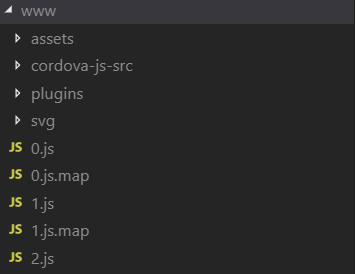
Convertir nuestra aplicación a PWA.

Configuracion de una PWA.  
Documentacion de ionic  
<https://ionicframework.com/docs/publishing/progressive-web-app>  
Corremos el comando



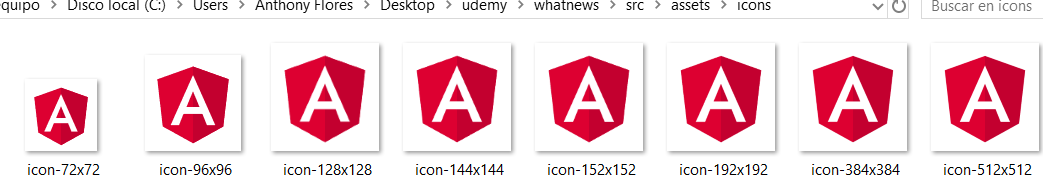
Luego:  


Nuestra carpeta www del proyecto tiene todo el código que permite ejecutar la aplicación en el navegador web. El comando de arriba permite configurar esta carpeta para que este lista para un ambiente de producción.  


Investigar http-server NODE

Ahora preparamos la aplicación para desplegarla en un hosting

Bajamos del curso el icono y se pega en src/assets/icons

En esta carpeta existen los iconos para las diferentes dimensiones, como es necesario hay un link donde se puede subir la imagen de 512x512 y este genere las demás.  


<https://app-manifest.firebaseapp.com/>

Luego de subirla se genera un zip, el cual accedemos a la carpeta icons  
app-images.zip\images\icons . Copiamos y reemplazamos estos archivos en el icons del proyecto.

Al correr el primer comando se modificó el index.html y se agrego un estilo de tema el cual hay que modificar al estilo que tenemos configurado en el proyecto



Nuestro color lo podemos encontrar en   

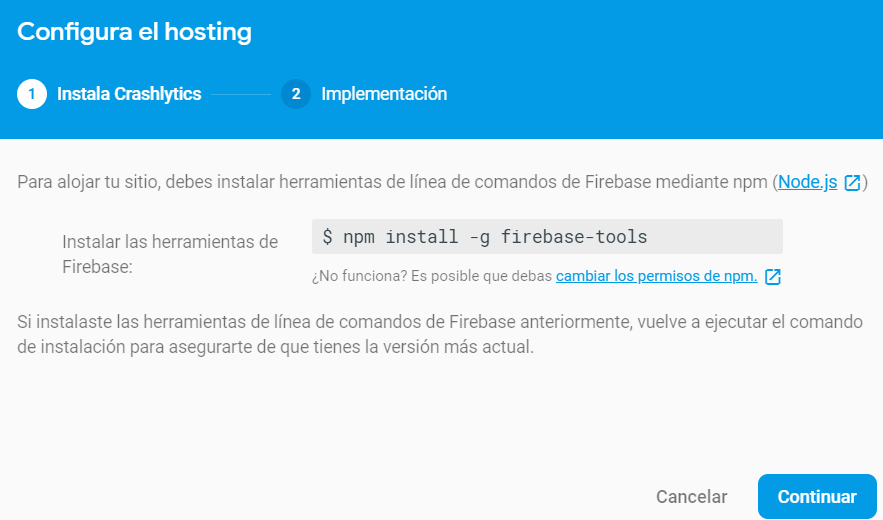



Desplegando nuestra aplicación en la nube con firebase.

<https://firebase.google.com/>

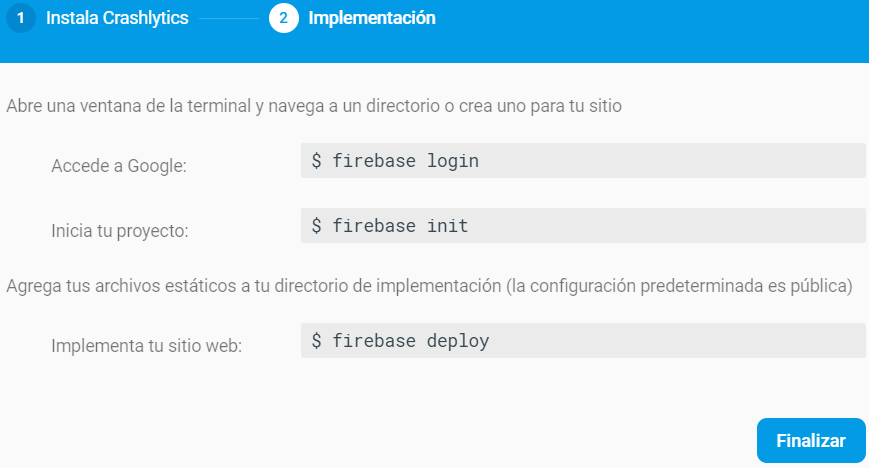
En esta pagina accedemos a go to console para ver nuestros proyectos.

