

Proyecto Ionic: Recetas Realfood



Entornos de Desarrollo

Yaiza Muñoz

05/06/2018

DW31

Tabla de contenido

Introducción2

1. Listar Recetas3

2. Mostrar Recetas4

3. Añadir Recetas5

4. Cambiar Estilos de la app de Ionic6

5. Eliminar Recetas8

6. Crear alertas9

7. Información sobre Realfooding10

Introducción

Para realizar este proyecto lo primero que he realizado ha sido crear un proyecto nuevo en ionic.

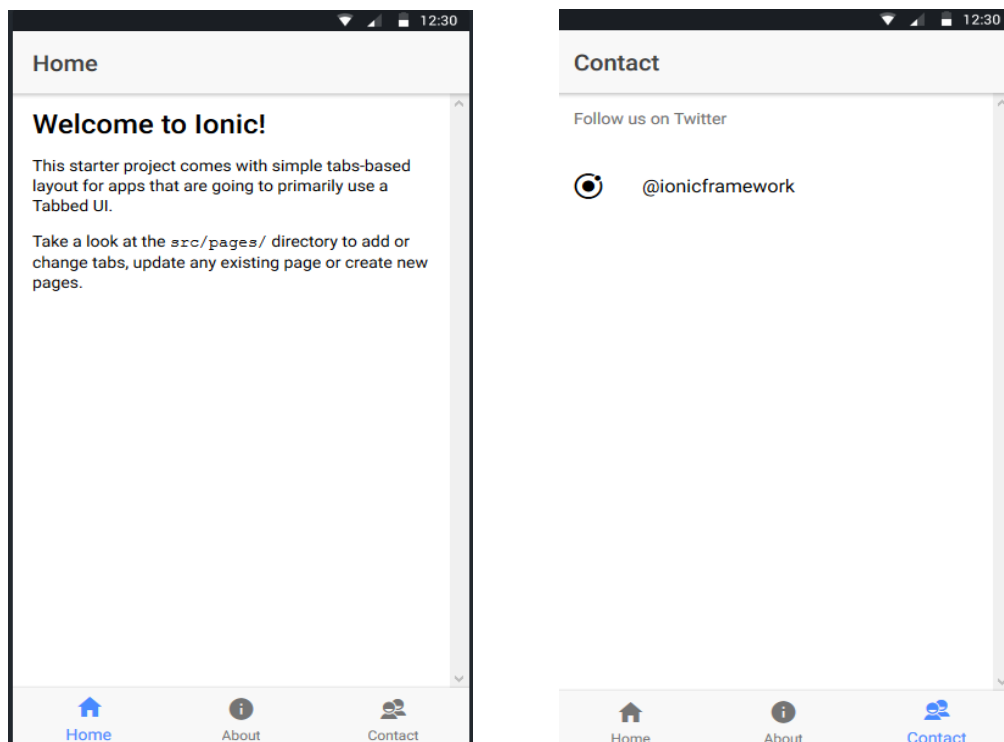
Para crearlo desde la terminal del proyecto el proyecto llamado recetas he ejecutado lo siguiente:

`Ionic start recetas tabs` → Modo de ionic con tabs predeterminadas



Nombre del proyecto

El proyecto por defecto que se crea es el siguiente:



Una vez hecho esto, lo que realizaré será una lista en la que salen recetas, y dentro de ellas saldrá la descripción de la misma. Además de poder añadir más a la lista y poder eliminar recetas de la lista.

En conclusión, las funciones que implantaré en la app son:

1. Listar recetas
2. Mostrar recetas
3. Añadir recetas a la lista de recetas
4. Cambiar Estilos
5. Eliminar Recetas

6. Crear alertas
7. Información sobre Realfooding

1. Listar Recetas

Para hacer esto lo primero que empecé a modificar para dejar solo uno en el que me quedara lista, la cual la he realizado en el tabs de home (Lista) a través de json.

La página de tabs queda finalmente de la siguiente manera:

```
1 <ion-tabs>
2   <!-- tabs para lista -->
3   <ion-tab [root]="tab1Root" tabTitle="Lista" tabIcon="list"></ion-tab>
4   <!--saca el tab con el icono y su nombre que es lista -->
5
6 </ion-tabs>
7
```

```
1 import { Component } from '@angular/core';
2
3 import { HomePage } from '../home/home';
4
5 @Component({
6   templateUrl: 'tabs.html'
7 })
8
9 export class TabsPage {
10
11   //hacia donde redirige el tab 1
12   tab1Root = HomePage;
13
14   constructor() {
15
16   }
17
18 }
```

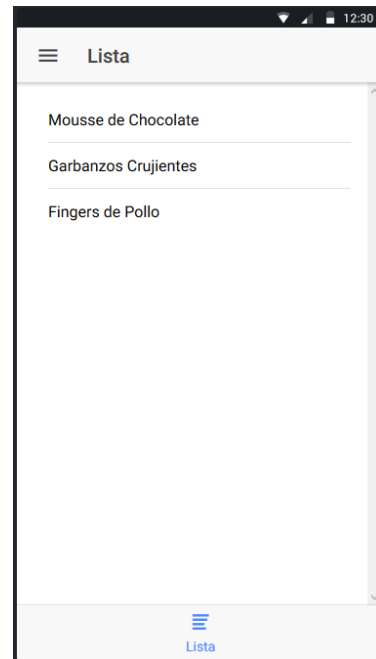
En el código de home.ts la lista queda de la siguiente manera:

```
// array de recetas vacío + todas las recetas en json
recetas: Array<any> = [
{
  nombre: "Mousse de Chocolate",
  foto: "../../assets/imgs/moussedechocolate.jpg",
  ingredientes: "150gr de chocolate puro, 1 sobre de agar agar, 3 huevos, 10 gr de pistachos",
  descripción: "Fundimos el chocolate al baño maría, añadimos los huevos y removemos bien hasta que se enfrie todo. Ponemos las claras al punto de nieve y vamos mezclándolo todo poco a poco. Una vez hecho
},
{
  nombre: "Garbanzos Crujientes",
  foto: "../../assets/imgs/garbanzos.jpg",
  ingredientes: "400gr de garbanzos, 1 chorrito de aceite de oliva, 2-3 cucharadas de especias al gusto",
  descripción: "Precalentar el horno a 200°C con el ventilador y calor arriba y abajo.Si los garbanzos son de bote, lavarlos y dejarlos bien escurridos. Si los hemos cocido nosotros, simplemente dejar escurr
},
{
  nombre:"Fingers de Pollo",
  foto:"../../assets/imgs/fingers.jpg",
  ingredientes:"300gr de pechuga de pollo, 1 huevo, 20gr de harina de almendras, 30gr copos de maíz, pimentón dulce, oregano, cebolla en polvo, curcuma y una pizca de sal",
  descripción: "Precalentamos el horno a 180º. Cortamos la pechuga de pollo limpia en tiras. En un plato hondo o cuenco batimos el huevo o ponemos las claras. En otro recipiente ponemos los copos de maíz tri
}
]
```

En el home.html:

```
46
47 <ion-list>
48   <!-- Hacer un for que muestre n item de las recetas con la opción de "Elegir receta" -->
49   <button ion-item *ngFor="let item of recetas" (click)="recetaElegida(item)">
50     {{ item.nombre }} <!--nos lista todas las recetas -->
51   </button>
52 </ion-list>
53
```

Junto al json del home.ts y el código que ahí vemos en el home.html que se encarga de listar los nombres de las recetas en el home nos quedaría la app de la siguiente manera visualmente:







2. Mostrar Recetas

Una vez que la lista de recetas ya está hecha, nos centraremos en el siguiente paso, el cual será sacar la información de cada receta. Para ello crearemos una nueva página en la app que en este caso la he llamado description.

Para realizar esto de manera automática y que se cree directamente toda la estructura, desde la línea de comando de Node.js si me situó en mi proyecto puedo crearla con el comando:

`Ionic generate page description`

La estructura que nos saca es la siguiente:

 description.html	28/05/2018 11:09	Firefox HTML Doc...	1 KB
 description.module.ts	18/05/2018 13:28	Archivo TS	1 KB
 description.scss	18/05/2018 13:28	Archivo SCSS	1 KB
 description.ts	28/05/2018 11:10	Archivo TS	1 KB

Para sacar la info de cada receta en home.ts he creado una función llamada “recetaElegida(item)” que sirve para que cuando hagamos click en la receta nos muestre la página de descripción con toda la receta. El código de la función es:

```
// pasar una receta a la descripcion para que se abra la description page y la receta en si
// haciendo le clic a la receta elegida

recetaElegida(item){
  this.navCtrl.push( DescriptionPage, {item: item});
}
```

