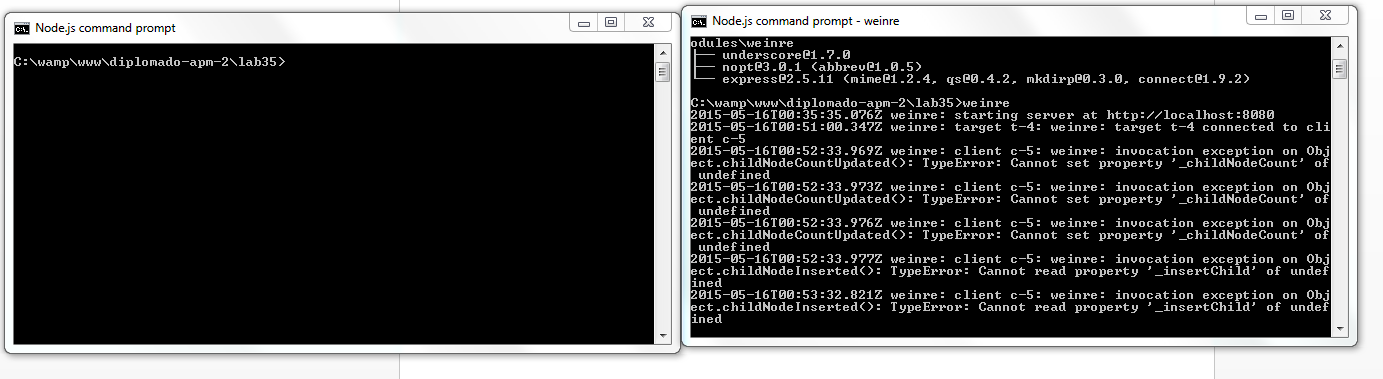
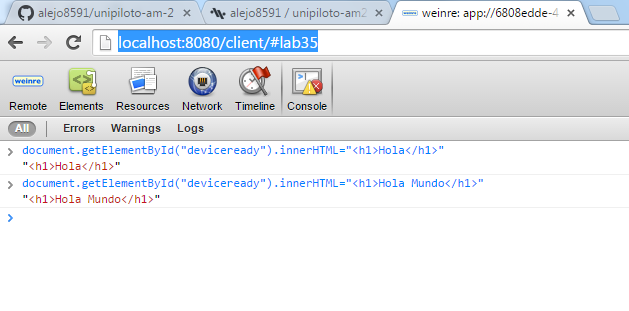
Se deben tener dos consolas abiertas de node.js para utilizar weinre

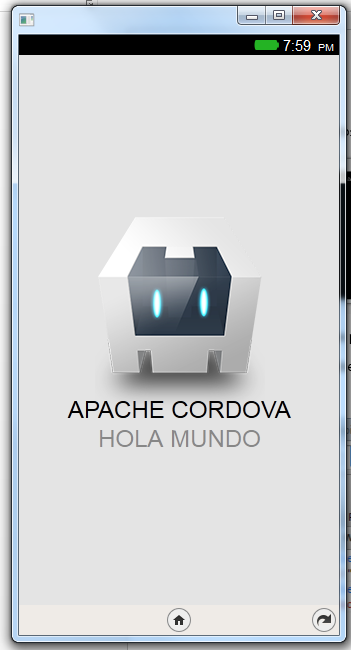


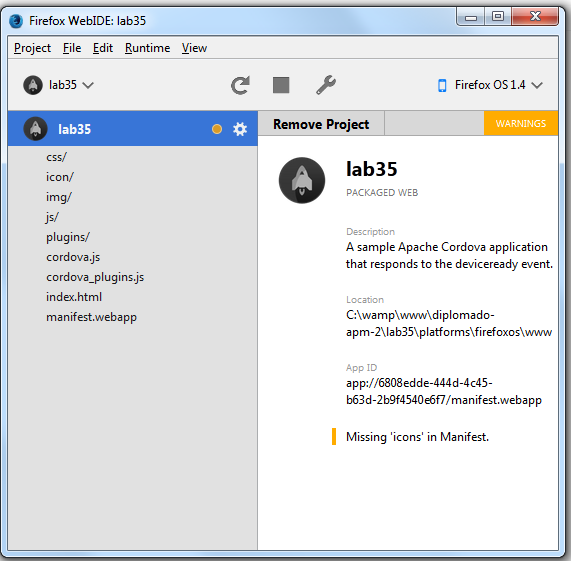
En una de ellas se pone a ejecutar (listen) con el comando: weinre y enter

