

Día 4

Conceptos

- Componentes
- Módulos
- Servicios
- Enrutamiento y navegación entre páginas

Explicación de componentes

- Partir de la aplicación *explanation-app* del día anterior
- Mover la página de lista de frutas a una carpeta *pages*
- Crear un componente *fruit-card* que representa una tarjeta de fruta:

ionic generate component components/fruit-card
- Crear un módulo desde el cual se exportará el componente (si se declarasen más componentes, también se exportarían desde aquí):

ionic generate module components
- En dicho módulo se debe declarar *FruitCardComponent*; debe importar *CommonModule* e *IonicModule*; y debe exportar *FruitCardComponent*
- Este módulo de componentes se debe importar en el módulo de la página de la lista de frutas, porque en ella se usará *FruitCardComponent*
- Probar que el componente funciona correctamente (basta con que contenga un texto estático y se visualice en una página)
- El componente tendrá un parámetro de entrada *fruit* que representa la fruta
- También tendrá un evento de salida *deletePressed* que se emitirá cuando se pulse sobre el botón eliminar

Explicación de servicios, enrutamiento y navegación entre páginas

- Crear un proyecto base con:

ionic start explanation-app-2 sidemenu --no-git

- Dar un vistazo general al Proyecto
- Limpiar el contenido de la página Home
- Reubicar las páginas en una carpeta *pages*.
- Refactorizar la página *list*: eliminar la lógica de *itemSelected*, crear un modelo para *Item*, mover el código del *constructor* a *ngOnInit...*
- Crear un servicio *ItemsService* donde se alojará la lógica de items:

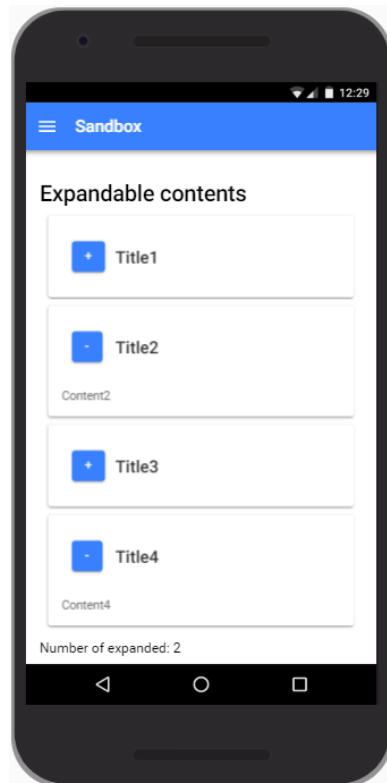
ionic generate service services/items

- Como los ítems se generan de manera aleatoria, lo ideal es disponer de un atributo privado del servicio donde se almacenen los items. En principio este atributo estaría vacío, pero si se solicitan los items, se cumplimentaría y sería éste el que se devuelve la próxima vez que se soliciten.
- Crear una nueva página *Details*:
ionic generate page pages/details
- Esta página se abrirá cuando se pulse sobre algún elemento de la lista
- Preparar en la cabecera un botón de atrás para que se pueda navegar a la página anterior
- Preparar la ruta de la pantalla de detalles para que se pueda pasar como parámetro el índice del ítem a mostrar
- Recuperar en el *ngOnInit* de la página de detalles el índice del ítem usando *ActivatedRoute*
- Recuperar el *Item* a partir de su índice y mostrar su detalle en la vista
- Con lo que conocemos ahora, resulta más fácil ver cómo funcionan los eventos del ciclo de vida de un componente: *ngOnInit* y *ngOnDestroy*
- Las páginas de Ionic usan *ion-router-outlet*. Ofrece dos métodos muy útiles del ciclo de vida de las páginas, *ionViewDidEnter* y *ionViewDidLeave*, que indican cuando se entra y se sale de ellas.

Ejercicio 1

Crear una nueva aplicación con la base de *sidemenu*. Crear una página Sandbox que contendrá un ejemplo de uso de los componentes *expandable-content* y *rating*.

