

ISSI-2 IS: Simulacro de examen de laboratorio

Enunciado

Una vez se ha puesto en marcha la primera versión de DeliverUS, los inversores han solicitado la inclusión de una nueva funcionalidad que consiste en ofrecer a los propietarios la posibilidad de promocionar sus restaurantes. **Cada propietario sólo podrá promocionar uno de sus restaurantes.**

(Validations)

Un propietario podrá promocionar un restaurante de **dos maneras** distintas:

- ① • En el formulario de **creación de restaurante**. Por defecto, se seleccionará la opción de **no** **promocionado**. Si el propietario indica que el **nuevo restaurante** debe estar **promocionado**, pero **ya existían restaurantes promocionados** del mismo propietario, al pulsar el botón `Save` se mostrará un **error** y no se creará el restaurante.
 (Switch) →
- ② • En la pantalla de "Mis restaurantes", mediante un **botón** mostrado junto a cada restaurante, que permitirá mediante su pulsación **promocionar** el restaurante en cuestión. Si el propietario pulsa el botón para promocionar un nuevo restaurante y **ya existían otros restaurantes promocionados** del mismo dueño, se procederá a **promocionar el restaurante indicado** y se marcará como **"no promocionado"** el restaurante que lo fuese **anteriormente**. La aplicación debe pedir **confirmación** al propietario cuando se pulse el botón; utilice para ello el componente suministrado `ConfirmationModal`, similar al componente `DeleteModal` utilizado en clase.
 (Requirements) (model Restaurant)

Además, los restaurantes **promocionados** aparecerán **siempre** al **principio de los listados** de restaurantes que se le presentan tanto a los propietarios como a los clientes. Además de presentarse al principio, los restaurantes promocionados deben destacarse visualmente, por lo que aparecerá una etiqueta de texto **¡En promoción!** con el color principal de la marca.

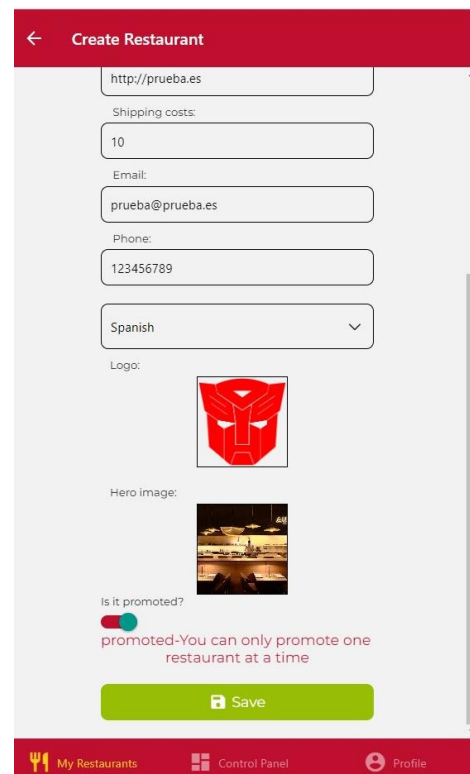
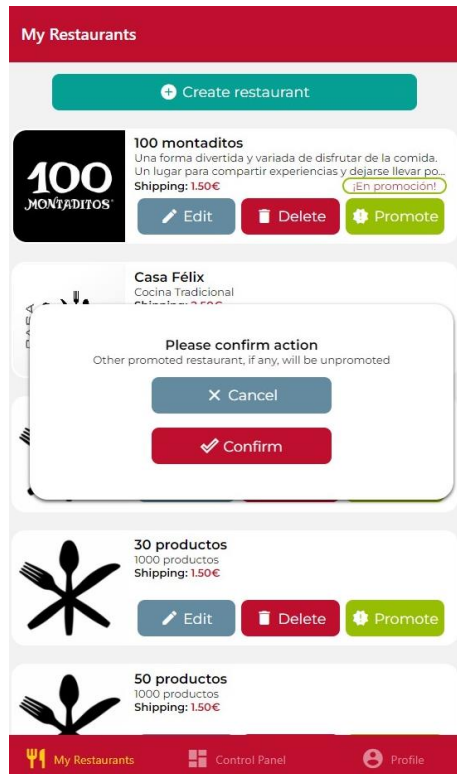
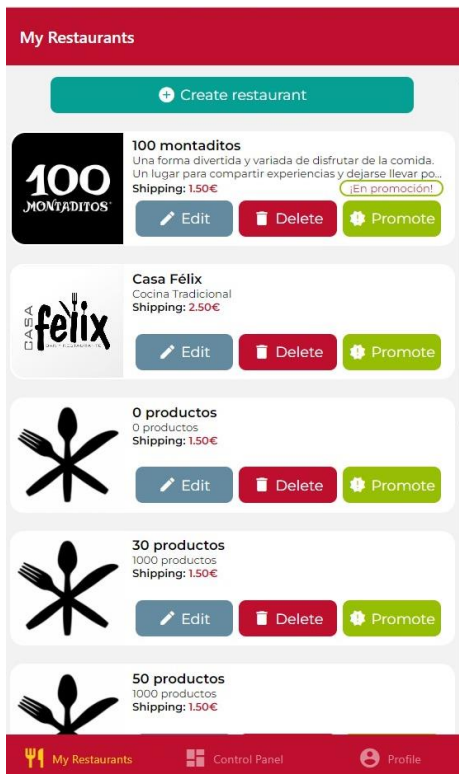
Ejercicio 1

