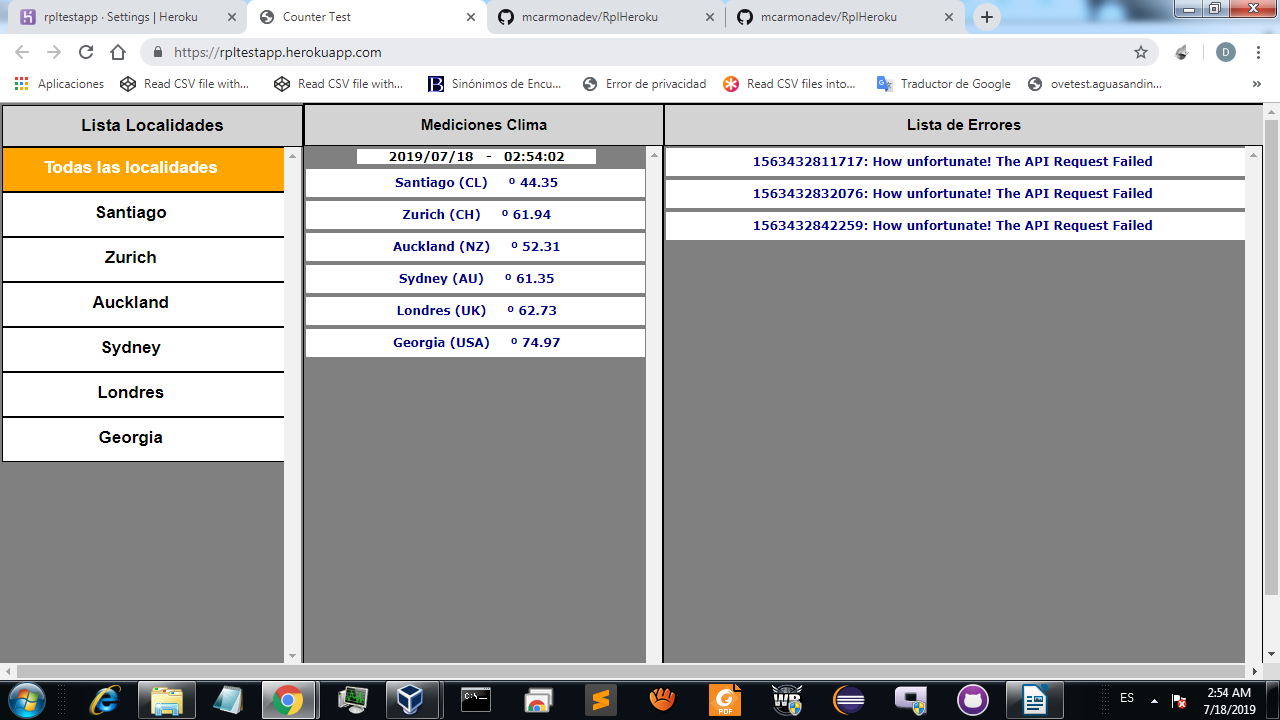
Más abajo en la sección ‘Domains and certificates’, se informa la url del aplicativo

Pinchar en esta url para visitar el aplicativo, en mi caso la url final es:

