

Día 4

Conceptos

- Instalar Ionic: *npm install -g cordova ionic*
- Proyecto base de Ionic
- Patrón modelo-vista-controlador
- Componentes
- Typescript
- Cordova y plugins
- Desarrollo orientado a componentes web. Vista general de los componentes web existentes en Ionic.
- Directivas

Proyecto base de Ionic

- Crear un proyecto de base:
ionic start ExplanationApp blank --no-git
- Ejecutar el proyecto:
ionic serve
- Es un proyecto NodeJS (*package.json*, *node_modules*)
- Es un proyecto Webpack (transpila el código fuente de *src/* en *www/*)
- Es un proyecto Cordova (desde que se añade alguna plataforma).

Typescript

Tipos básicos

```
let isActive: boolean = false
let index: number = 1
let sentence1: string = `Hello my name is ` + name
let sentence2: string = `Hello my name is ${name}`
let evenNumbers: number[] = [2, 4, 6, 8]
enum State { Loading, Ready, Error }
let currentState: State = State.Loading
```

```
let something: any = 5
null
undefined
```

Declaración de variables

```
var a = 1 // Ámbito de función
let a = 1 // Ámbito de bloque
const a = 1 // Ámbito de bloque, sin poder cambiar su valor
```

Funciones

```
function sum(a: number, b: number): number {
    return a + b;
}
const sum = (a: number, b: number): number => {
    return a + b;
}
let a = 1 // Ámbito de bloque
```

Importación y exportación

```
import/export
```

Interfaces

```
interface User {
    name: string,
    age: number,
    date: Date
}
```

Clases

Atributos y métodos

extends: Herencia

implements: Implementación de una interfaz

constructor: Constructor

private, public, abstract: Modificadores de acceso

this

Explicación sobre un proyecto vacío

- Crear un proyecto de ejemplo:

ionic start ExplanationApp blank --no-git

- Configurar Webstorm: que no transpile TS ni SCSS automáticamente
- Quizás se requiera instalar explícitamente tslint:

npm install --save tslint

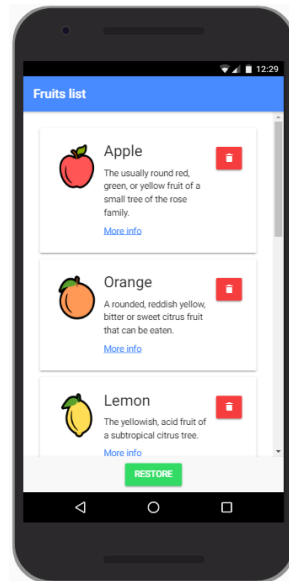
- Es un proyecto NodeJS (*package.json*, *node_modules*)
- Es un proyecto Webpack (transpila el código fuente de *src/* en *www/*)
- Es un proyecto Cordova (desde que se añade alguna plataforma). Instrucciones:
 - o *ionic start*
 - o *ionic serve*

- Carpeta *assets*, página *HomePage*.

- Añadir al *app.component.ts*:

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

Ejemplo de lista de frutas



Explicación 1

- Copiar los datos de las frutas del mockServer al controlador principal y elaborar una maquetación “estática” usando card, grid, ion-footer, etc.

Explicación 2

- Ciclo de vida de un componente (*ngOnInit*, *ngOnDestroy*) y de una página (*ionViewDidEnter*, *ionViewDidLeave*).
- 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 *isRestoreAllowed* que determinará si el botón de restaurar del pie estará o no activado. En principio estaría desactivado.
- 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 las frutas usando la directiva `*ngFor`

- Directivas `*ngIf` y `[hidden]`
- 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 y restaurar, que muestren por pantalla índice de la fruta que se pulsa y el texto “Restaurar” respectivamente.

Explicación 4

- Si la lista está completa, el botón de restaurar estaría deshabilitado, habilitado en caso contrario.
- Cuando se pulse sobre el botón borrar de una fruta, se debe eliminar de la lista.
- Cuando se pulse el botón de restaurar, se restauran los elementos de la lista.

