

ISSI2: Evaluación individual laboratorio Octubre 2023

Preparación del entorno

Windows:

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

Linux/MacOS:

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

Ejecución y depuración

- Para **ejecutar el backend**, abra un terminal y ejecute el comando `npm run start:backend`.
- Para **ejecutar el frontend**, abra un nuevo terminal y ejecute el comando `npm run start:frontend`.
- 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, pulse en el botón Run and Debug de la barra lateral, seleccione Debug Frontend en la lista desplegable, y pulse el botón de Play.
- Para **rehacer las migraciones y seeders del backend**, abra un terminal y ejecute el comando `npm run migrate:backend`.

Enunciado

La empresa ha decidido ofrecer a los propietarios la posibilidad de **cambiar** manualmente el **estado** de sus **restaurantes abiertos** (online u offline). En caso de que un restaurante tenga un estado **online u offline**, la nueva funcionalidad proporcionaría un **botón** para **alternar entre ambos estados**. De lo contrario, dicho botón no debe estar **disponible**.

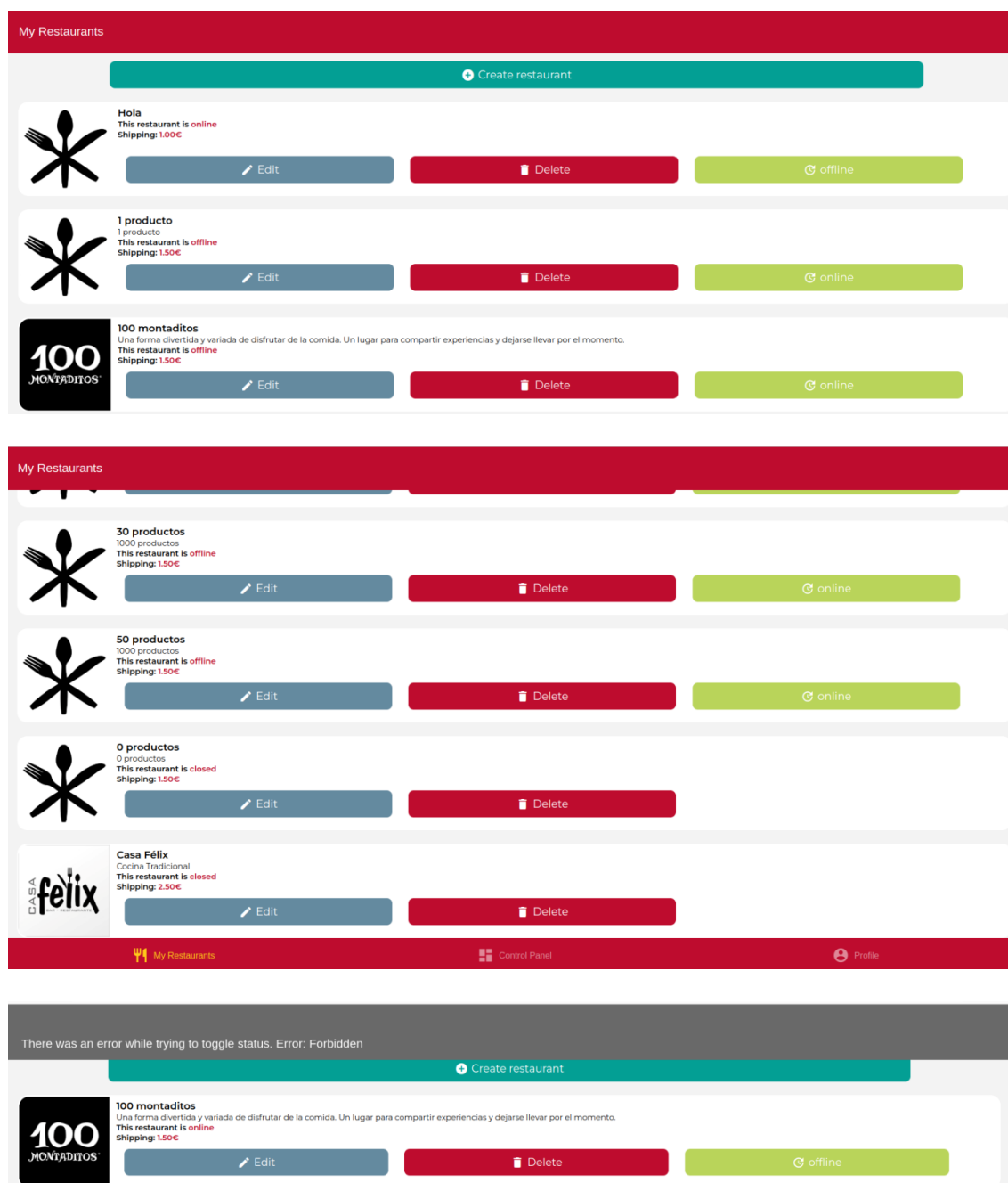
Tenga en cuenta que cualquier **restaurante nuevo** se almacena **inicialmente** como **offline** de forma **predeterminada**. **Solo los restaurantes offline pueden estar en línea y viceversa**. Por otro lado, si un restaurante alcanza un estado de **cerrado o cerrado temporalmente**, la posibilidad de **cambio manual** **no** estará **disponible** y su uso estará prohibido por el sistema.

Además, tenga en cuenta que un restaurante **tampoco podrá cambiar su estado** si tiene **pedidos** con un valor **nulo** en **deliveredAt**.

La nueva funcionalidad también implicará proporcionar la **lista de restaurantes propiedad** del usuario **ordenados** por **estado** (**ascendentemente**) y, para el **mismo estado**, por **nombre**.

Finalmente, **el estado de cada restaurante debe ser visible**.

El sistema debe mostrar estos requisitos como se muestran en las siguientes capturas de pantalla:



Le pedimos que implemente los cambios necesarios en el backend y frontend para incluir la funcionalidad requerida.

Procedimiento para la entrega

- 1. Elimine las carpetas `node_modules` de los proyectos de backend y frontend y la carpeta `.expo` del proyecto frontend.
- 2. Cree un zip que incluya ambos proyectos. **OJO: Compruebe que el zip resultante no es el mismo que descargó y que por lo tanto incluye sus soluciones**
- 3. Avise a su profesor antes de realizar el envío.
- 4. Cuando tenga el visto bueno de su profesor, envíe el fichero resultante a través de enseñanza virtual.

BACKEND

① CONTROLLERS

↳ RestaurantController → Añadimos el método status que cambie entre 'online' y 'offline' la propiedad 'status' ya existente. Lo hacemos siguiendo las restricciones indicadas.

Modificamos el método indexOwner para cumplir con la forma de ordenamiento pedida.

② ROUTES

↳ RestaurantRoutes → Añadimos una nueva ruta para el método creado.

FRONTEND

① ENDPOINTS

↳ RestaurantEndpoints → Añadimos el endpoint de la ruta creada.

② SCREENS

↳ RestaurantScreen → Añadimos el botón para cambiar el 'status' junto con la legenda para hacer el cambio y siguiendo las restricciones correspondientes.

Añadimos el texto que nos indica el estado actual del restaurante.