En el browser debemos ir a la consola y teclear el siguiente comando



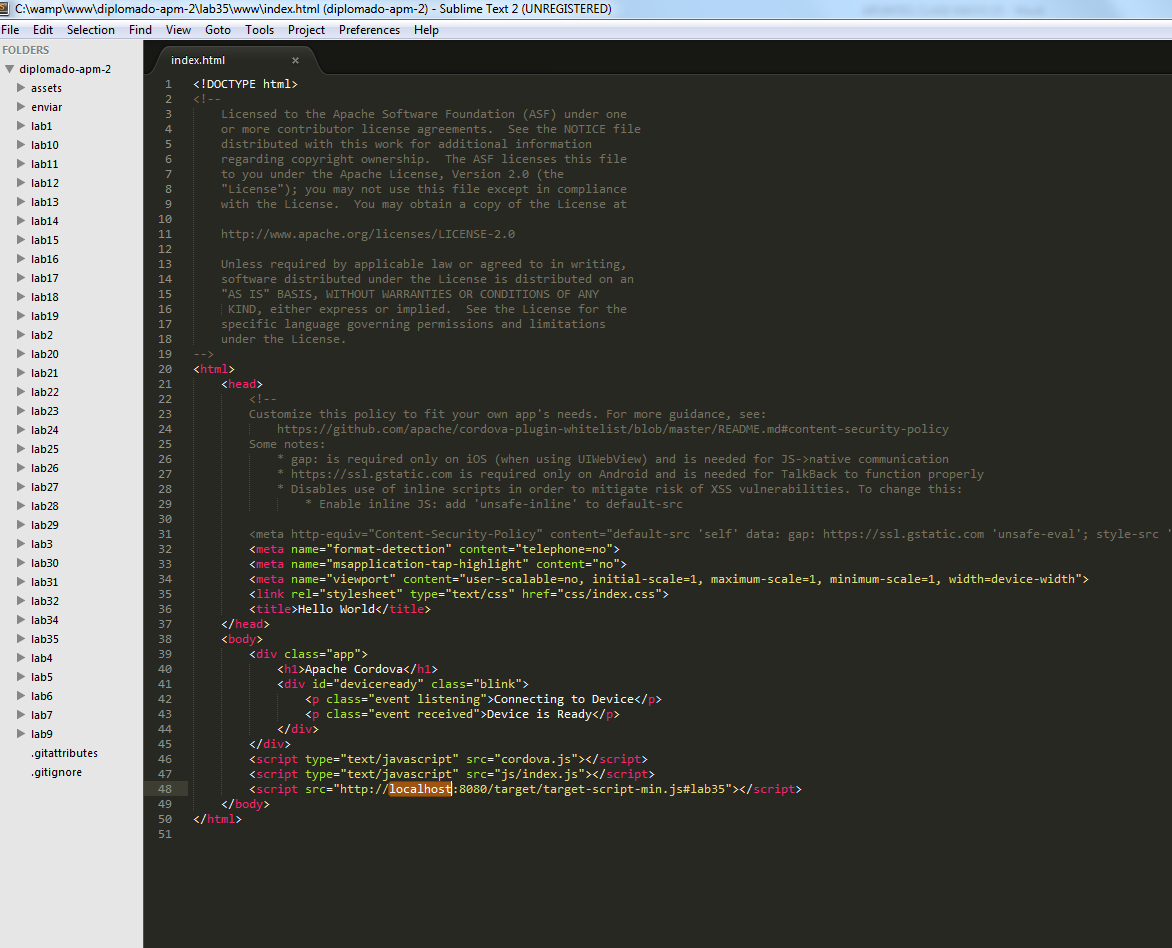
Luego veremos el simulador de firefoxos de la siguiente manera



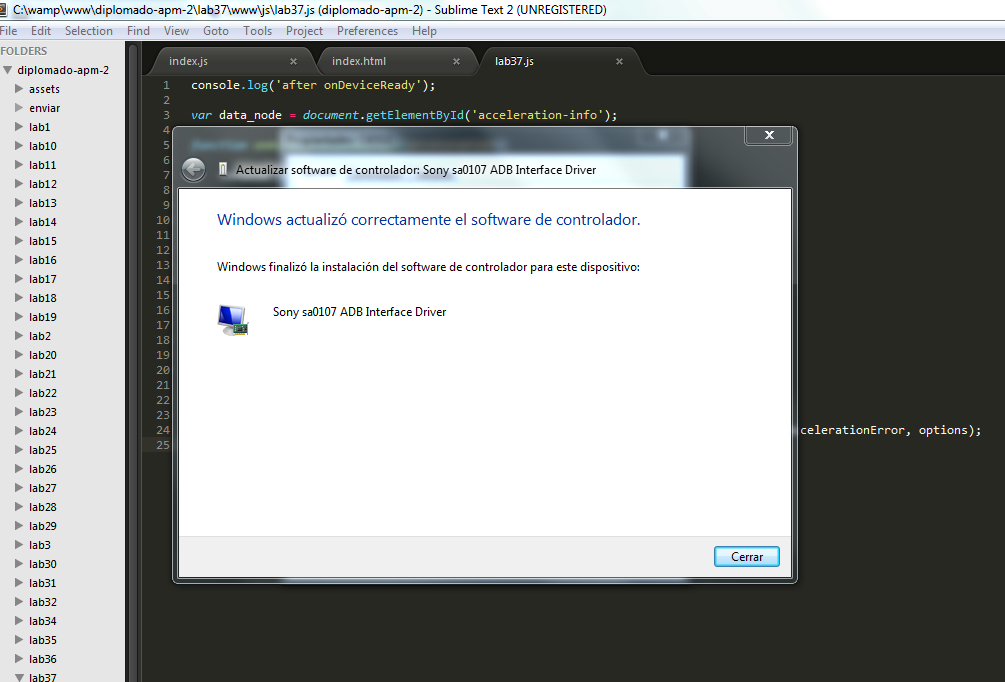


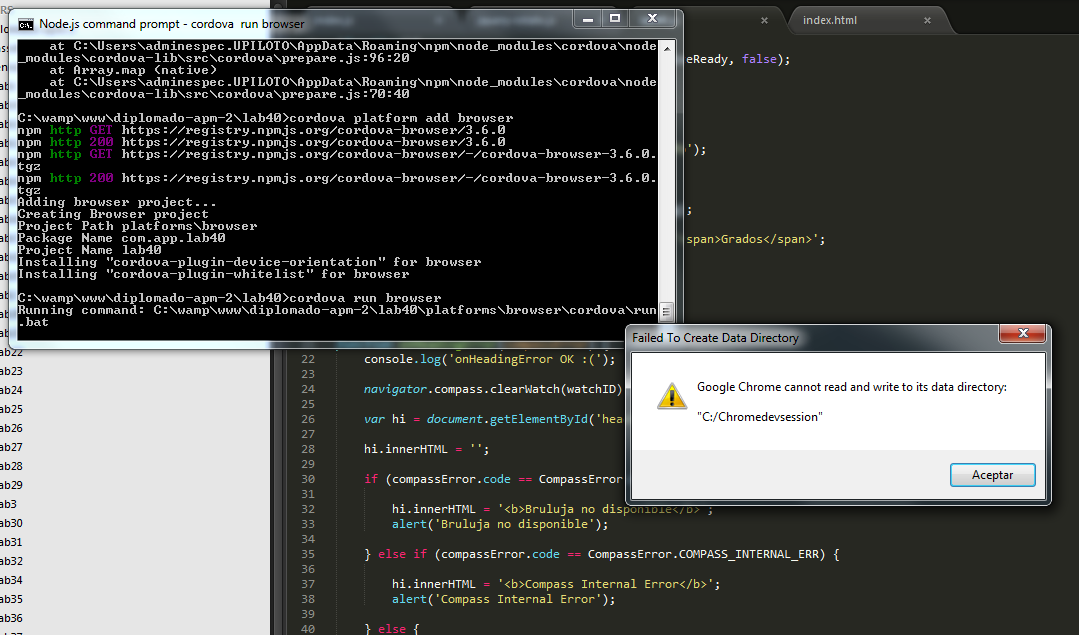
Ahora vamos a ejecutar ese mismo ejemplo en Android