<https://rpltestapp.herokuapp.com/>



En 10 segundos, debería haber cargado, la primera tanda de lecturas a darksky



En caso de pasar más de 10 segundo, recargar en el navegador tipeando tecla F5

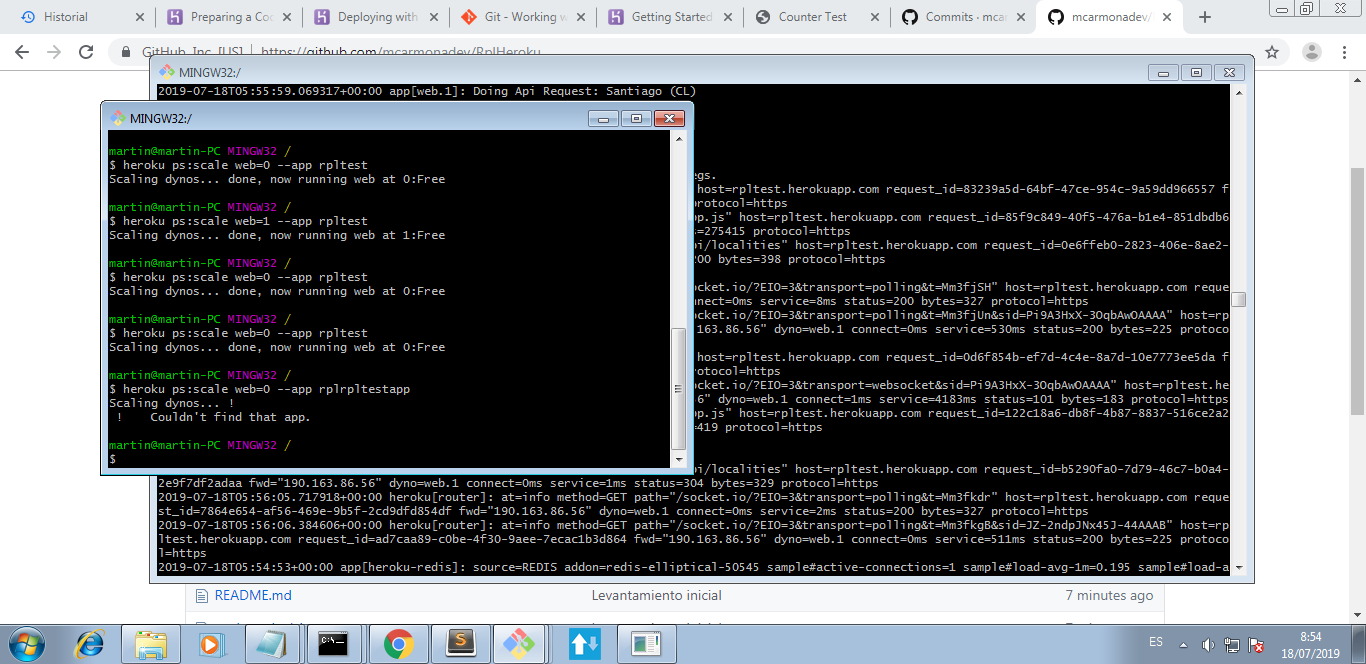
Si desea detener el aplicativo para detener los request a DarkSky, detener el contenedor con el comando

heroku ps:scale web=0 --app rpltestapp

...remplazando ‘rpltestapp’ por el nombre de la nueva instancia

Para volver a inciar el contendor, se cambia el parámetro web=1, con el mismo comando

heroku ps:scale web=1 --app rpltestapp



Adicionales, bonus track:

Opcionalmente, si deseas cambiar la key de la api DarkSky para hacer los request contra otra instancia de cuenta, configurar la variable de ambiente:

DARKSKY\_KEY

Esto en la sección ‘Settings’ >> ‘Config Vars’

No te preocupes si no has creado esta variable, por defecto esta mi cta de la api SkyDark: ‘264f037d720e899ec02431f178da7196’

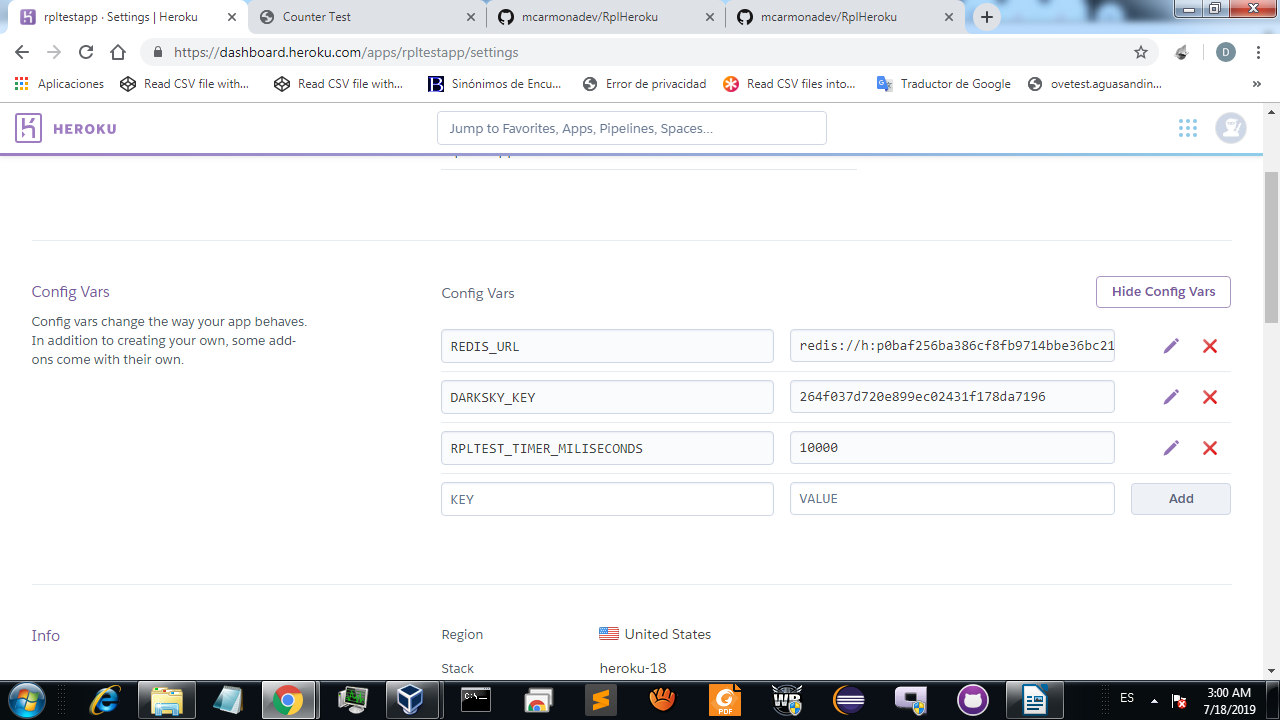
Pero si deseas utiilizar tu propia cuenta o si esta ya agotó su límite diario, tienes esta opción.

