# Día 5

## Conceptos

* Enrutamiento y navegación entre páginas

## Explicación

* Crear un proyecto base con:

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

(el proyecto resultante también está disponible en *explanation-app-base*)

* Dar un vistazo general al Proyecto: cómo funciona una aplicación con sidemenu, cómo se enrutan otras vistas, etc.
* Mover la carpeta de la página *folder* dentro de una carpeta *pages* (en el código sólo implicaría un cambio en el *app-routing.module.ts*).
* En *app.component.ts*, mover el código del *constructor* a *ngOnInit.*
* Crear un servicio *LabelsService* donde se residirá la lógica de gestión de la lista de etiquetas que aparece en el menú:

*ionic generate service services/labels*

* En vez de usar el decorador *Injectable* para que nos lo provea en el módulo raíz, lo proveeremos nosotros explícitamente en el módulo de la página que se requiera.
* Crear en el servicio *LabelsService* un método *getLabels* que devuelva la lista de labels que está en *app.component.ts.*
* En el *ngOnInit* de *app.component.ts,* obtener las labels a partir del método *getLabels* de *LabelsService*.
* Crear una nueva página *EmailsList* con:

*ionic generate page pages/emails-list*

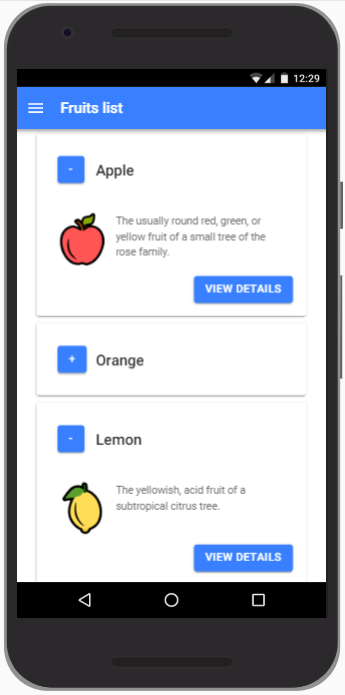
* Explicar cómo funciona el módulo de *routing* de una página y que podría evitarse incluyendo la lógica de enrutado dentro del propio módulo de la página.
* La idea es que la ruta de Inbox abra la página de *EmailsList* que estaría enrutada como *emails/:folder-id*. Hasta el momento es muy similar que la página folder. De hecho, su controlador, por el momento, es igual cambiando el parámetro.
* Prepararemos el HTML para mostrar una lista de con los títulos de diferentes emails.
* Crear un modelo/interfaz que representará un email. Tendrá un id, un title, un body y el folder al que pertenece el correo.
* Crear un servicio *EmailsService* que contendrá dos métodos: *getEmails(): Email[]* y *getEmail(id: number): Email*.
* Obtener la lista de emails a través del servicio.
* Crear una página *EmailDetails*, que se enrutaría como *emails/:folder-id/:email-id.*
* Preparar en la cabecera un botón de atrás para que se pueda navegar a la página anterior.
* Con lo que conocemos ahora, resulta más fácil ver cómo funcionan los eventos del ciclo de vida de un componente: *ngOnInit* y *ngOnDestroy.* Implementadlos en la página *EmailDetails* incluyendo un *console.log*.
* Las páginas de Ionic usan *ion-router-outlet*. Este 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. Implementadlos en la página *EmailDetails* incluyendo un *console.log* y observad el orden en que se ejecutan todos.

## Ejercicio 1

Crear una nueva aplicación con menú lateral que daría acceso a las páginas: *Sandbox* y *FruitsList*. La página *Sandbox* contendrá la solución dada al ejercicio del día anterior, por lo que habría que mover de forma controlada toda la funcionalidad desarrollada a esta nueva aplicación (página, componentes, módulos, etc.); la página *FruitsList* daría solución a uno de los puntos del ejercicio que se enuncia a continuación.

## Ejercicio 2

1. Desarrolle la página de la 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:



1. 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:

