

# IISSI-2 IS: Examen de laboratorio Junio 2024.

---

## Restaurantes Fijados. 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 fijar sus restaurantes. Cada propietario podrá fijar tantos restaurantes como desee.

Un propietario podrá fijar restaurantes de dos maneras distintas:

- En el formulario de creación de restaurante. Por defecto no estará fijado, pero podrá fijarlo. Para ello se deberá proporcionar un `Switch` que trabaje con una propiedad que debe llamarse `pinned`. Si se marca el `Switch`, el restaurante deberá crearse como fijado`
- En la pantalla de "Mis restaurantes", mediante un icono que actuará como botón y será mostrado junto a cada restaurante. Mediante su pulsación fijará o desfijará el restaurante en cuestión. 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. El sistema informará al usuario si el restaurante ha quedado fijado o si el restaurante ha quedado desfijado.

Finalmente, los restaurantes fijados aparecerán siempre al principio de los listados de restaurantes que se le presentan a su propietario y ordenados por fecha en la que fue fijado (más antiguos primero) y después los no fijados.

## Ejercicio 1

Realice todos los cambios necesarios en el proyecto de backend para implementar el nuevo requisito. Los test de backend esperan que la ruta sea: `/restaurants/:restaurantId/togglePin` y que los restaurantes tengan una nueva propiedad llamada `pinnedAt`.

Recuerde que puede correr los tests con:

