Crearemos una aplicación donde se almacenan quotes y se pueden agregar a una lista de favoritos.

Ionic start quotes-app blank

Debemos crear \_

Library  
Quotes  
Favorites tab  
Modal ( pregunta si se desea de favorito)  
Settings ( menú desplegable a la izquierda arriba)

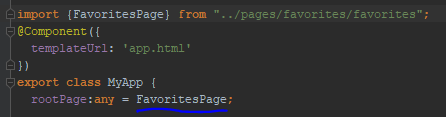
Iniciando servidor: ionic serve

Creando las paginas

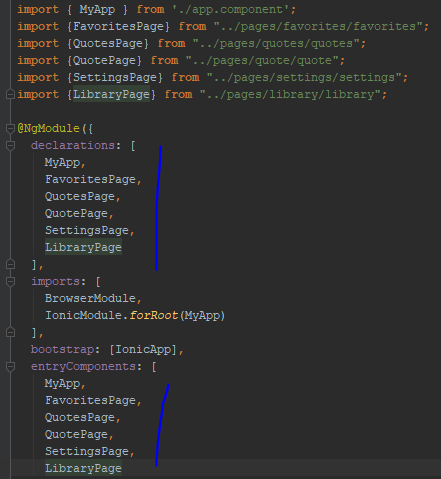
Ionic generate page library  
y asi creamos todas las otras paginas.

Borramos la carpeta home, y luego arreglaremos sobre eso.

Queremos que nuestro root page sea favorites, entonces en el app component



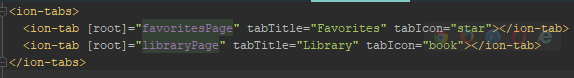
Luego hay que agregarlo al modules, junto con todas las otras paginas que vamos a utilizar.



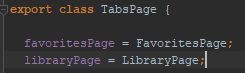
**Navigation Tab**

Creamos un componente nuevo en pages llamado tabs,

Al ser un tab no llevará la estructura completa de un html de ionic



Desde el componente creamos variables y asignamos a las paginas donde se hará el root



Al manejarnos por tabs, ahora haremos el TabsPage nuestra root en app.

**Añadiendo Contenido**

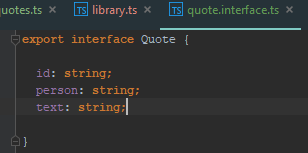
Para visualizar contenido en los quotes los traeremos de “afuera” en este caso un archivo aparte llamado quotes, que estará ubicado en una carpeta de src llamada data:

Este archivo contiene un objeto mock con estructura de quotes.  
category: 'inspirational',  
 quotes: [  
 {  
 id: '1',  
 person: 'Theodore Roosevelt',  
 text: 'Believe you can and you\'re halfway there'  
 },  
 {  
 id: '2',  
 person: 'Norman Vincent Peale',  
 text: 'Change your thoughts and you change your world.'  
 },  
 {  
 id: '3',  
 person: 'Robert H. Schuller',  
 text: 'What great thing would you attempt if you knew you could not fail?'  
 }  
 ],  
 icon: 'brush'  
},

Primero debemos desplegarlos en nuestro componente de library:

Creamos una variable que recibirá el objeto, al conocer la estructura que lleva el objeto creamos la interfaz para asignarle el tipo a la variable.

En la misma carpeta de data, creamos el quote.interface.ts



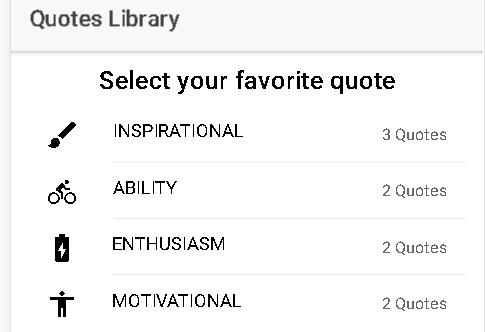
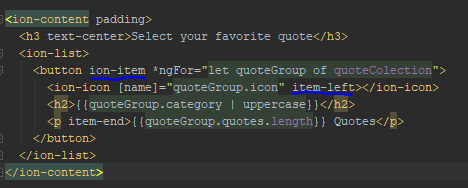
Declaramos la variable local del tipo  


Ya con esto podemos hacer uso de quoteCollection en nuestro html

En el html ponemos un titulo y una lista ,dentro botones, aunque también pueden ser ion-item

Es importante saber que al momento de usar tags existen directivas para presentarlos de una forma estándar ya definida amigable.

