

Día 3

Conceptos

- Repaso de la estructura de proyecto de Ionic, creación y uso de componentes (con parámetros de entrada y emisión de eventos).
- Vista detallada de los componentes existentes en Ionic.
- Typescript y SCSS.
- Navegación entre páginas.
- Menú, tuberías...

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

SASS

- Anidamiento
- Variables
- Mixins

Ejemplo de aplicación sobre la que hacer las próximas pruebas

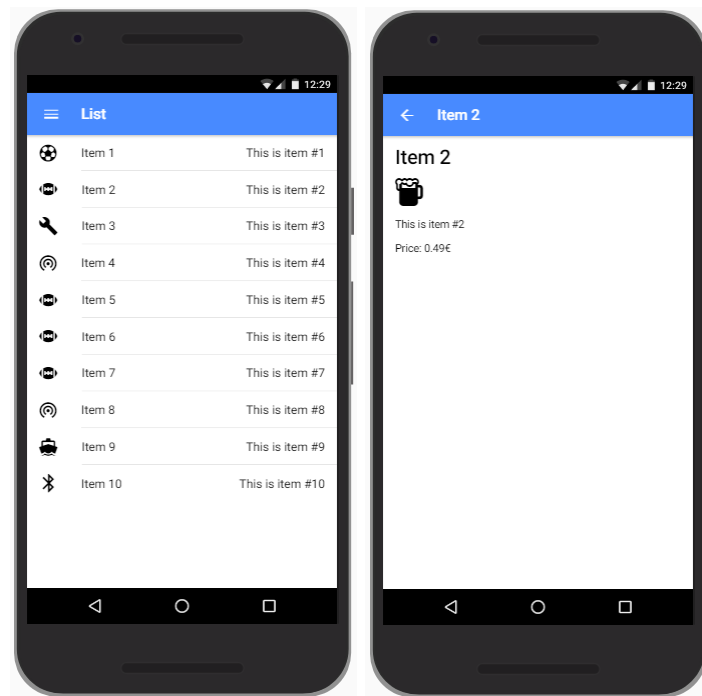
- Crear un proyecto de base con menú:

```
ionic start ExplanationApp sidemenu --no-git
```

- Ejecutar el proyecto:

```
ionic serve
```

Ejemplo de aplicación con menú, varias páginas y diálogo



- Estructura de la aplicación Ionic
- Establecer *ListPage* como la página de inicio
- Crear un servicio *ExampleItemsService* con una función que provea los datos que se muestran en *ListPage*
- Cargar los datos de los ítems que devuelve *ExampleItemsService* en *ngOnInit*. Ahora el atributo *items* es de tipo *ExampleItem*
- Crear una página de detalle para visualizar *ExampleItem* cuando se pulse sobre él en la pantalla de la lista. Tomar como referencia el HTML de la pantalla de la lista
- Refactorizar *ListPage*: ya no hace falta *selectedItem*, *navParams*, etc.
- Que el servicio una función que devuelva con 2 segundos de retraso los datos que se muestran en *ListPage*
- Explicar *Subject*, suscripciones y des-suscripciones
- Poner un *Spinner* para que dé feedback de carga mientras no se muestran los elementos de la lista
- Crear un botón *Navegar a Lista* en la pantalla de inicio, que muestre un diálogo para confirmar si el usuario desea navegar a la pantalla de la lista. Si lo confirma, que navegue a ella; si no, que no haga nada

