

## Día 3

### Conceptos

- Instalar Ionic y Cordova: *npm install -g ionic cordova*
- Proyecto base de Ionic 4 / Proyecto base de Angular
- Patrón modelo-vista-controlador
- Typescript
- Introducción a Cordova y plugins
- Componentes. Desarrollo orientado a componentes.
- Documentación de Ionic: Quick start, componentes visuales y API

### Estructura de un proyecto Ionic 4

- Crear un proyecto base con:  
*ionic start explanation-app blank --no-git*
- Configurar el entorno de desarrollo (general, no concreto del proyecto actual) para que en Typescript, HTML y SCSS las tabulaciones sean de 2 espacios.
- Es un proyecto NodeJS (*package.json*, *node\_modules*)
- Es un proyecto Webpack (transpila el código fuente de *src/* en *www/*)
- Es un proyecto Angular
- Es un proyecto Cordova (desde que se añade alguna plataforma).
- Evitar algunos de los warnings de la consola añadiendo al *app.component.ts*:

```
if(platform.is('cordova')) {
```

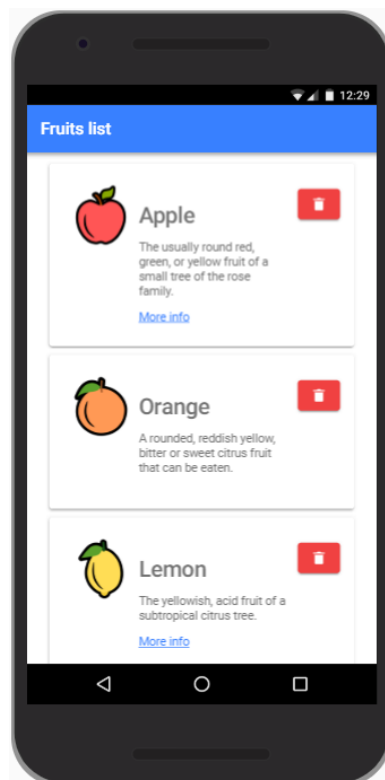
### Typescript

- Abrir *src/app/home/home.page.ts*
- Tipos básicos: boolean, number, string, array, enum, any, null
- Interfaces: definición de contratos de clases, pero también usadas para definir modelos de datos
- Clases: atributos y métodos, getters y setters herencia, implementación de interfaces, modificadores de acceso, palabra clave *this*

## Estructura de un proyecto Angular

- Punto de entrada de la aplicación *main.ts*
- Módulo principal de la aplicación AppModule
- Componente principal de la aplicación AppComponent (controlador y vista)
- Módulo de definición de rutas de la aplicación AppRoutingModule
- Módulo de la página home de la aplicación HomePageModule y su componente HomePage (controlador y vista).
- Carpeta assets
- Carpeta environment
- Themes
- *angular.json*

## Ejemplo de aplicación de lista de frutas



### Explicación 1

- Copiar los datos de las frutas contenidos en mocks al controlador principal y elaborar una maqueta "estática" usando ion-header, ion-content, ion-card...

### Explicación 2

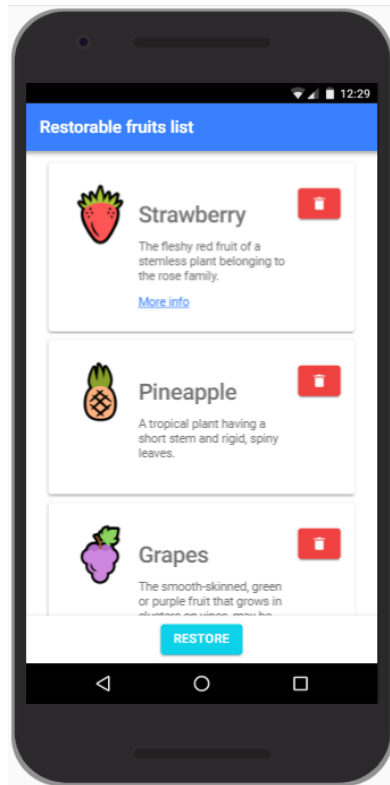
- Ciclo de vida de un componente: *ngOnInit* y *ngOnDestroy*.
- Crear una variable para almacenar cada una de las propiedades de la manzana, Inicializarlas en *ngOnInit* y parametrizar en el HTML cada uno de los atributos.
- Data binding usando `{{ expr }}` en los contenidos, `attr="{{ expr }}"` en los atributos y también `[attr]="expr"`
- Crear una variable para almacenar un array de frutas, inicializarla en *ngOnInit* y parametrizar en el HTML cada uno de los atributos. En el IDE se puede contraer el array con el fin de hacer el código más legible.