En el index.html se debe agregar la siguiente IP: 10.0.2.2 en lugar del localhost que siempre utlilizamos, ya que Android tiene su propia dirección de localhost, se deja el mismo puerto 8080

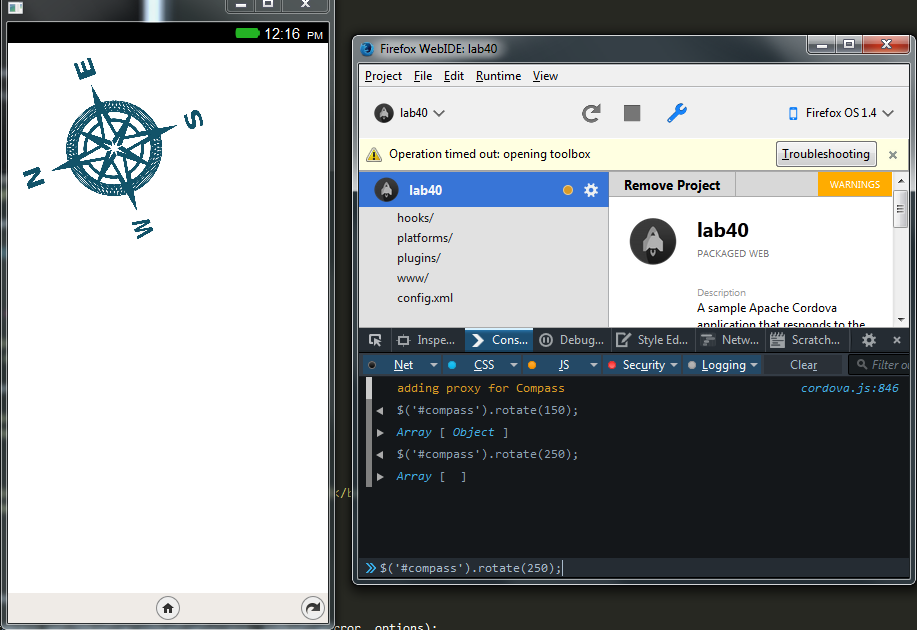


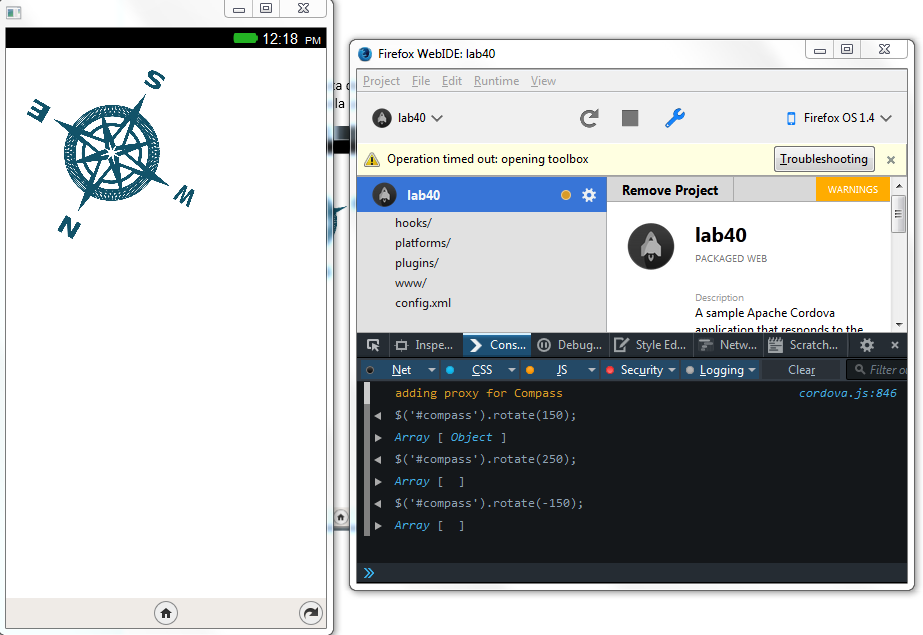




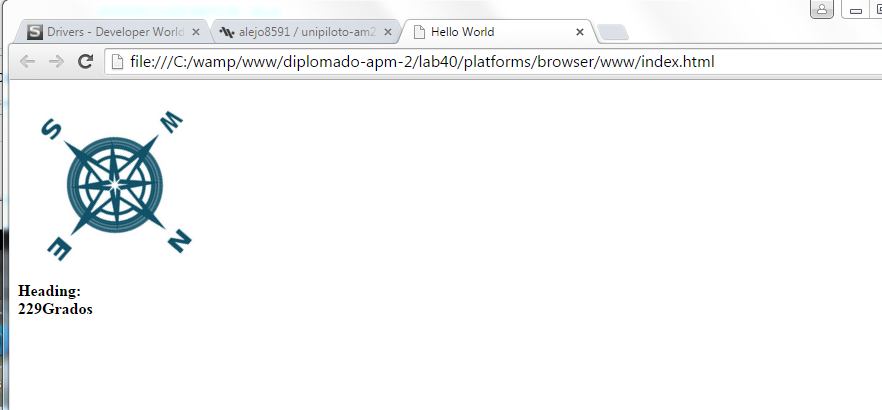


Cuando se ejecuta desde Firefox os, en WebId, al tener corriendo la app, ir a Project -> debug app y buscar la consola para ejecutar el siguiente comando, ahí se le indica cuantos grados girar