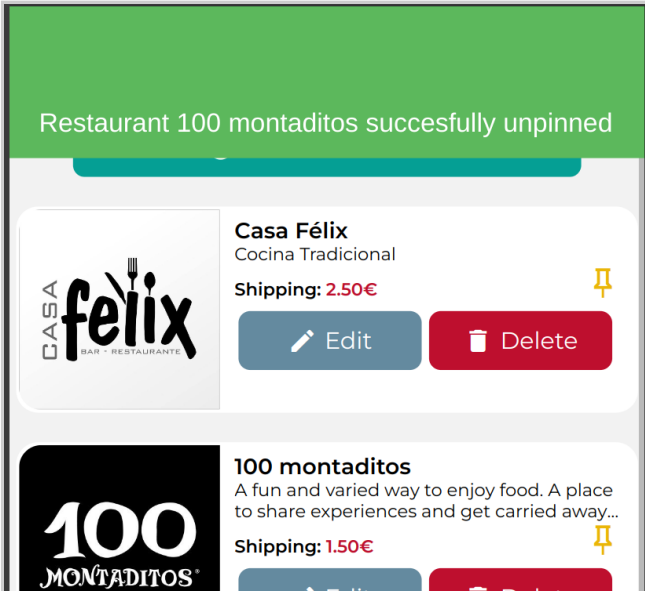
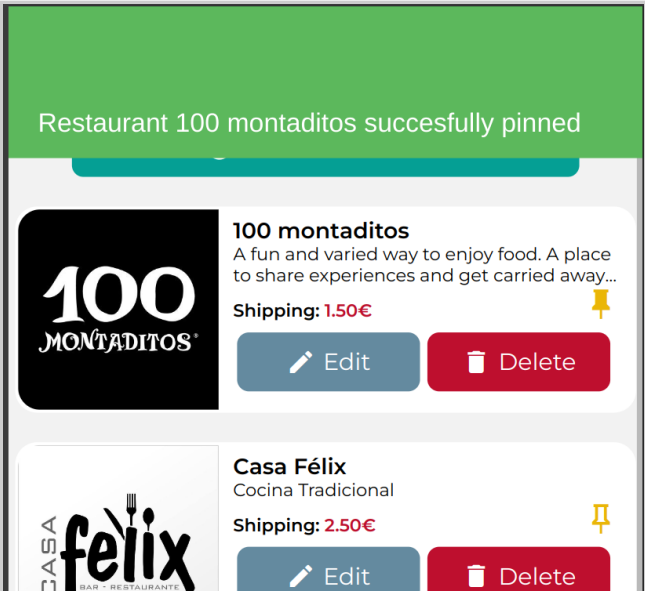
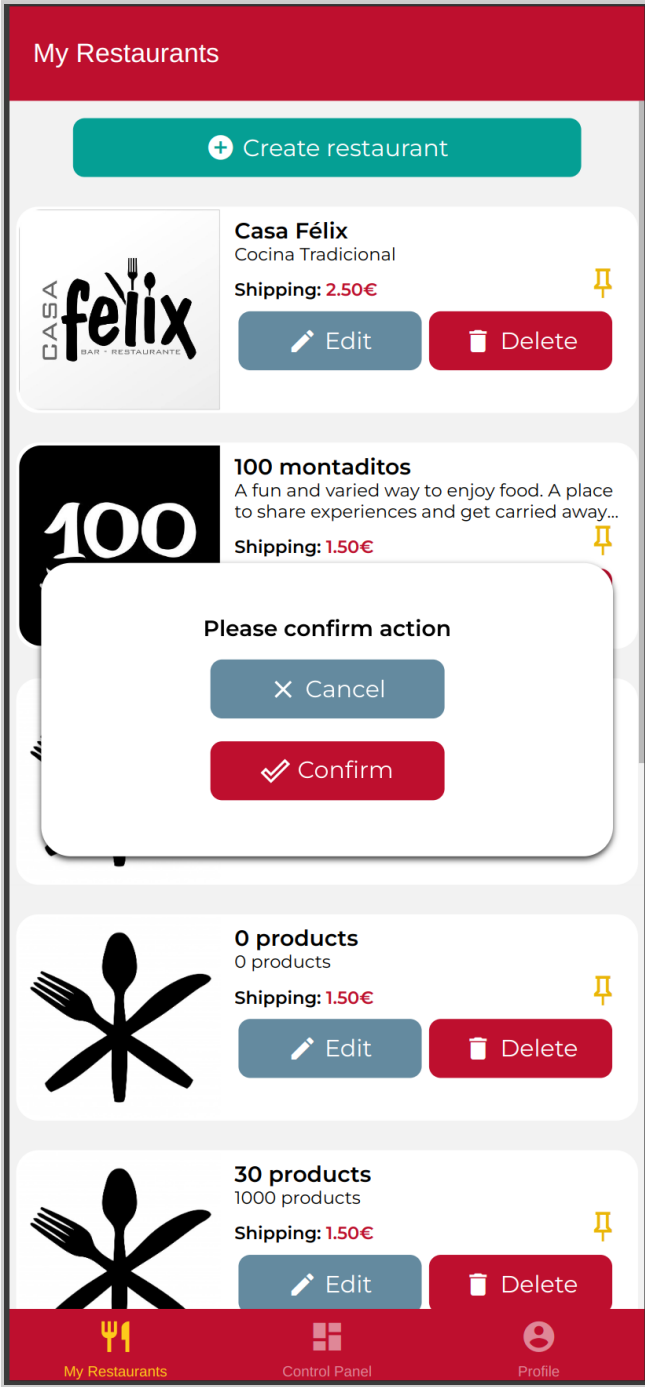
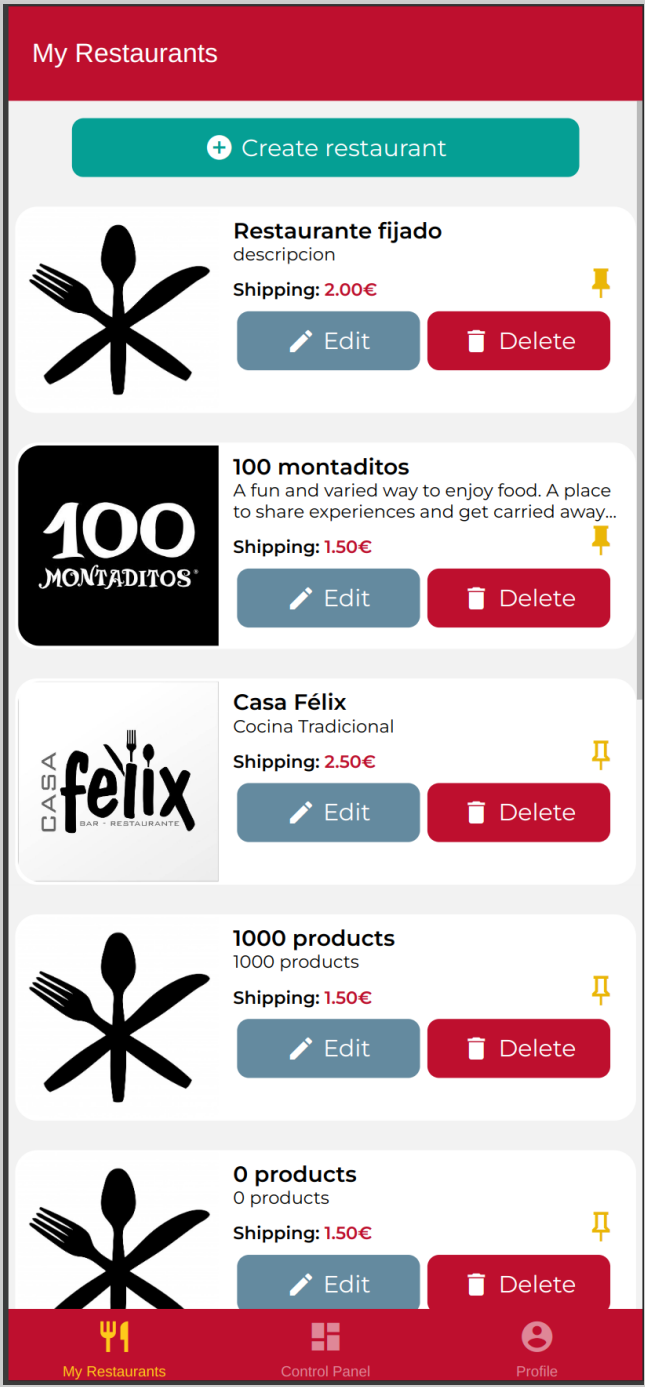
```
npm run test:backend
```