Realice todos los cambios necesarios en el proyecto de backend para implementar el nuevo requisito.

Ejercicio 2

Realice todos los cambios necesarios en el proyecto de frontend para implementar el nuevo requisito.

Pantallas



BACKEND

- migrations y modelo Restaurant → Añadir atributo promoted, con valor por defecto FALSE
- Validations Restaurant → Crear método que compruebe si hay un solo restaurante promocionado y lo incluimos en los check de create y update.
- Restaurant Controller → Añadir el ordenamiento pedido tanto a index como a indexOwner (Es DESC pg vale T a F)
- Añadir un método para actualizar la promoción → promote
Lo que hace este método es comprobar si hay algún restaurante promocionado, si lo hay, le quita la promoción y se lo pone al restaurante que queremos promocionar.
- Restaurant Routes → Añadir una ruta específica para la promoción → `restaurants/:restaurantId/promote'.patch()`

FRONTEND

- RestaurantEndpoints → Añadir el endpoint de la nueva ruta → promote
- Configuración Modal → Tenemos que crear una ventana de confirmación para el momento de promocionar con el botón.

1 Create Restaurant Screen

- ↳ 1 En los valores iniciales añadimos {promoted: false}, con valor por defecto
- ↳ 2 En el return final → Añadir el SWITCH por seleccionar la promoción del restaurante.
- ↳ 3 Al final añadimos el estilo que va a tener es switch.

2 Restaurant Screen

- ↳ 1 Añadir constante restaurantToBePromoted → useState(null)
- ↳ 2 Añadir el texto 'En promoción' en render Restaurant, solo si está en promoción
- ↳ 3 Añadir botón promote en render Product
- ↳ 4 Añadir un método para promocionar un restaurante que utilice el Endpoint creado (que está conectado con el backend a través de la ruta)
- ↳ 5 Añadir la ventana de promoción en el return final
- ↳ 6 Creamos el estilo del texto 'En promoción' (badge)

Introducción

Este repositorio incluye el backend completo (carpeta `DeliverUS-Backend`) y el frontend de `owner` (carpeta `DeliverUS-Frontend-Owner`). Servirá como base para realizar el examen de laboratorio de la asignatura.

Preparación del entorno

a) Windows

- Abra un terminal y ejecute el comando `npm run install:all:win`.

b) Linux/MacOS

- Abra un terminal y ejecute el comando `npm run install:all:bash`.

Ejecución

Backend

- Para **rehacer las migraciones y seeders**, abra un terminal y ejecute el comando

```
npm run migrate:backend
```

- Para **ejecutarlo**, abra un terminal y ejecute el comando

```
npm run start:backend
```

Frontend

- Para **ejecutar la aplicación frontend de owner**, abra un nuevo terminal y ejecute el comando

```
npm run start:frontend:owner
```

Depuración

- Para **depurar el backend**, asegúrese de que **NO** existe una instancia en ejecución, pulse en el botón `Run and Debug` de la barra lateral, seleccione `Debug Backend` en la lista desplegable, y pulse el botón de *Play*.
- Para **depurar el frontend**, asegúrese de que **EXISTE** una instancia en ejecución del frontend que desee depurar, pulse en el botón `Run and Debug` de la barra lateral, seleccione `Debug Frontend` en la lista desplegable, y pulse el botón de *Play*.

Test

- Para comprobar el correcto funcionamiento de backend puede ejecutar el conjunto de tests incluido a tal efecto. Para ello ejecute el siguiente comando:

```
npm run test:backend
```

Advertencia: Los tests no pueden ser modificados.