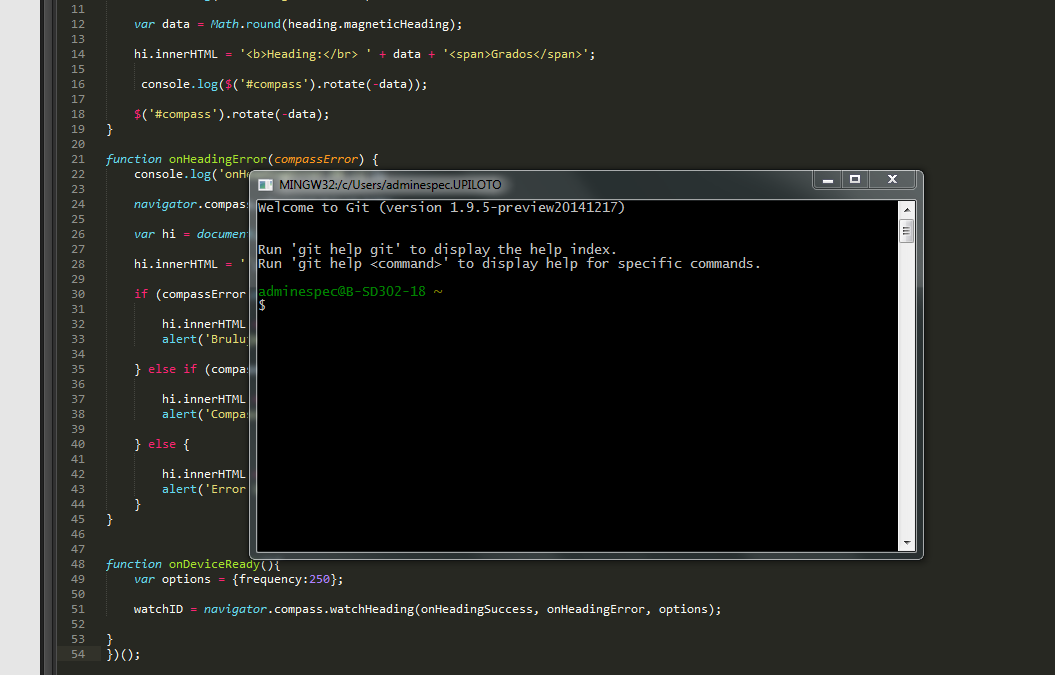




Ejecutándolo desde browser



Usando Git Bash



Tecleando el comando git init (recordemos que esta consola es tipo Linux),

Solo la primera vez



Recordemos que git maneja 3 estados, directory, stage, repositorio

Para dejar los cambios en estado “stage”

Con los comandos:

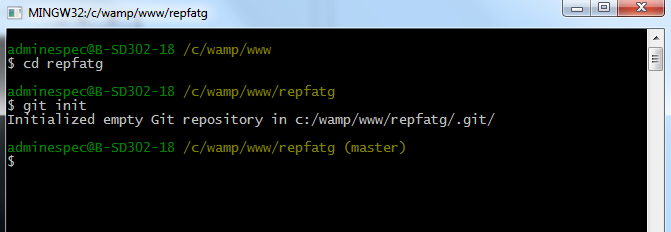
git add – A -> esto agrega todo el repositorio

git add . -> agrega al repositorio desde donde se está ubicado en adelante

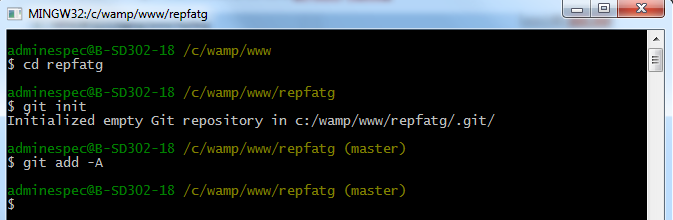
git commit –a –m “message“ -> hace el commit al repositorio

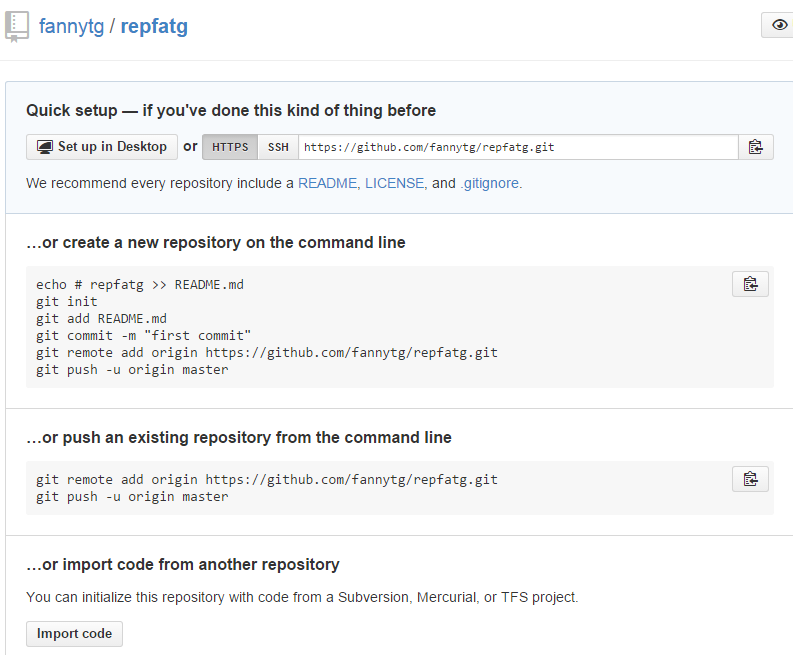
EJERCICIO MAYO 22 DE 2015

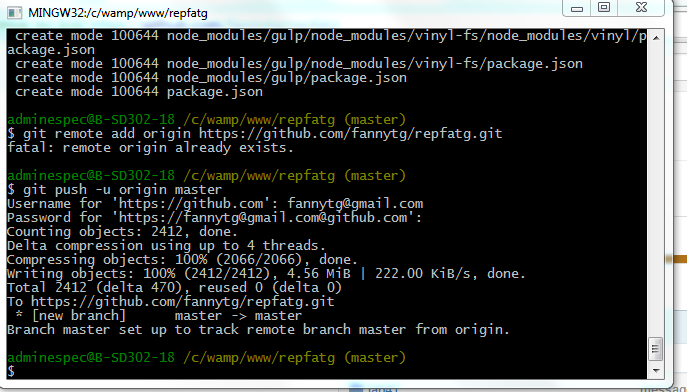
1. Crear un repo local desde la consola git bash, repfatg