Utilizando las directivas ítem-left, si el elemento se encuentra dentro de un ion ítem(ion-xxx)

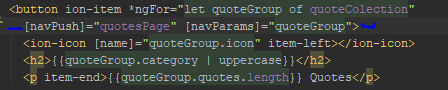


Ya tenemos presentadas las categorías, ahora vamos a acceder a los quotes que tiene cada una de ellas.

Tenemos que al momento de clickear en una categoría abrir el componente de quotes y pasar la variable **quoteCollection**.

Para esto en el html utilizamos una directiva [**navPush**] dentro del boton y la igualamos a la ruta de la pagina que deseamos abrir.

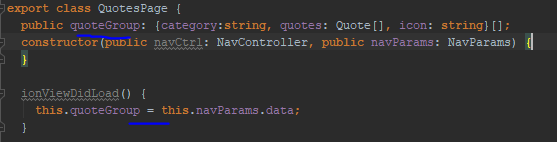
El valor que ponemos dentro del “” del navPush debe sser la variable declarada en el component.ts que recibe la clase de la pagina a ir.  
Ahora para pasar el valor de la variable **quoteCollection** utilizamos el [**navParams**]



**Recibiendo los parámetros desde el componente llamado**

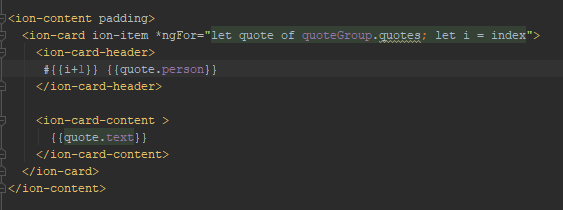
Creamos una variable del mismo tipo del objeto que vamos a recibir, haciendo la importación de la interfaz, nos aseguramos que en el constructor este declarado el NavParams.

Luego en el ionViewDidLoad() le pasamos el valor a nuestra variable local al componente lo que trae navParams.



Importante en DidLoad aparece error ya que la variable tiene valor ya cuando debería presentar asi que se puede utilizar WillLoad

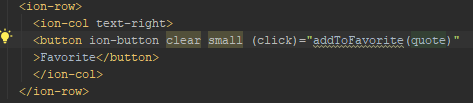
**Ionic Card**



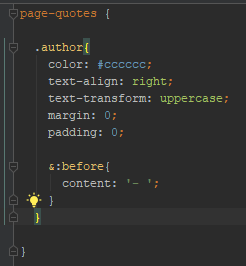
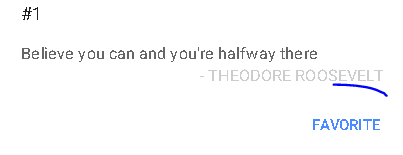
**Ionic Grid**

Ionic tiene sus propios estilos entre estos el grid, col, row que se usan de igual forma

Tambien están las directivas para poner el botón transparente, letra pequeña, etc.



Y añadiéndole estilos desde el css.