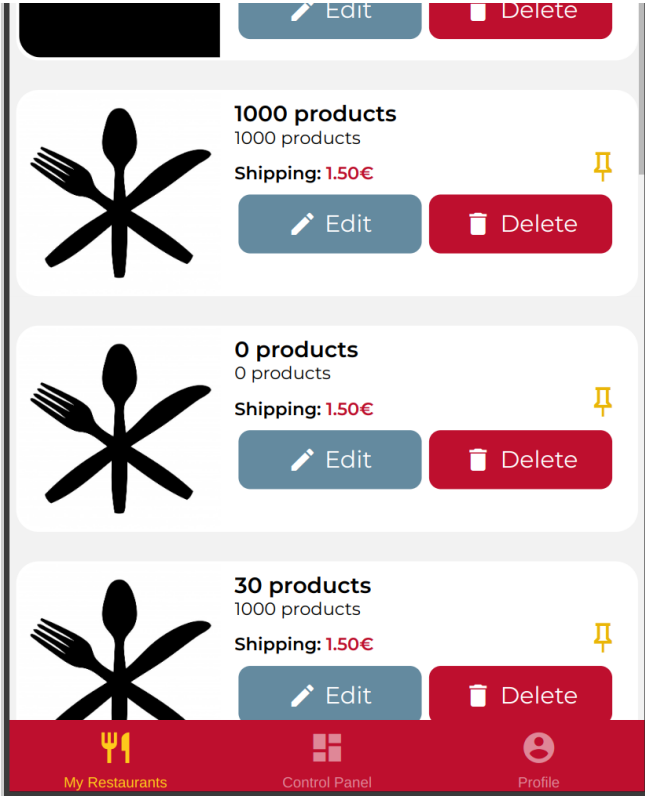
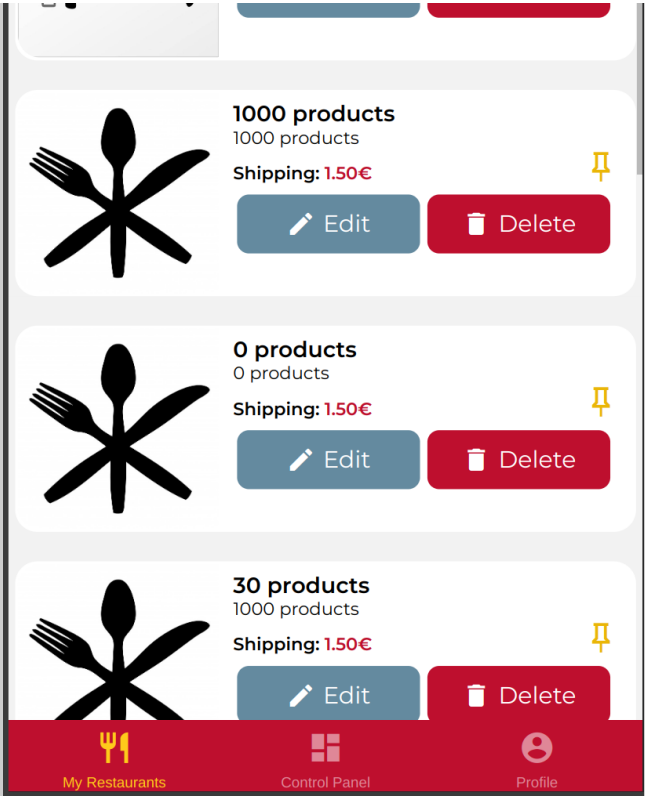
## Ejercicio 2

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

Puede renderizar el icono de fijado propuesto con

```
<MaterialCommunityIcons
  name={item.pinnedAt ? 'pin' : 'pin-outline'}
  color={GlobalStyles.brandSecondaryTap}
  size={24}
/>
```





Create Restaurant

Postal code:

41012

Url:

http://nuevo.com

Shipping costs:

2

Email:

nuevo@nuevo.com

Phone:

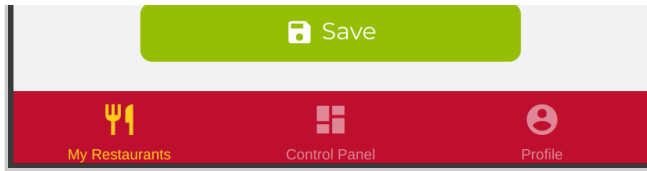
123456789

Spanish

Logo:

Hero image:

Pin restaurant?



## 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.**