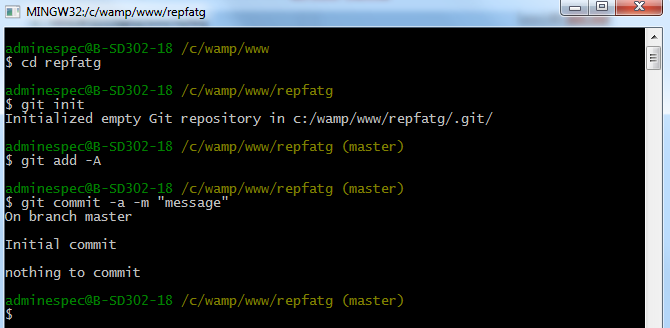
1. Crear repo remoto “repfatg”







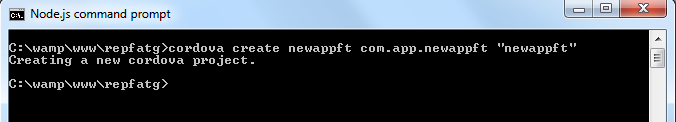
1. Sincronizar el repositorio





1. Crear una nueva app, en mi caso la llamare newappft

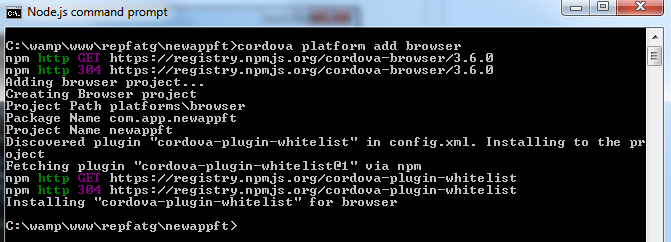
Voy a node.js



1. Adicionar plataforma “browser”

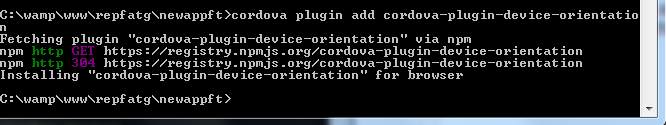
Tengo que ubicarme en el directorio de la app y teclear el comando

Cordova platform add browser

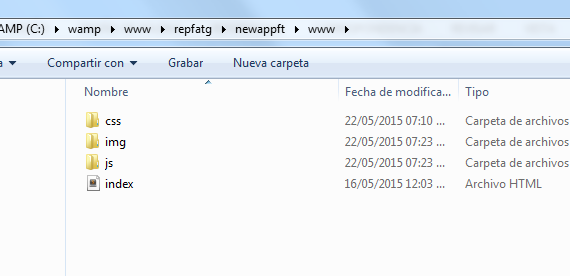


1. Adicionar plugin device-orientation, tecleando el siguiente comando

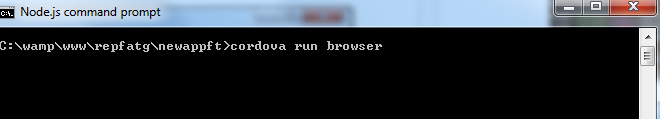
Cordova plugin add cordova-plugin-device-orientation



1. Adicionar interacción del último lab



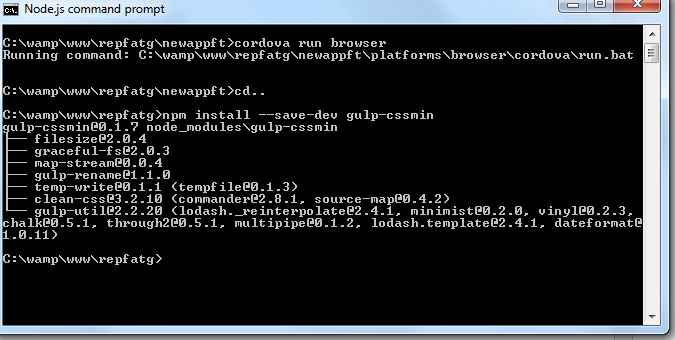
1. Probar la ejecución de la app y si funciona minificar