1. Crear el componente *expandable-content* a partir de la maqueta elaborada el día anterior. La página Sandbox tendrá en su cabecera un botón para mostrar el menú principal. Tendría la siguiente apariencia:



La manera en la que se invoca el componente sería:

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

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

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

Pista: Pasar datos de entrada al componente como contenido de su tag se llama *transclusión*.

2. Dote al componente de la lógica necesaria para que indique si el contenido está expandido o no. Use esta información para incluir en la página Sandbox el número de contenidos expandibles que están expandidos.

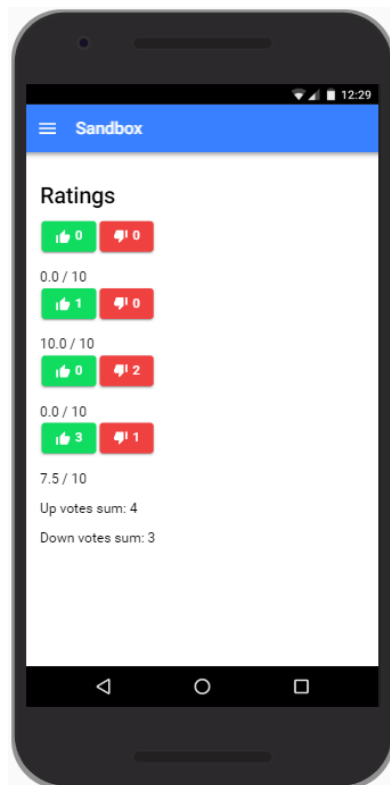
Pista:

```
<app-expandable-content title="Title1"
                        (change)="expandableContentChanged($event)">
  Content1
</expandable-content>
```

Siendo *\$event* una variable especial de las template que contiene lo que se le pasa como parámetro a la función *emit* del EventEmitter del componente.

Ejercicio 2

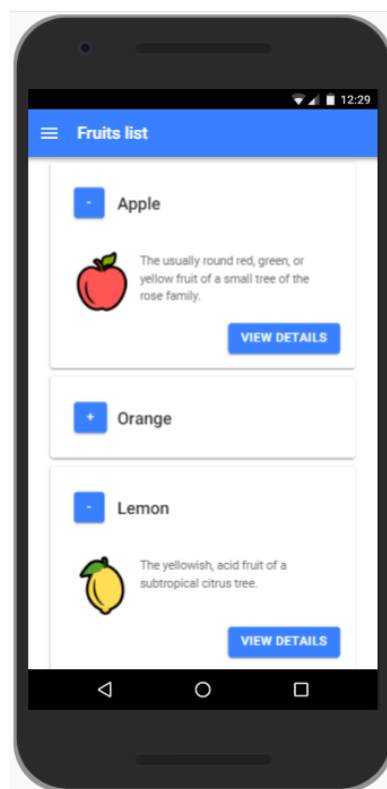
1. Crear el componente *rating* a partir de la maqueta elaborada el día anterior. Añada los componentes de prueba en la página Sandbox detrás de los *expandable-contents*.



2. Dote al componente para que se le puedan pasar como parámetros de entrada (de manera opcional) el número de votos positivos y el número de votos negativos. Si no se le pasa, se consideran cero. Inicialmente, al segundo componente se le pasa 1 voto positivo; al tercero 2 votos negativos; y al cuarto 3 votos positivos y un negativo.
3. Incluya en la parte inferior un contador que indique el número total de votos positivos y el número total de votos negativos de todos los componentes *rating*.

Ejercicio 3

1. Crear una página de lista de frutas. Las frutas deben ser obtenidas de un servicio. Se reutilizará el componente *expandable-content* para cada uno de los elementos de la lista. La página tendrá en su cabecera un botón para mostrar el menú principal y tendría la siguiente apariencia:



2. Crear una página de detalle de fruta, que se accederá al pulsar sobre el botón View Details de una determinada fruta en la página de la lista. La cabecera de esta pantalla tendrá un botón para poder volver a la página anterior. Tendría la siguiente apariencia:

