

## Día 4

### Conceptos

- Internacionalización de la aplicación
- json-server
- Servicios que hacen peticiones a un servidor

### Ejemplo de internacionalización

- Crear un nuevo proyecto Ionic.
- Añadir al proyecto las dependencias necesarias:

```
npm install --save @ngx-translate/core@9.1 @ngx-translate/http-loader
```

- Añadir a *app.module.ts*:

```
import { HttpClient, HttpClientModule } from '@angular/common/http';
import { TranslateModule, TranslateLoader } from '@ngx-translate/core';
import { TranslateHttpLoader } from '@ngx-translate/http-loader';

export function HttpLoaderFactory(httpClient: HttpClient) {
  return new TranslateHttpLoader(httpClient, './assets/i18n/', '.json');
}
```

- Añadir al nodo *imports*:

```
imports: [
  ...
  HttpClientModule,
  TranslateModule.forRoot({
    loader: {
      provide: TranslateLoader,
      useFactory: HttpLoaderFactory,
      deps: [HttpClient]
    }
  })
],
```

- En *app.component.ts* hay que cargar el idioma adecuado:

```
import { TranslateService } from '@ngx-translate/core'
...
constructor(
  ...
  private translate: TranslateService) {
}

ngOnInit() {
```

```
    this.translate.use('en');  
  }
```

- Crear un archivo *assets/en.json* que contiene información sobre las cadenas.
- Exponer las diferentes formas de acceder a los literales. Considerar que pueden estar parametrizados.

### **Ejemplo de integración con un servicio backend**

- Arrancar el json-server.
- Crear un servicio que sirva la lista de frutas que ofrece el json-server.
- Crear una pantalla que liste las frutas.
- Mientras carga la lista, que se muestre un spinner.
- Si ocurre un error al cargar, que se informe de éste y se ofrezca reintentar.

### **Ejercicio de pantallas, alertas, componentes, tuberías, etc. (Parte II)**

Continúe el ejercicio del día anterior añadiéndole nuevas funcionalidades.

#### **Ejercicio 4**

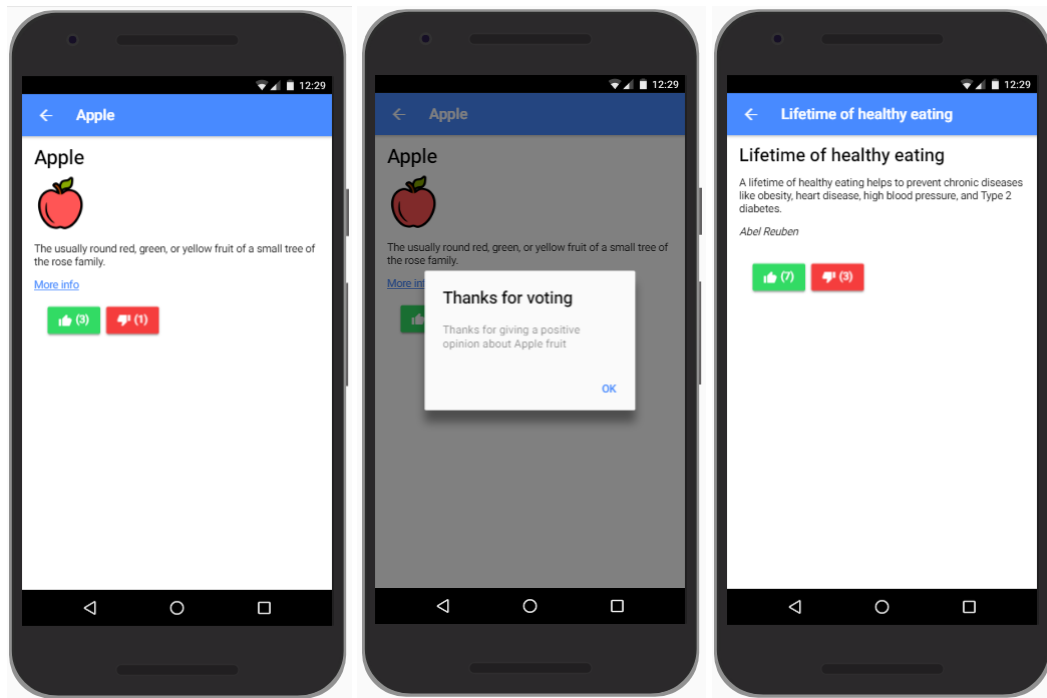
Parametrizar las cadenas de la aplicación para que puedan ser internacionalizadas.

#### **Ejercicio 5**

Preparar la aplicación anterior para que tome los datos del servidor de mocks. Dar feedback de carga y gestionar errores.

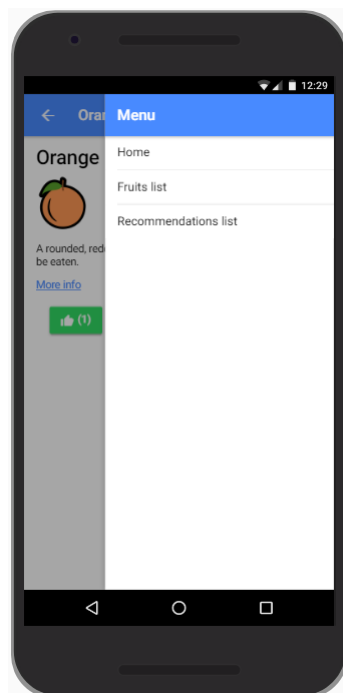
#### **Ejercicio 6**

Incorporar el control de votación en las pantallas de detalle de frutas y recomendaciones. Particularmente, para el detalle de fruta, cuando se realice una votación, se debe indicar un diálogo que informe sobre el nombre de la fruta sobre la que se votó y el tipo de voto (positivo o negativo). En el detalle de recomendación no tendría este comportamiento.



## Ejercicio 7

Hacer los cambios oportunos para que el menú principal salga desde la derecha y esté presente en todas las pantallas (incluso en las de los detalles). Resulta interesante que leas la documentación de la directiva *MenuToggle* y el controlador *MenuController*. No usar código CSS para conseguir el propósito: es posible hacerlo adecuadamente usando los componentes que provee Ionic según se indica en la documentación.



## Ejercicio 8

Llegado a este punto, habrás observado que el código relativo a la definición de las cabeceras de las pantallas es muy similar y se repite una y otra vez en todas ellas, y seguirá repitiéndose a medida que la aplicación crezca. Crea y usa un componente *CommonNavbarComponent* para dar solución a este problema.