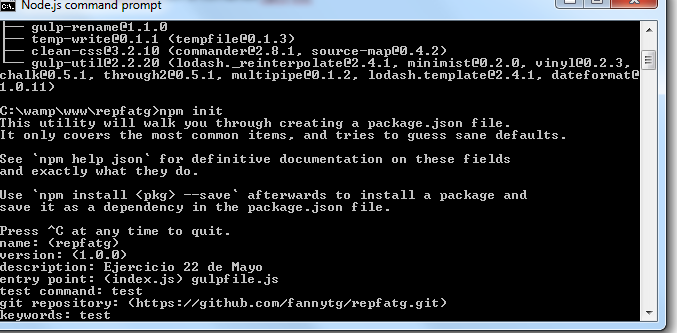


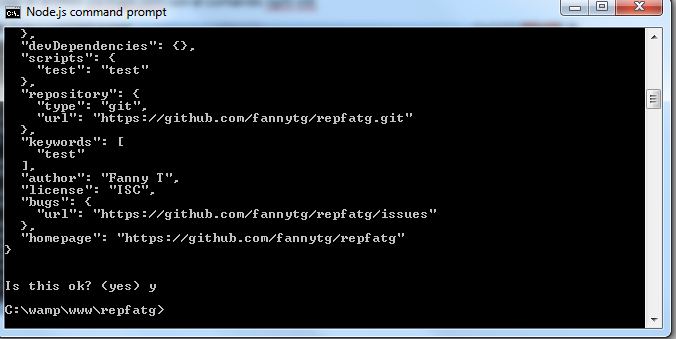


Instalar el plugin en la raíz del repositorio



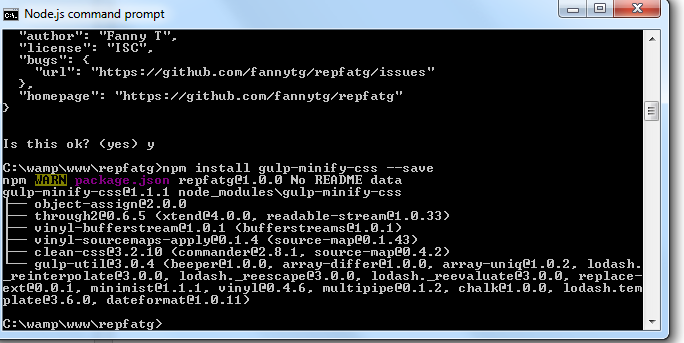
Crear el archivo package.json con el comando npm init

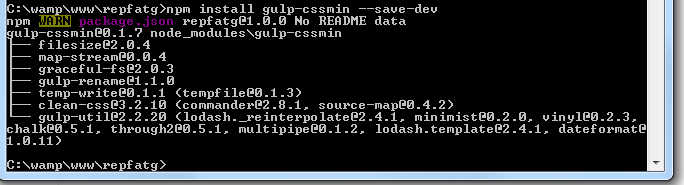




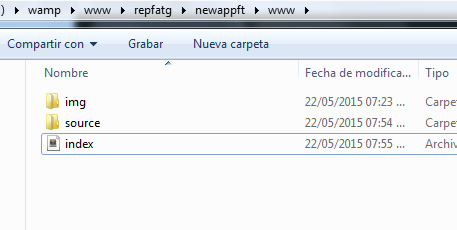
Minificar css







Se crea una carpeta source y allí se depositan los directorios css y js



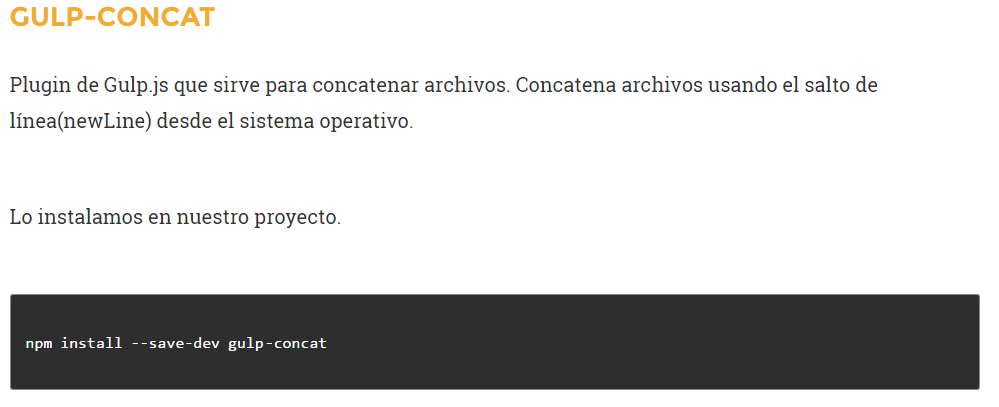
Se direccionan los archivos en index.html

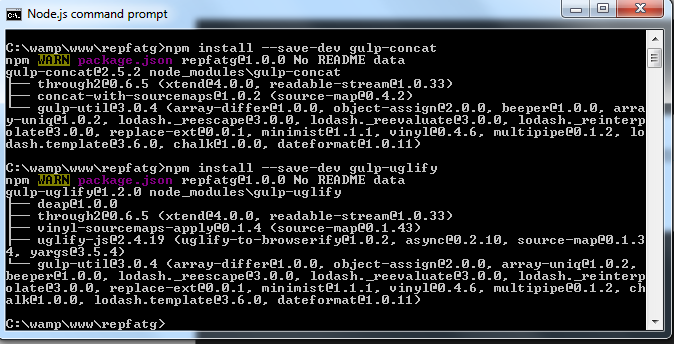


Crear la tarea para concatenar los js, para ello primero instalamos gulp

Npm install gulp

Luego hay que instalar los módulos:



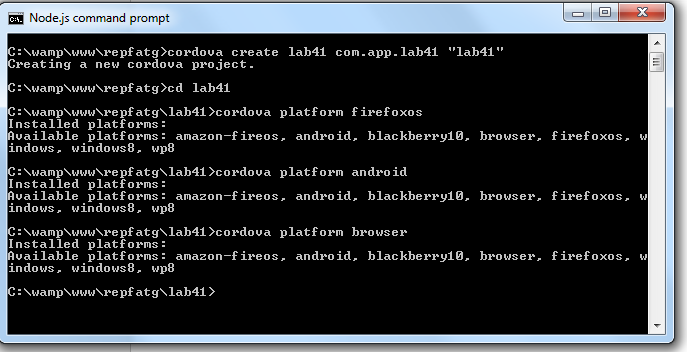


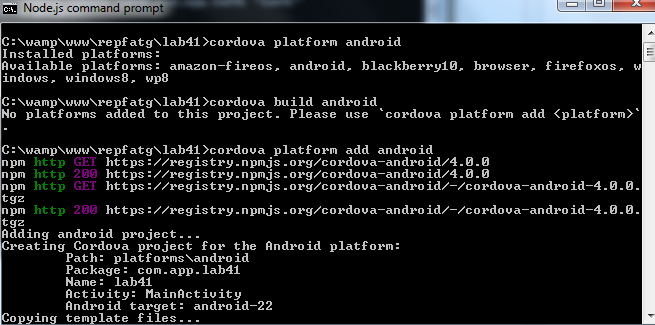
Luego se edita el gulpfile.js



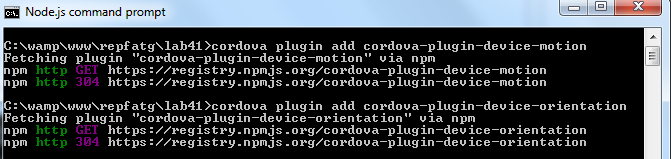
Revisar el ejercicio

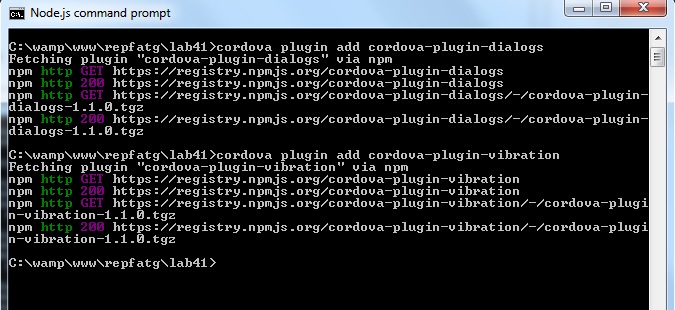
OTRO LABORATORIO





Ahora instalamos los plugin





Métodos

Notification.alert

Notification.prompt

Notification.beep

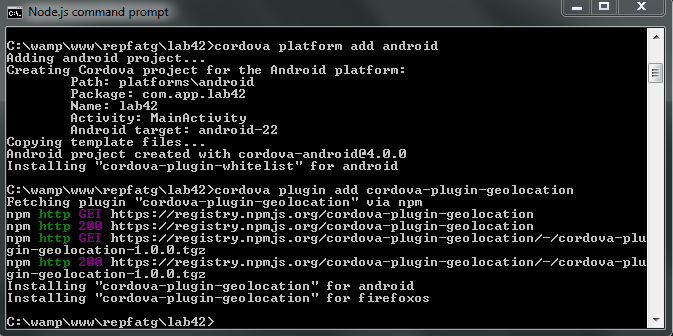
Compilamos con el comando: cordova build android

Ejecutamos con el comando: cordova run android

Lab 42 mayo 23 de 2015 GEOLOCALIZACION

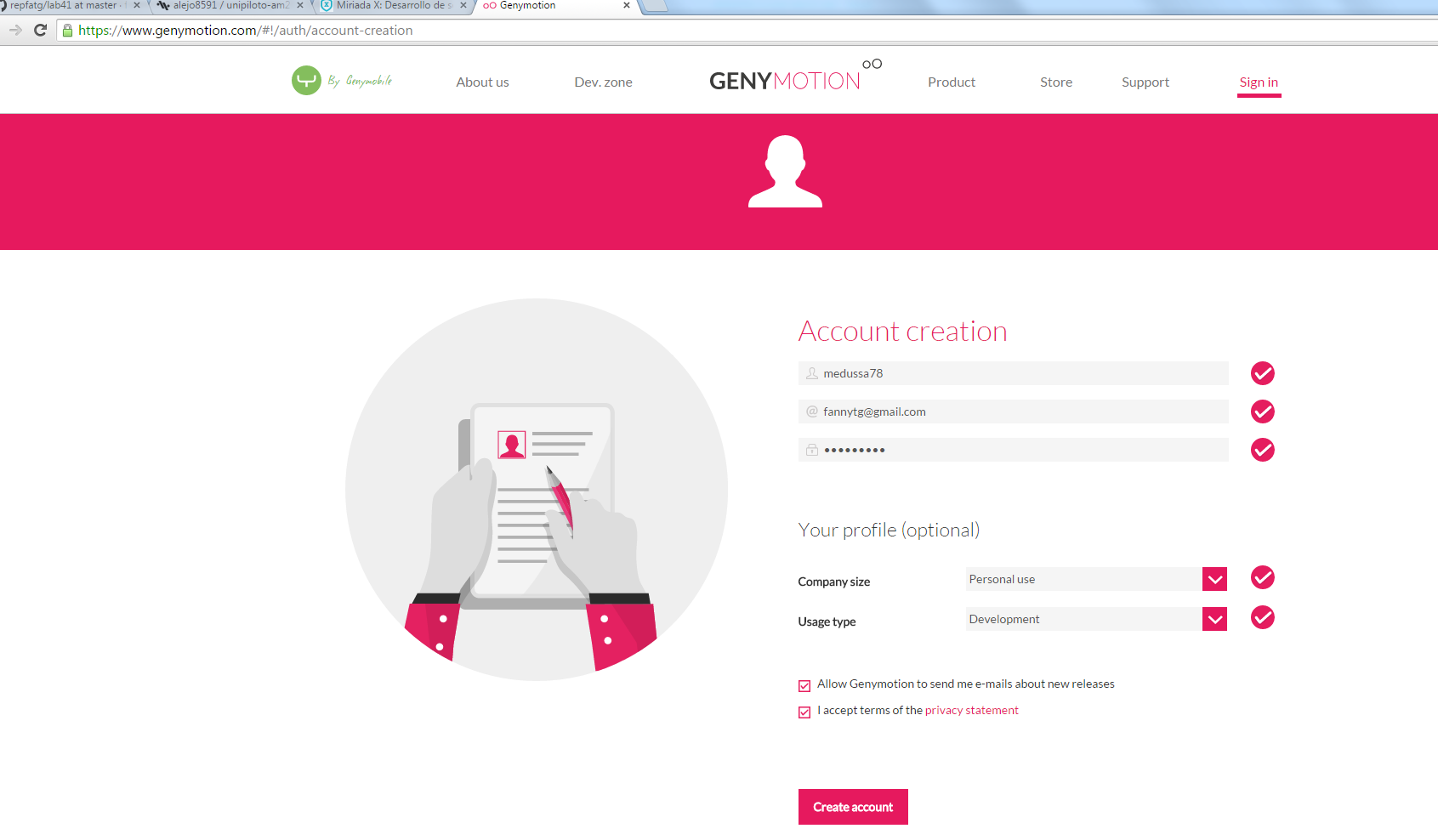
Se crea app, se agregan plataformas y se instala el plugin geolocation,