Después, en la DescriptionPage, que es a la que manda el ítem el html nos queda de esta manera, de tal forma que aparezcan todos los ítems de la receta.

```

<!--CONTENIDO DE DESCRIPCION DE RECETA-->
<ion-content padding>
  <!--busca la imagen de la receta y si no la encuentra pilla una por defecto-->
  
  <ion-list>
    <p>{{ receta.ingredientes }}</p><br> <!--saca los ingredientes de la receta-->
    <p>{{ receta.descripcion }}</p><!--saca la descripción de la receta-->
  </ion-list>
</ion-content>

```

Por lo que la página de descripción con el ítem seleccionado nos quedaría así:

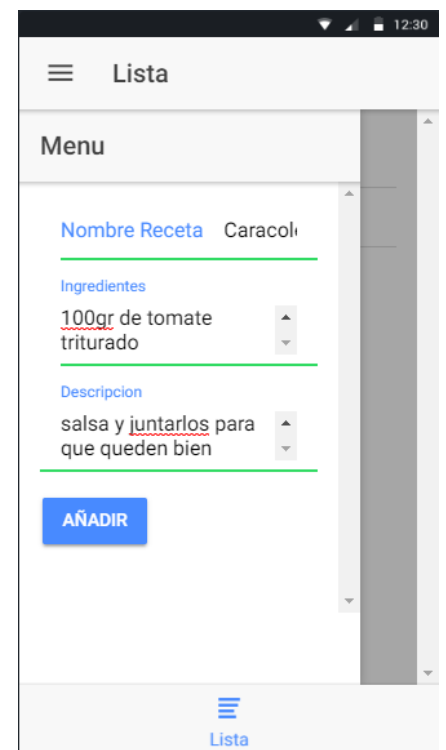


En este caso yo he elegido la receta llamada Mousse de Chocolate, por lo que al realizar click en ella nos aparecerá la imagen del mousse de chocolate con los ingredientes más la receta a seguir.

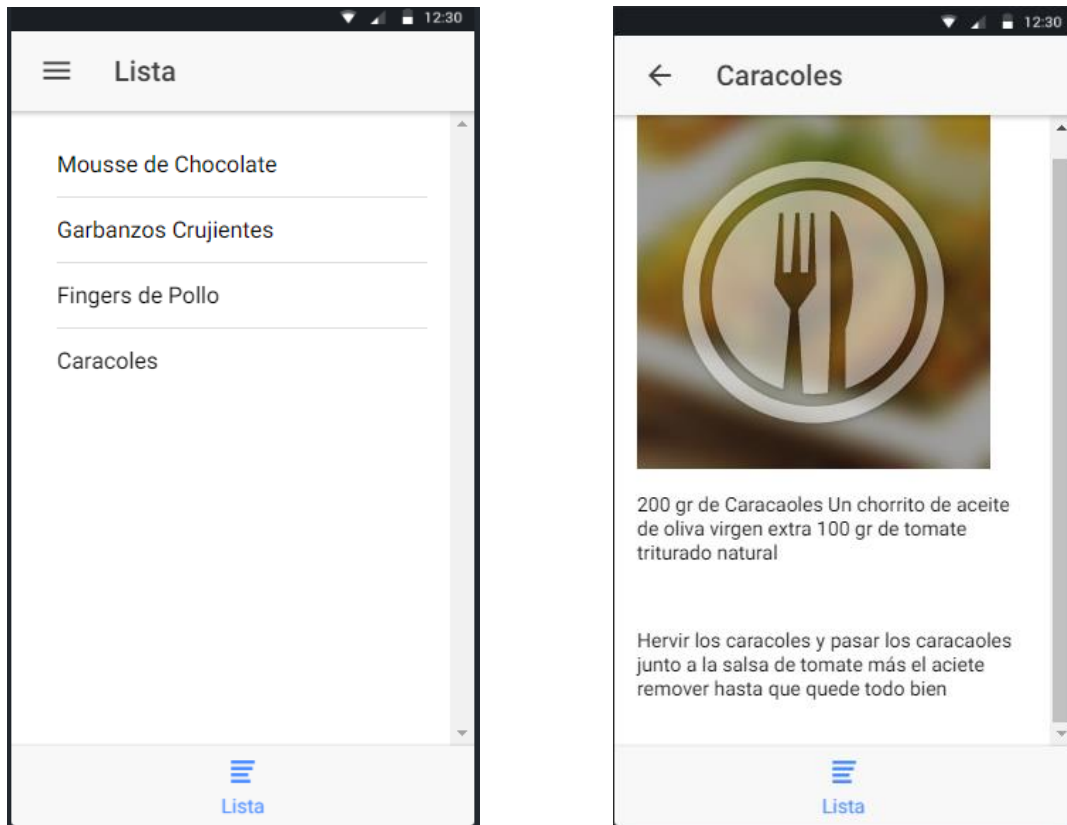
3. Añadir Recetas

Una vez que ya tenemos la descripción de la lista hecha, el siguiente paso es el de añadir recetas al array de recetas.

