

Día 7

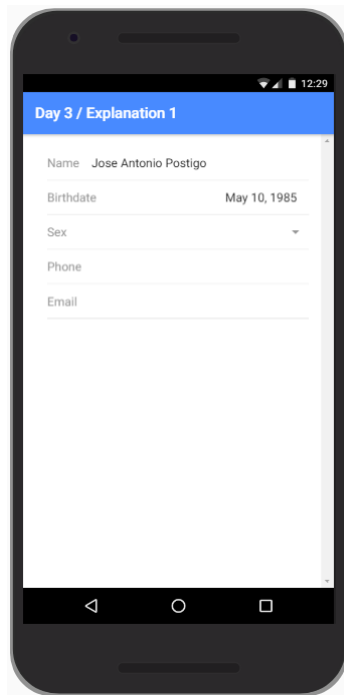
Conceptos

- Formularios *template-driven* y reactivos

Ejemplo de formularios

Explicación 1

Usar los componentes de Ionic para crear un formulario (*template-driven*) con un campo de texto, uno de fecha, un select, uno numérico y otro email.



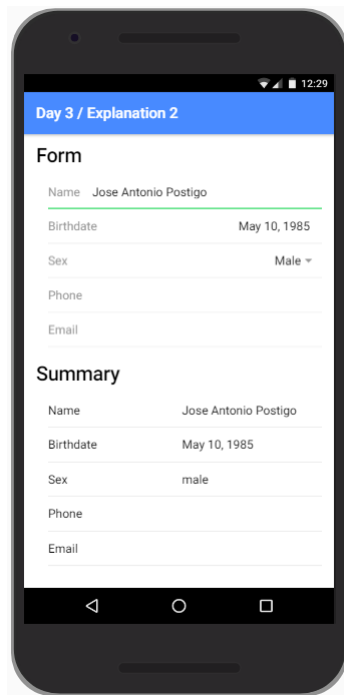
A smartphone mockup displaying a form titled "Day 3 / Explanation 1". The form contains the following fields:

- Name: Jose Antonio Postigo
- Birthdate: May 10, 1985
- Sex: (dropdown menu)
- Phone: (text input)
- Email: (text input)

The form is displayed on a black smartphone with a white background. The status bar at the top shows the time 12:29 and signal strength. The bottom navigation bar shows the back, home, and recent apps buttons.

Explicación 2

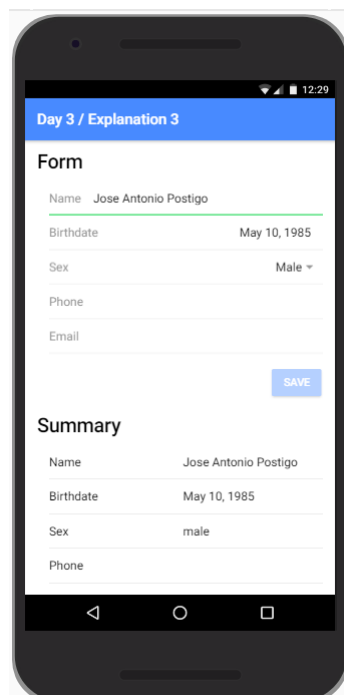
Crear bajo el formulario un apartado de resumen que indique el valor de los campos.



The screenshot shows a mobile application interface with a blue header bar labeled "Day 3 / Explanation 2". Below the header, there is a section titled "Form" containing five input fields: "Name" (filled with "Jose Antonio Postigo"), "Birthdate" (filled with "May 10, 1985"), "Sex" (a dropdown menu showing "Male"), "Phone", and "Email". Below the form section is a section titled "Summary" which displays the values from the form: "Name" (Jose Antonio Postigo), "Birthdate" (May 10, 1985), "Sex" (male), "Phone", and "Email". The bottom of the screen shows the standard Android navigation bar.

Explicación 3

Adaptar el formulario de manera que sea reactivo (no olvide importar el módulo *ReactiveFormsModule*), de forma que sea más fácil de gestionar en el controlador. Añadir un botón de envío del formulario que sólo estará activado cuando el formulario sea válido, y que imprima el objeto del formulario cuando éste sea pulsado.



The screenshot shows a mobile application interface with a blue header bar labeled "Day 3 / Explanation 3". Below the header, there is a section titled "Form" containing five input fields: "Name" (filled with "Jose Antonio Postigo"), "Birthdate" (filled with "May 10, 1985"), "Sex" (a dropdown menu showing "Male"), "Phone", and "Email". To the right of the "Email" field, there is a blue button labeled "SAVE". Below the form section is a section titled "Summary" which displays the values from the form: "Name" (Jose Antonio Postigo), "Birthdate" (May 10, 1985), "Sex" (male), "Phone", and "Email". The bottom of the screen shows the standard Android navigation bar.

Explicación 4

Preparar el formulario para que por defecto tenga todos sus campos vacíos y 2 segundos después se rellene con los datos de un usuario.