Colocar jquery



Se compila la aplicación con cordova build android y se ejecuta con run, previamente revisar si el dispositivo tiene habilitada la opción de debug por usb

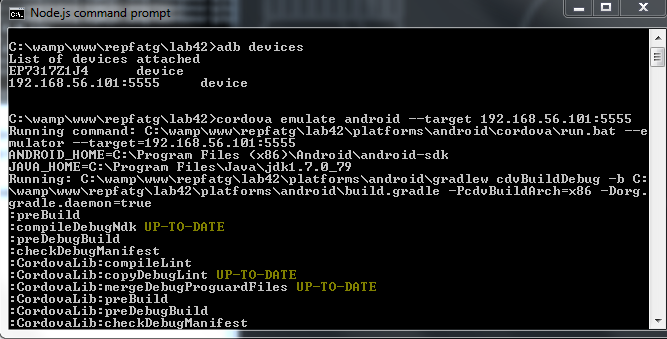
Creamos un nuevo usuario en

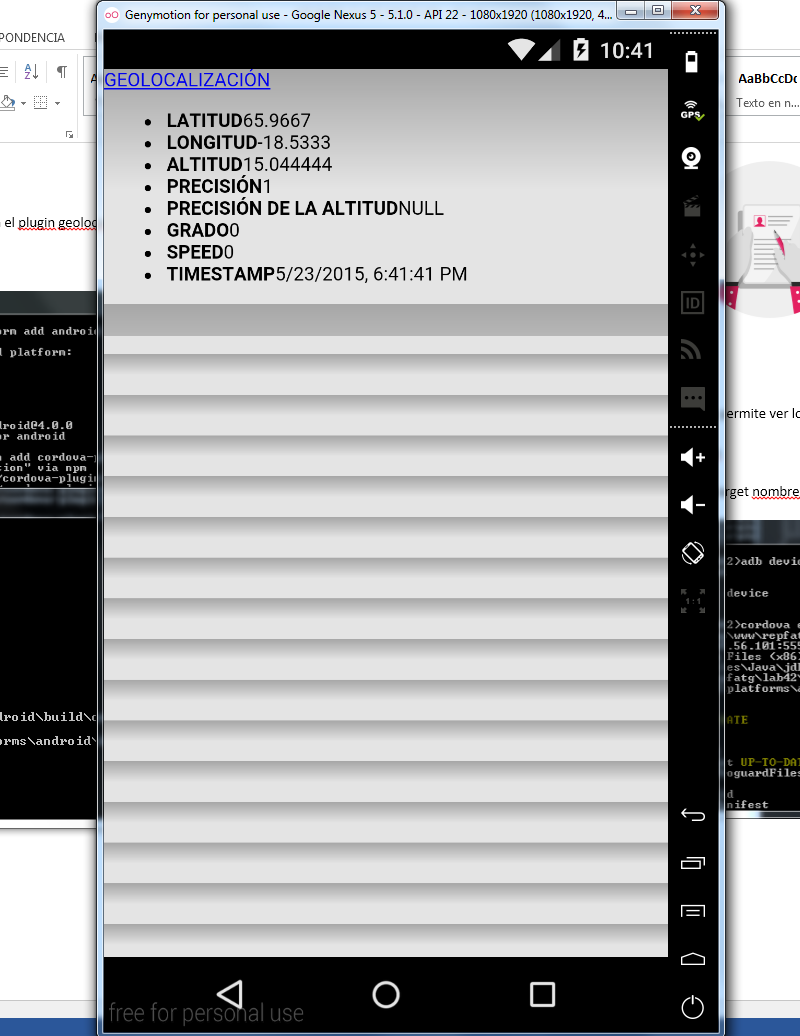


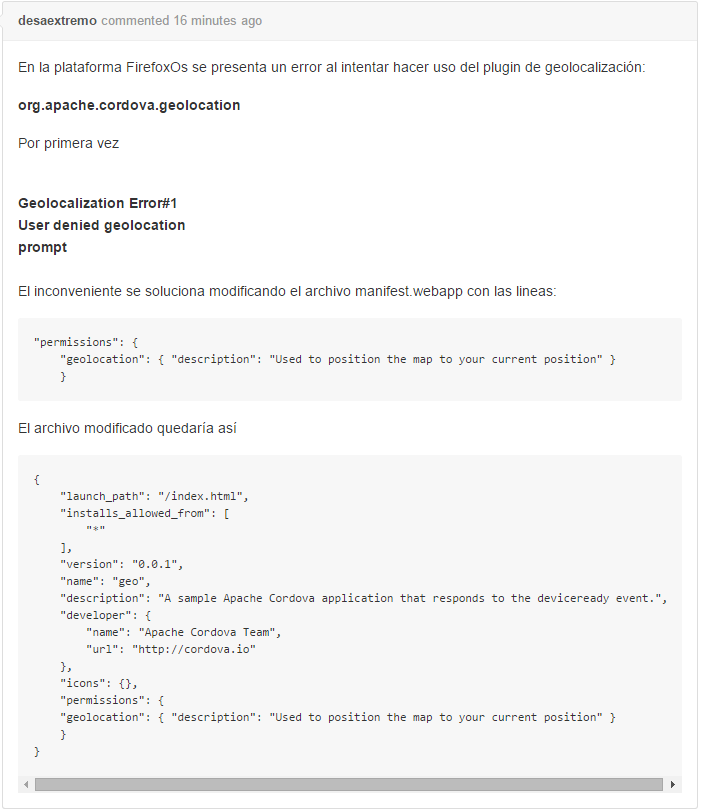
El comando abd devices me permite ver los dispositivos que están attached

Vamos a utilizar genymotion

Cordova emulate android –target nombre\_dispositivo (tal cual aparece en adb devices)







<https://www.miriadax.net/web/javascript-node-js> --> Cursos virtuales