Explicación 5

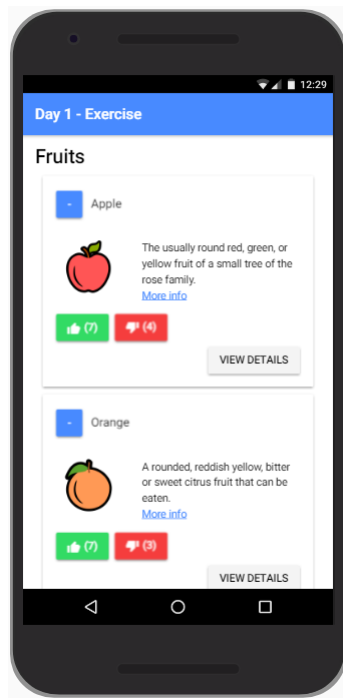
- Crear un componente *FruitCardComponent*
- Usar *FruitCardComponent* para representar sólo la primera fruta
- Gestionar la pulsación del botón eliminar
- Usar `*ngFor` para representar todas

Explicación 6

- Incorporar en explicación 5 la directiva *highlight*: se trata de una directiva de atributo que cuando se incluye en un tag, el elemento se resalta cuando se pasa por encima de él con el código de color que se especifica como parámetro de la misma.

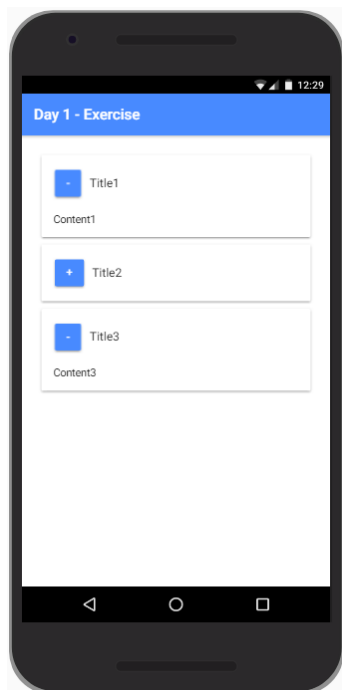
Ejercicio de combinación de componentes

Se pretende hacer una aplicación que liste frutas y recomendaciones hábitos saludables. Estos elementos se agruparían en cajas de contenidos expandibles. Además, cada uno de ellos dispondría de un control para indicar si te gusta o no similar al de los vídeos de YouTube.



Ejercicio 1

Implementar un componente de contenido expandible. Se compondrá de un botón para expandir y contraer el contenido, el título del contenido y el propio contenido. El resultado y código que ilustra su uso sería el siguiente:



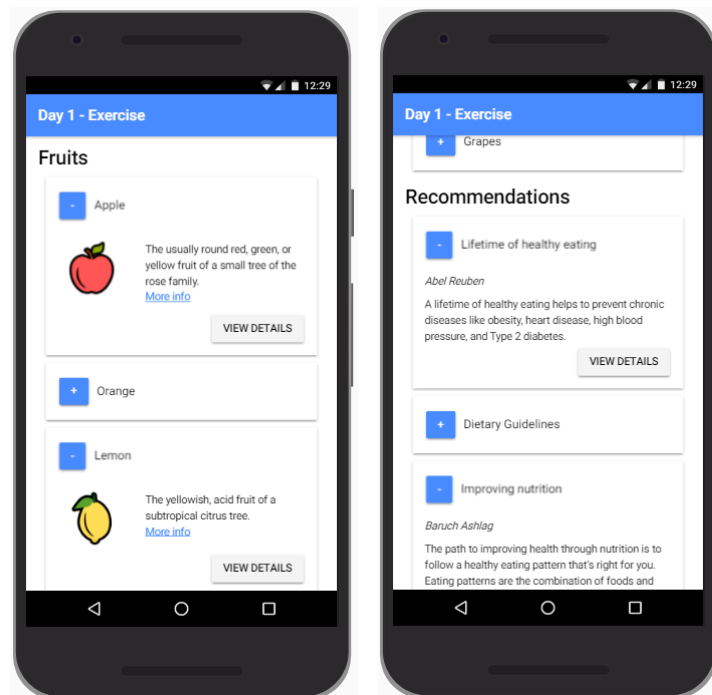
```
<expandable-content title="Title1">
  Content1
</expandable-content>

<expandable-content title="Title2">
  Content2
</expandable-content>

<expandable-content title="Title3">
  Content3
</expandable-content>
```

Ejercicio 2

Implementar un componente para representar el contenido de una fruta y otro componente para representar el contenido de una recomendación. Ambos componentes tendrán un botón de ver detalles, que cuando el usuario pulse emitiría un evento que puede ser capturado desde el componente desde donde se use (para probarlo, que muestre por consola la información del objeto del que se quieran conocer los detalles). Crear en el controlador principal arrays que almacenen los datos de las frutas y las recomendaciones. Representarlas en los contenidos expandibles como sigue:



Ejercicio 3

Implementar un componente para votar positivamente una fruta o una recomendación. Cada vez que se pulse uno de los botones, se incrementará el contador.