En este caso he hecho un sidemenu para poder añadir las nuevas recetas al array. Por lo que desde el menú hamburger podremos acceder al sidemenu donde se encuentran los inputs a rellenar.



Aquí podemos ver como he escrito la receta, y después si le diésemos a añadir este se sumaría a la lista. Con lo que, la lista y la descripción quedarían de la siguiente manera:



Añadir que la foto que tenemos en estos mismos instantes es una por defecto que se establece en el espacio de foto si no encuentra ninguna otra.

4. Cambiar Estilos de la app de Ionic

Para cambiar los estilos básicos de ionic y que la app se ponga de la forma y color que nosotros queramos lo primero que se ha de hacer es mirar en ***src/theme/variables.scss*** los colores que ya tenemos predefinidos.

```
$colors: (  
  primary:    #488aff, //azul  
  secondary:  #32db64, //verde  
  danger:     #f53d3d, //rojo  
  light:      #f4f4f4, //gris clarito  
  dark:       #222    //negro  
);
```

En este caso, yo quería usar el verde, por lo que no he necesitado añadirle nada más al scss.

En cambio, para ver esto reflejado en nuestro proyecto, tenemos que dirigirnos a todos nuestros .html de nuestra aplicación. Desde ahí editaremos todo lo que nosotros queramos cambiar.

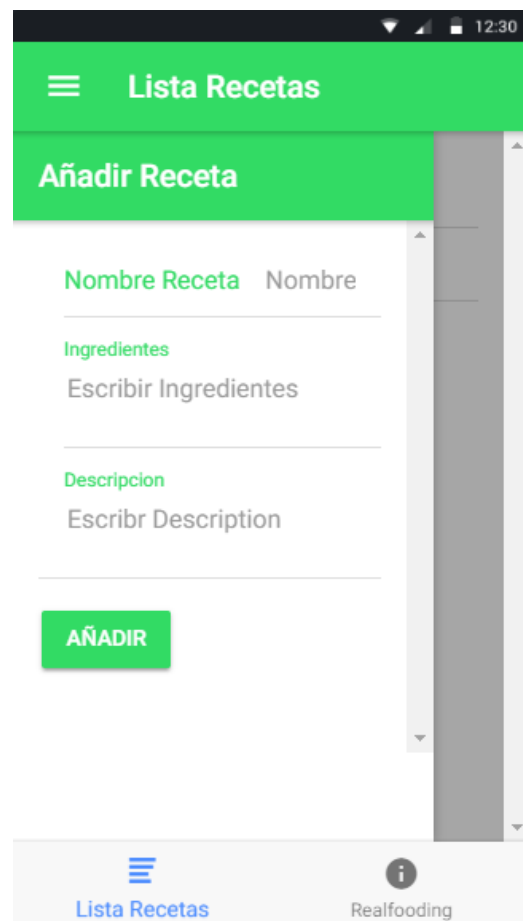
En mi caso, lo que mayormente he cambiado han sido los navbar, toolbar y el nombre de los inputs al añadir nueva receta.

```
<ion-navbar color="secondary"> <!--cambio de estilo al navbar -->
  <button ion-button menuToggle icon-only>
    <ion-icon name='menu'></ion-icon>
  </button>
  <ion-title>Lista</ion-title>
</ion-navbar>
```

```
<ion-toolbar color="secondary"> <!--cambio de estilo al toolbar-->
  <ion-title>Menu</ion-title>
</ion-toolbar>
```

```
<ion-item>
  <ion-label color="secondary">Nombre Receta</ion-label> <!--cambio de estilo al color de los labels de los inputs-->
  <ion-input [(ngModel)]="nombre" name="nombre" placeholder="Nombre"></ion-input>
</ion-item>
```

Una vez realizados estos cambios, podemos decir que el estilo de la app quedaría de la siguiente manera.



5. Eliminar Recetas

Para eliminar recetas en home.ts he añadido una nueva función llamada eliminar recetas, lo que hace esta es elegir el ítem que vamos a eliminar de la lista de recetas y después lo eliminaremos.

```
//con index of elegimos el item de this.recetas que es el array
//con splice le decimos que queremos borrar ese item al que le hemos llamado index y que solo ese

eliminarReceta(item){
  const index: number=this.recetas.indexOf(item);
  (this.recetas).splice(index, 1);
}
```

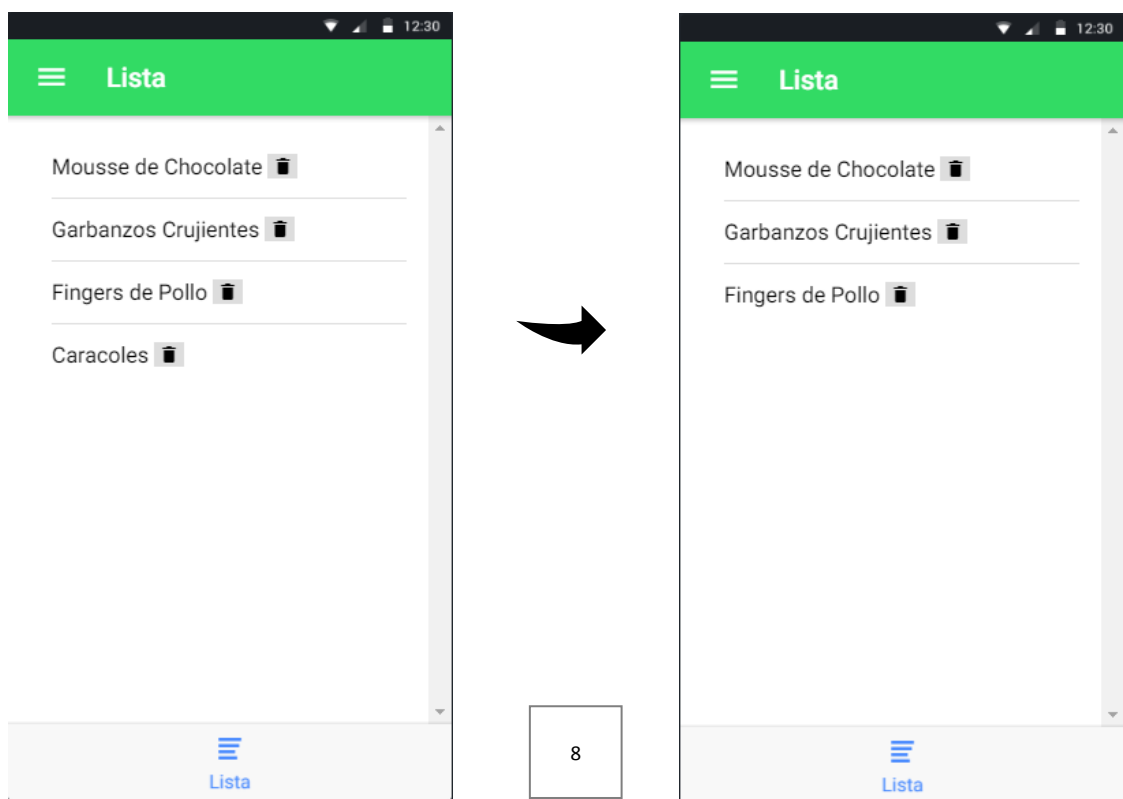
Una vez tenemos la función creada, nos dirigiremos a la parte del html en la cual incluiremos el icono de basura, simulando la eliminación del ítem.

Aplicando esta función lo que finalmente nos queda en el html de listar la recetas es lo siguiente:

```
<ion-list>
  <!-- Hacer un for que muestre un item de las recetas con la opcion de "Elegir receta" -->
  <div ion-item *ngFor="let item of recetas" (click)="recetaElegida(item)">
    {{ item.nombre }} <!--saca el nombre de las recetas de la lista-->
    <button (click)="eliminarReceta(item)"><ion-icon name="trash"></ion-icon></button>
    <!--sirve para hacer clic en el icono de basura y que lleve a la funcion de eliminar-->
  </div>
</ion-list>
```

Si nos vamos a la lista, vemos que las recetas son las siguientes, y que nos sale el icono de la basura al pie del nombre de la receta:

Una vez que le damos al icono de la basura, esta receta se elimina y nos dejaría con las otras tres recetas que tenemos en la lista.