El time por defecto es de 10 segundos = diez mil milisegundos = 10.000

Est lo puedees modificar en la variable de ambiente:

RPLTEST\_TIMER\_MILISECONDS

Que por defecto tiene configurado 10.000, de acuerdo a la especificación de comportamiento, de requerimiento inicial



Este aplicativo, tiene una implementacion front end, que es un archivo java script, hosteado de forma estática en

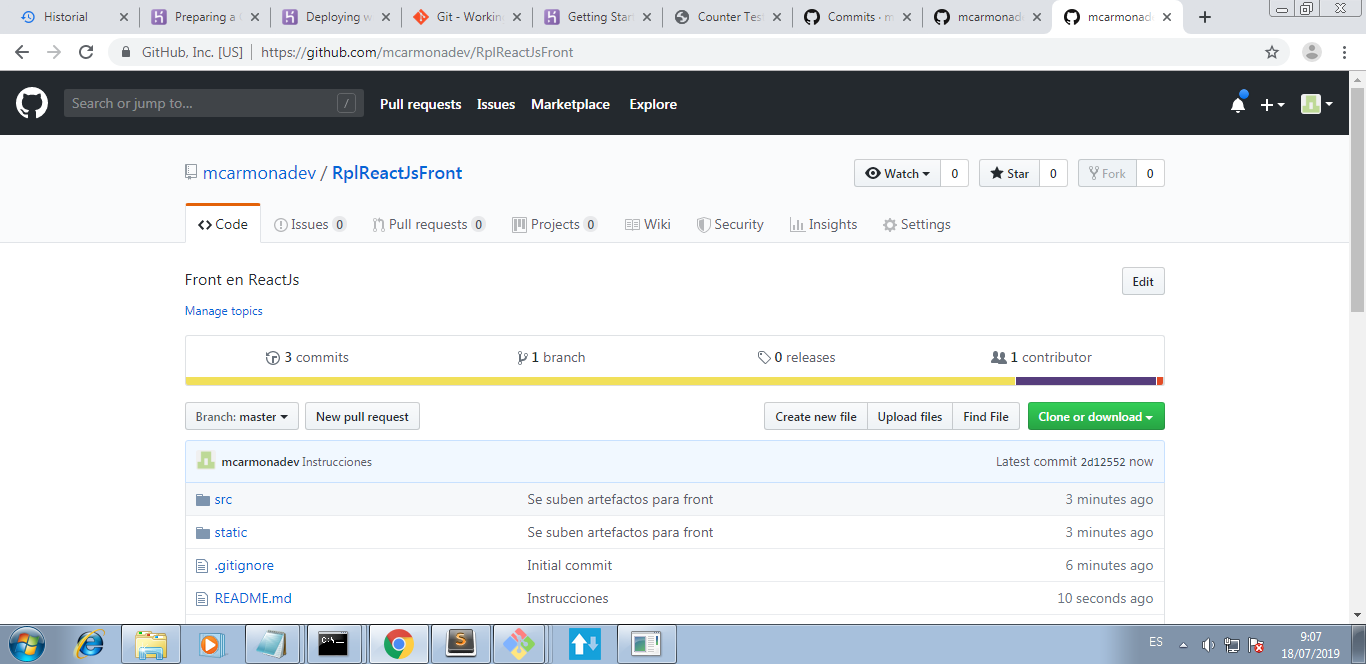
*/static*/app.sj

Sin embargo este archivo se genera en otro proyecto, que tiene respositorio a parte:

<https://github.com/mcarmonadev/RplReactJsFront>

Este repo, tiene como salida, el generar el archivo bundle, transpilado con la parte front del aplicativo, y este se genera descargando este segundo repo, instalandole las despendencias con: ‘npm install’… finalmente construyendo el bundle, con: ‘npm run build’

La idea de este segudo repo, es que dispongas de los fuentes de front para si deseas revisar su implementación



Cuando construyes con ‘npm run dev’, puedes opcionalmente modificar el front si deseas. Para esto tienes que copiar el archivo:

*/static* /app.js, en esa misma ruta en el otro repo, via commit, y deplotar nuevamente por el suitio Heroku, o mediante el CLI de Heroku si prefieres