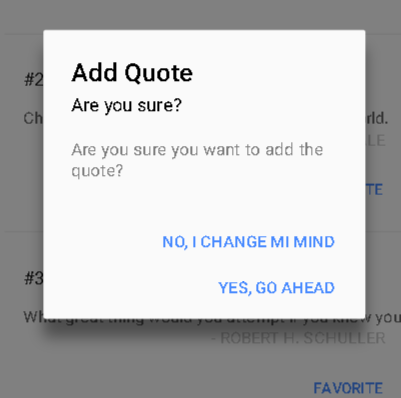
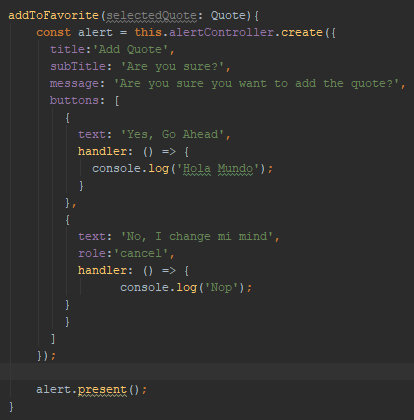


Añadiéndole la clase autor al párrafo de autor.

**Añadiendo un alert al presionar favorito**

Para implementar alertas se debe importar y declarar en el constructor el AlertController

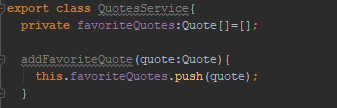
La estructura para llamar la alerta se basa en una estructura de objeto para configurar su contenido, donde podemos llenarlo con title, subtitle,message, buttons, buttons es un arreglo que puede recibir un objeto donde manejara el mensaje y acción que se hará al presionarlo.  
También puede tener un atributo rol, el cual se implementa de querer cancelar.  
Esto se hace para que angular sepa que aunque el usuario presione fuera del alert, hará esta acción para cancelar.



**Utilizando un servicio para manejar favoritos**

Añadimos la carpeta services a src

Declaramos una variable favoriteQuotes arreglo de tipo Quote[] de la interfaz creada anteriormente

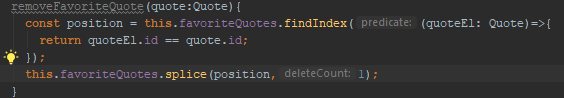


Para borrar el quote de favoritos necesitamos su posición, por lo que declaramos una constante position y se le asigna el valor que traiga la función findIndex del arreglo de favoriteQuotes que tenemos.

Dentro del findIndex en sus parámetros esta espera otra función, una función flecha donde usaremos la variable quoteEl para comparar el id del quote buscado si existe su índice.



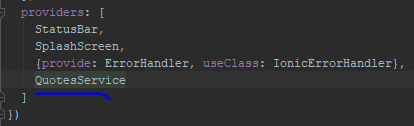
Luego de obtener la posición eliminamos utilizando la función splice pasándole como parámetro posición y cantidad de elementos ósea 1



El método getFavoriteQuotes se usara para retornar la lista de quotes favoritos pero utilizando el slice() sin parámetros

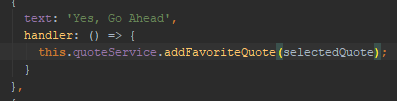
**Implementando el servicio**

Una vez hecho el servicio es necesario “proveerlo” al ser un provider hay que agregarlo en el app modules.ts



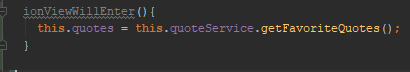
Ahora ya podemos acceder a la información que provee el servicio simplemente inyectándola en el constructor del componente donde vamos a usarlo.

En este caso el quotes.  
Añadimos ahora un quote a favorito utilizando la variable y llamando a la función del servicio pasándole como parámetro el quote elegido.

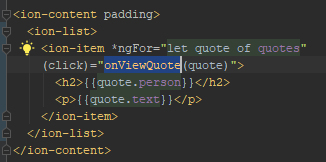


**Preparando la página de favoritos**

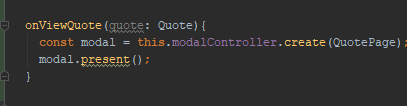
Desde el componente favorites creamos la variable quotes de tipo Quote, y en el constructor traemos el servicio QuotesService para luego asignarle el valor que trae la función de get favoritequotes



Luego en el html mostraremos la variable en una lista. Y para llamar al modal que mostrara se incluye un click



Y en el ts , creamos la función utilizando el modalController:



En ese html solo se mostrara una parte, al hacer click se desplegara un modal con el quote completo. Esto desde Quote.component