Creamos un proyecto nuevo y luego nos dirigimos a hosting, comenzar lo que nos lleva a instalar por npm firebase-tools

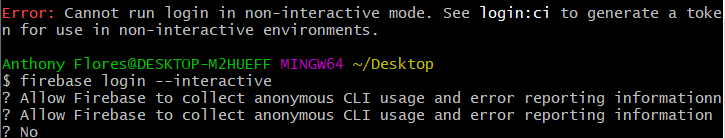


Para revisar la versión instalada firebase –v

Luego loguearnos



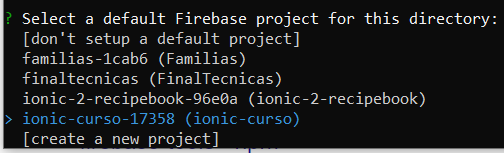
Si no funciona el login añadir al comando “—interactive”



El siguiente comando firebase init es importante ejecutarlo en el path del proyecto

Seleccionamos el tipo de configuración de nuestro proyecto en este caso hosting con firebase y luego el proyecto que creamos en la pagina de firebase.

Ojo: no funciona la navegación en el git bash. Correr cmd.



Elegimos como el directorio publico del proyecto la carpeta www





Importante no sobreescribir el index.html



En caso de que al seleccionar el proyecto de firebase no nos apareciera el creado, se selecciona no añadir aun. Y al final del proceso correr el comando:

 siendo después de add el id de nuestro proyecto que se puede ver desde la consola de firebase.

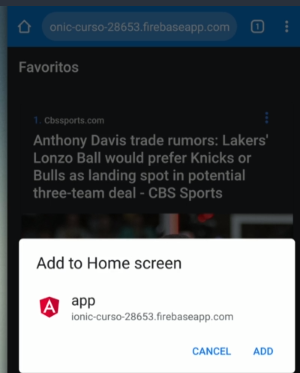
Y finalmente   


Y al final nos despliega nuestro url donde se puede acceder a la aplicación.



Corriendo la pwa en un dispositivo móvil

En estos momentos al correr la aplicación en un dispositivo la barra del navegador y el icono no se cambiaron como deseábamos.



Si realizamos cambios en la aplicación es necesario correr nuevamente el comando que la deja lista para el despliegue



Y cuando termina seguimos con

Firebase deploy  
y podremos ver todos los cambios que hicimos.

En este momento al intentar compartir nos aparece un error al intentar compartir una noticia ya que esta función es con cordova y el navegador web no lo utiliza. En pc no se podrá usar pero en el dispositivo móvil debemos utilizar un api llamado navigator share.

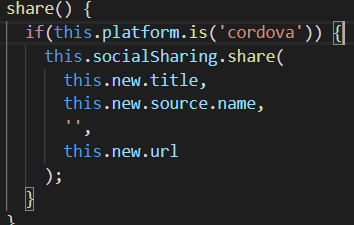
<https://developers.google.com/web/updates/2016/09/navigator-share>

Iniciamos nuevamente en proyecto   
ionic serve

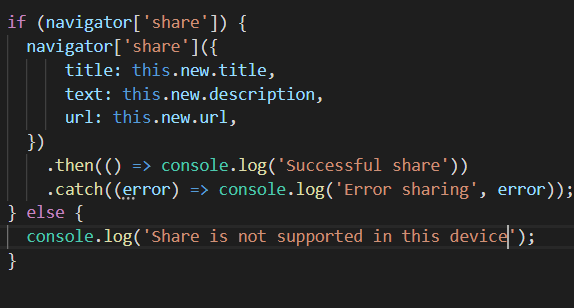
Nos ubicamos en el new component que es donde se levanta la opción para compartir.

Importamos platform en nuestro constructor para poder verificar la plataforma en que estamos corriendo la aplicación.

Modificamos la función de share



Y luego en caso que el navegador soporte “share” ejecutarlo.



Ahora corremos nuevamente el build de prod



Y firebase deploy