Explicación 5

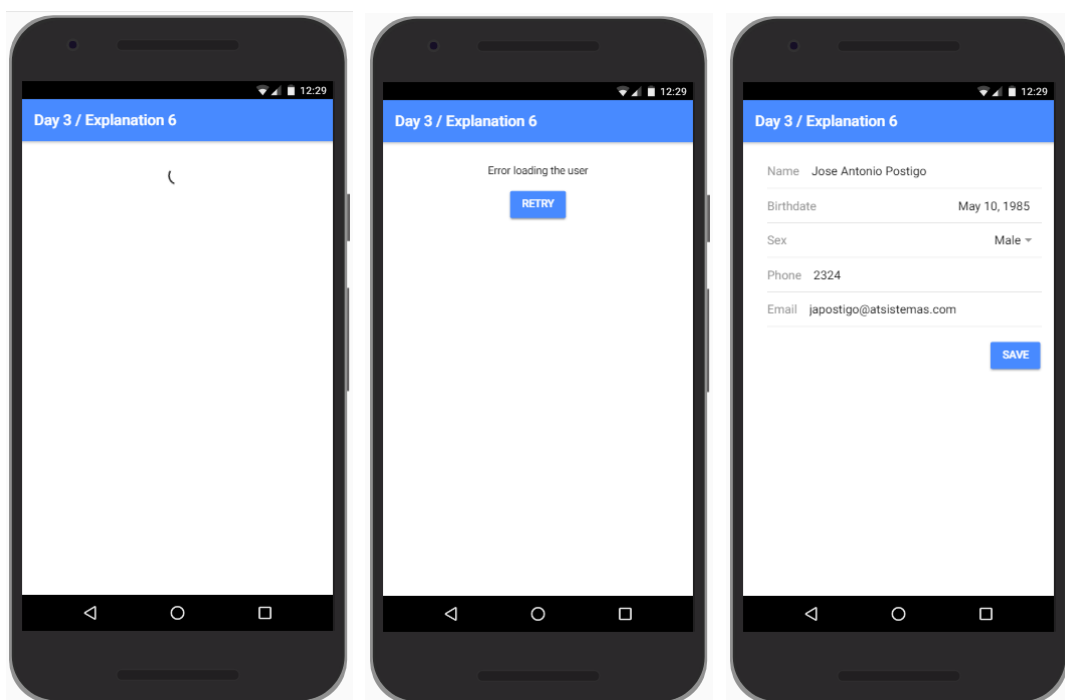
Crear un servicio que provea los datos de un usuario una vez que transcurren 2 segundos. Mostrar en formulario una vez que los datos estén disponibles, mientras mostrar el spinner de carga. Eliminar la sección de resumen de datos. Definir el tipo de dato Usuario en el mismo fichero donde se implemente el servicio.

Posteriormente, hacer la mismo, pero obteniendo los datos del mockServer. Implementar también el guardado de datos.

Explicación 6

Hacer una gestión correcta del feedback al usuario y los errores:

- Mostrar el spinner de cargando mientras cargan los datos del usuario
- Si hubiera un error al cargar (que se puede provocar deteniendo el mockServer), se debe mostrar un mensaje de error y un botón para reintentar.
- Mientras se está guardando el usuario, mostrar un spinner en lugar del botón de guardar.
- Informar sobre el guardado, ya sea exitoso o no usando un toast (ver *ToastController*).

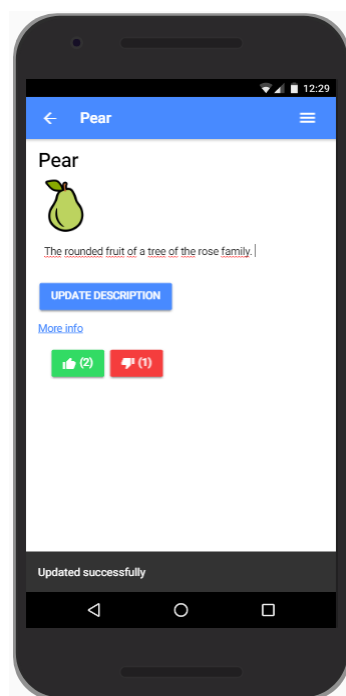


Ejercicio de integración con back

Se pretende complementar la aplicación del día anterior para que se integre con un servicio de backend (el mockServer). Se listarán los elementos que vengan del servidor y se dará la posibilidad de actualizar el detalle de ellos.

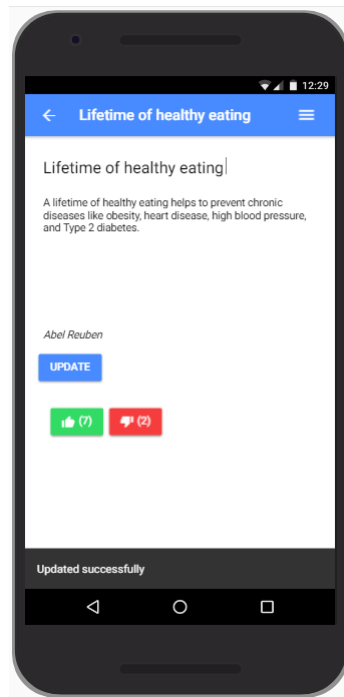
Ejercicio 1

Listar las frutas que vengan del servidor. Ofrecer la posibilidad de actualizar la descripción de la fruta en la pantalla de detalle. Mientras se actualiza, en vez del botón de actualizar, deberá mostrarse el spinner de carga. Dar al usuario feedback de la operación usando un toast.



Ejercicio 2

Hacer lo mismo que lo anterior con la funcionalidad de recomendaciones. En el detalle de una recomendación se podrían editar el título, contenido y autor. El título de la cabecera de la pantalla no debería actualizarse hasta que el servidor confirme la correcta realización de la operación.



Ejercicio 3

Habrás observado que el código relativo a la funcionalidad de mostrar los toasts se repite en los controladores del detalle de fruta y el detalle de recomendación. Saque ese código común a un servicio y hacer uso de él para evitar la repetición de código.