6. Crear alertas

Uno de las ultimas funciones que he añadido ha sido la de crear alertas una vez añadido la receta.

Es cierto que la función de añadir recetas ya la tenía creada, pero después de haber hecho todas las demás funciones anteriores se me vino a la mente el poder añadir una alerta en la app para que de esta manera estuviera más claro que la nueva receta se había añadido a la lista.

Por lo que en home.ts he creado la función, que se basa en crear una alerta mediante alertController en la cual podemos definir los elementos que esta va a tener via json.

```
//creamos una alerta con los siguientes parametros dentro
//para poder utilizarla después al añadir receta

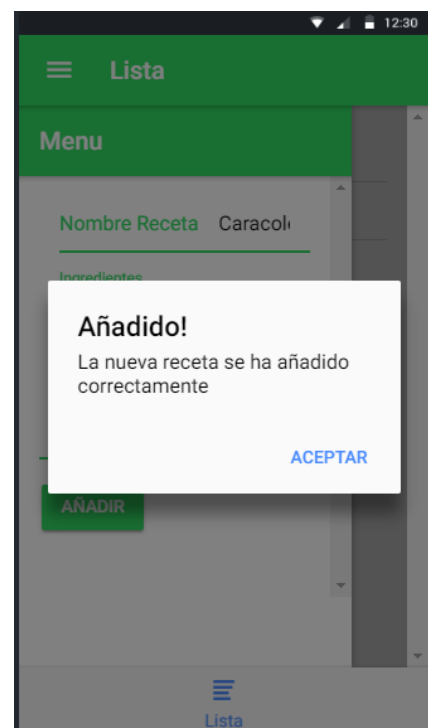
presentAlert() {
  let alert = this.alertCtrl.create({
    title: 'Añadido!',
    subTitle: 'La nueva receta se ha añadido correctamente',
    buttons: ['Aceptar']
  });
  alert.present();
}
```

Una vez tenemos la función esta la tenemos que ejecutar en el botón de añadir que tenemos creado en el sidemenu.

```
<button color="secondary" ion-button (click) = "crearReceta()" (click) = "presentAlert()">Añadir</button>
```

En este botón podemos ver que al hacer click en él nos está creando la receta, y, además, nos enseña la alerta de que nuestra receta se ha creado correctamente.

Visualmente en la app el alert nos aparecería de esta forma:



7. Información sobre Realfooding

Para adjuntar la información sobre en lo que consiste el Realfooding he añadido un nuevo tabs en el que voy a proporcionar toda la información desde ese nuevo tab.

El nuevo tab se va a llamar Realfooding y como icono va a tener una i de información.

En este nuevo tab vamos a tener información respecto a que significa el movimiento realfooding, que productos son comida real y que productos son ultraprocesados.

La información la he metido en un array en info.ts llamado información, el cual queda de la siguiente manera:

```
//array con informacion para realfooding

informacion: Array<any> = [

  {
    descripcion: "El Realfooding es un estilo de vida basado en comer comida real y evitar los ultraprocesados. Es un ultraprocesados: "1- Refrescos, 2- Bebidas energéticas, 3- Zumos envasados, 4- Lácteos azucarados, 5- Bollería, 6
    foto: '../assets/imgs/realfooding.jpg'
  }
]
```

Esta información la tenemos que mandar a info.html para que se haga visible. Por lo que, en info.html realizaremos un for llamando a este array. Y, de esta manera, visualizaremos la información.

```
<ion-content padding>
  <ion-list *ngFor="let item of informacion"><!--recorre el array informacion-->

    <!-- saca la infromacion del array de infromacion-->

    <p>{{ item.descripcion }}</p><br>
    <b>COMIDA REAL:</b><br>
    <br>
    <b>ULTRAPROCESADOS A EVITAR:</b><br>
    <p>{{ item.ultraprocesados }}</p>

  </ion-list>
</ion-content>
```

Una vez tenemos todo esto hecho, en el emulador de la aplicación nos aparecerá así:

Realfooding ¿Qué es?

ultraprocesados. Es un movimiento que defiende el derecho a una alimentación saludable para la población. Es una revolución que lucha con conocimiento y conciencia contra el lado oscuro de la industria alimentaria

COMIDA REAL:



ULTRAPROCESADOS A EVITAR:

1- Refrescos, 2- Bebidas energéticas, 3- Zumos envasados, 4- Lácteos azucarados,

Lista

Realfooding