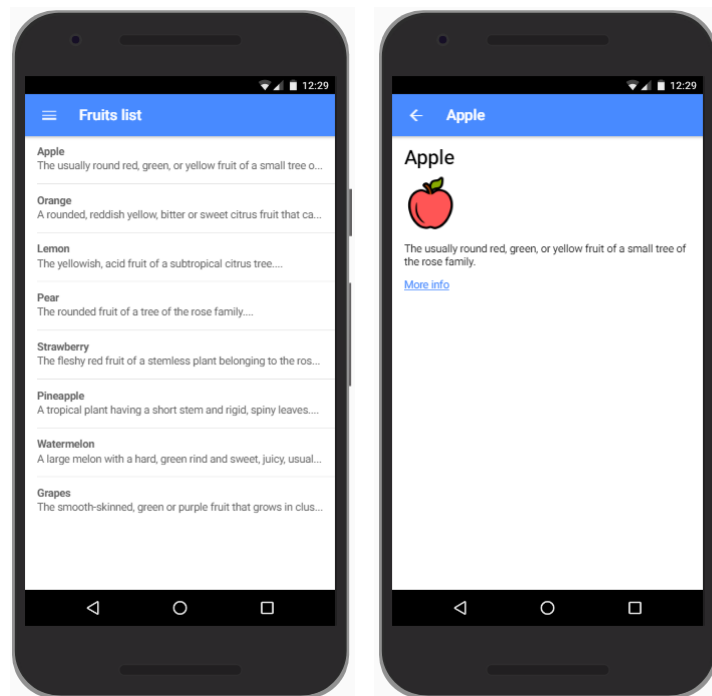
- Añadir un atributo *euroPrice* a *ExampleItem* que resultará un número aleatorio entre 0 y 5. Mostrarlo en la pantalla de detalle
- Crear una tubería *currencyFormat* para formatear el precio de manera que sólo muestre 2 decimales y concatene el símbolo del euro
- Parametrizar *currencyFormat* para que se le pueda especificar como parámetro la divisa que mostrará

Ejercicio de pantallas, alertas, componentes, tuberías, etc. (Parte I)

Se pretende hacer una aplicación con un contenido similar a la del ejercicio del día anterior. Esta vez, en vez de agrupar toda la información en una única pantalla, constará de varias pantallas y un menú que provee acceso a algunas de ellas. Constaría de una pantalla de inicio, una de lista de frutas, con su respectivo detalle de fruta y otra pantalla con una lista de recomendaciones, con su respectivo detalle de recomendación.

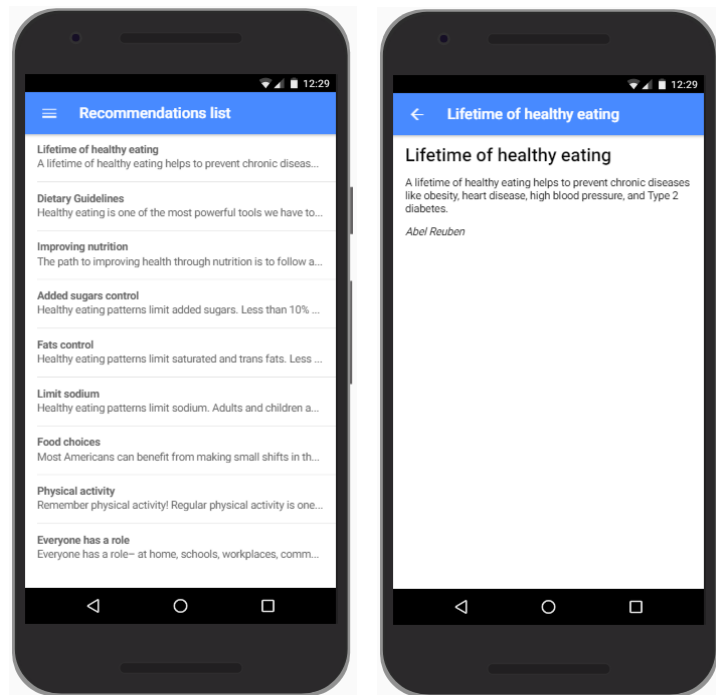
Ejercicio 1

Crear una pantalla de lista de frutas, que será accesible desde el menú principal. Las frutas serán provistas desde un servicio, que tardaría 2 segundos en servir los datos. Mientras los datos no estén disponibles en la pantalla, se mostrará un spinner. Cuando se pulse en uno de los elementos, se redirigirá al detalle. Para este caso, no haría falta crear componentes específicos para representar los datos del listado y del detalle: la propia página podría contener la lógica de representación de los elementos.



Ejercicio 2

Hacer lo mismo que en el Ejercicio 1 pero con las recomendaciones.



Ejercicio 3

Implementar adecuadamente que en la lista de frutas la descripción se corte a los 40 caracteres y en la lista de recomendaciones el contenido se corte a los 50 caracteres.