### Explicación 3

- Directiva `*ngFor`
- Preparar el HTML para que itere y muestre las frutas usando la directiva `*ngFor`
- Directivas `*ngIf` y `[hidden]` y sus diferencias
- Diferencia entre directivas estructurales y "básicas"
- Eliminar el link de algunas de las frutas y mostrarlo sólo en aquellas que lo tengan disponible.
- Gestionar el evento click sobre los botones de eliminar fruta. Cuando se pulse sobre alguno de ellos, se debe mostrar por consola el índice de la fruta que corresponda.

## Ejercicio 1

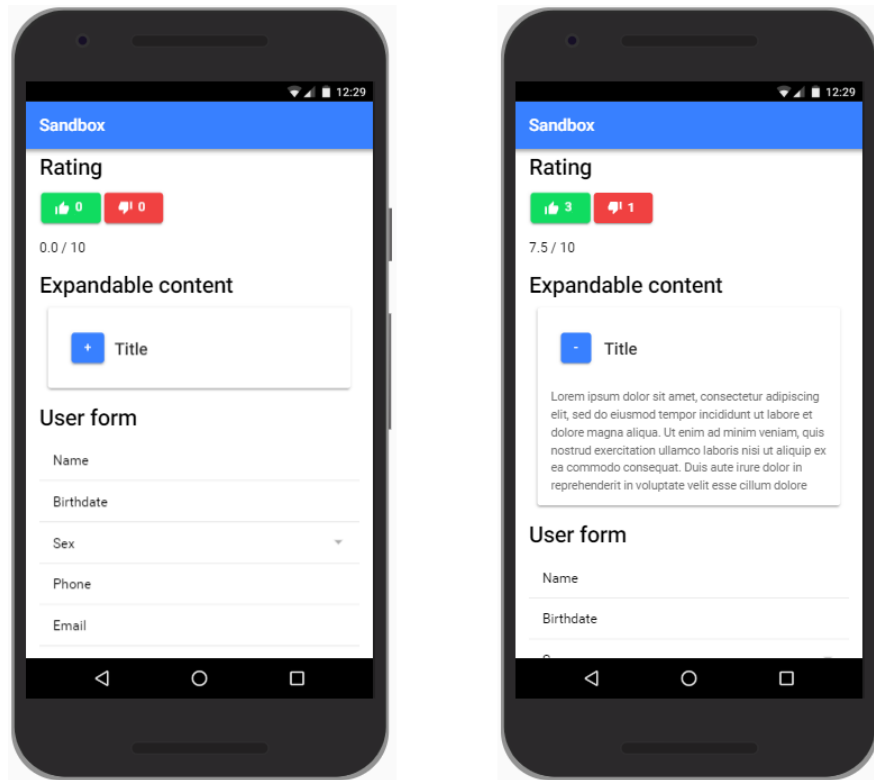
Amplíe la aplicación de explicación anterior con la funcionalidad que se indica a continuación:



1. Dote a los botones de eliminar de funcionalidad: cuando se pulse, se eliminará la fruta correspondiente del array y, consecuentemente, este cambio se reflejaría en la vista.
2. Incluya también un footer fijo con un botón de restaurar. Si todas las frutas están disponibles en la lista (no se ha eliminado ninguna), este botón estaría deshabilitado. En caso contrario, el botón estaría habilitado y al pulsarlo se restauraría la lista de frutas original.

## Ejercicio 2

Practique con el uso de componentes de Ionic creando la siguiente pantalla:



1. La sección *Rating* debe disponer de 2 botones: el verde representa un *like*; el rojo un *dislike*. Cada vez que se pulse, se incrementará el contador que incluye. También se actualizaría la media de votos que se indica abajo. Si no se ha realizado ninguna votación, se indicará 0; si no, se indicará la media aritmética de los votos, considerando que *like* cuanta como 10 y *dislike* como 0.
2. La sección *Expandable content* debe contener un panel que muestre u oculte un determinado contenido al pulsar sobre el botón de su cabecera. Este botón tendrá el símbolo + si está contraído; el símbolo - si está expandido.
3. La sección *User form* contendrá un formulario con los siguientes campos: nombre de tipo texto, fecha de nacimiento de tipo fecha, sexo de tipo desplegable (con valores masculino y femenino), teléfono de tipo número y email de tipo email.