En este informe explicaré paso a paso lo que hice en la actividad:

Primero creé desde el dashboard de angular una aplicación, la cual nombré “Publicaciones”, elegí angular con una template de tabs, luego de haber sido creada seguí los pasos dados:

npm install -g @ionic/cli @capacitor/assets

git clone https://github.com/.... /publicaciones.git publicaciones

cd publicaciones && npm install && ionic serve

Luego cloné mi repositorio en local y comencé a trabajarlo.

Para el caso en particular, tab1 será igual a la lista de publicaciones, tab2 será una nueva publicación, se ocupará un servicio que llamaremos publication service para conectar tanto la creación de publicaciones con la lista de publicaciones y se persisten las publicaciones con un plugin de storage.

Primero veremos la creación de una publicación por lo que partiremos describiendo tab2.

Publicación se compone de los siguientes parámetros: title, subtitle, description, photo, date. La ultima es, como se ha dado en la evaluación, automática a la actual de la creación de la publicación.

Texto

Descripción generada automáticamente

Se agrega el plugin para capturar la foto

Texto

Descripción generada automáticamente

Se acotan los parámetros para permitir la validación para guardar la publicación

Texto

Descripción generada automáticamente

En el html de la Tab2, se especifica cada uno de los inputs pertinentes, se permite mostrar además la publicación recién guardada en el mismo tab2.

Captura de pantalla de un celular

Descripción generada automáticamente

Ahora que ahondamos sobre guardar la publicación, es pertinente mostrar el servicio utilizado para persistir los datos, el cual es publication service en services. En dicho publication service, basta un crud para cumplir con las necesidades de la actividad. A continuación addPublication, getPublications, deletePublication, que son las utilizadas.

Captura de pantalla de computadora

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Luego, para finalizar, esta tab1, la cual su fin es mostrar todas las publicaciones guardadas. Cabe decir que le pertenece a tab1 la función de borrar una publicación elegida, por lo que a continuación se muestra el como se implementa. Además, del Pipe para formatear la fecha que se debe mostrar en cada publicación.

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

En el html de tab1 se demuestra la implementación, además de la utilización de ngIf y ngFor.

Texto

Descripción generada automáticamente

Se utiliza además una ventana modal para confirmar el borrar la publicación escogida a continuación:

Texto

Descripción generada automáticamente

A continuación las vistas:

Tab1 sin publicaciones:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Tab2 Sin rellenar:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Tab2 al rellenar:

Una captura de pantalla de una computadora

Descripción generada automáticamente con confianza media

Luego de rellenar:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Devuelta en Tab1:

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

Al borrar:

Captura de pantalla de un celular con la imagen de un gato

Descripción generada automáticamente

Luego de borrar:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente