# Prueba Angular

# Parte 1 Cuestionario (2 puntos)

- 1. ¿Qué son los hooks del ciclo de vida en Angular? Explique algunos hooks del ciclo de vida
- **2.** ¿Qué es la compilación AOT *(Ahead-Of-Time)*? ¿Cuáles son sus ventajas? ¿Conoces otra?
- 3. ¿En qué se diferencian los Observables de las Promesas?
- 4. ¿Qué es ViewEncapsulation y cuántas formas hay de hacerlo en Angular?

# Parte 2 Ejercicio (8 puntos)

Hay que crear una aplicación Front usando el FWKCNA Angular.

La aplicación consiste en un portal para realizar ciertas gestiones. Para la V1 quieren poder gestionar tornillos *(se refiere a productos)* que aún no se venden en Mercadona, tornillos candidatos.

Para ser ágiles nada más entrar en la aplicación, sin necesidad de iniciar sesión, verán una card que les informará si existen tornillos candidatos por revisar.

Si acceden a la sección, tras iniciar sesión, verán la lista de tornillos candidatos, donde verán el nombre, precio de venta del proveedor, el formato de venta y el nombre del proveedor.

Se deberá poder ordenar las columnas de la tabla como quieran.

Se podrá paginar el servicio y pedir más tornillos por páginas (5, 10 o 20).

Podrán eliminar tornillos, así como dar de alta.

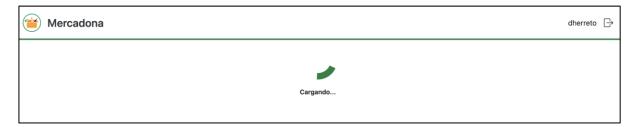
Para el alta de tornillos todos los campos serán obligatorios.

El equipo de UX/UI nos proporciona el diseño de la aplicación.

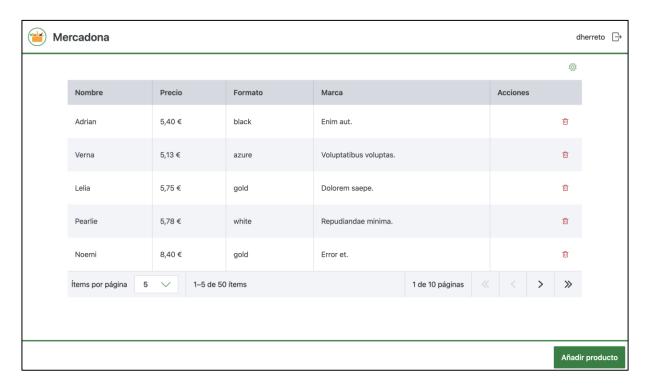
## Pantalla Home.



Pantalla cargando tornillos.

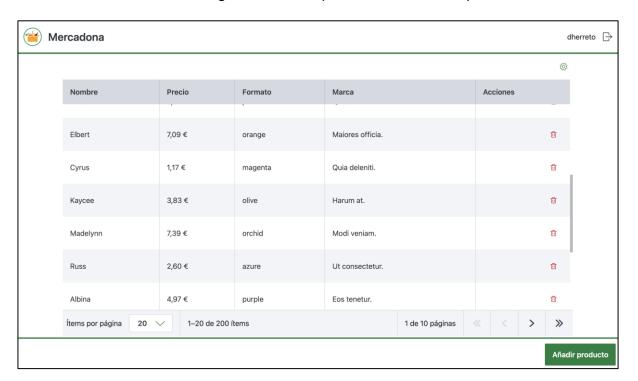


### Pantalla tornillos.



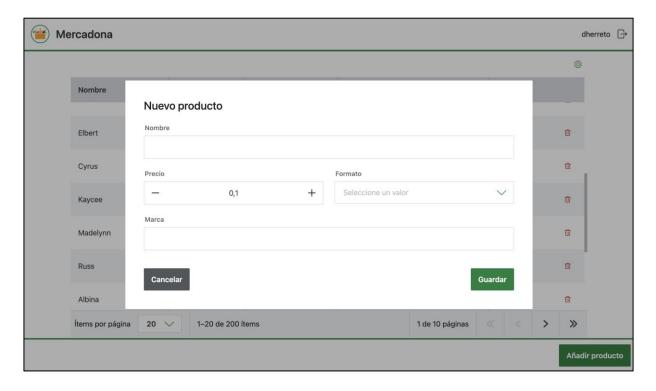
Nota de UX: el footer debe de ser fixed.

Pantalla tornillos con más registros de los que caben en el viewport.



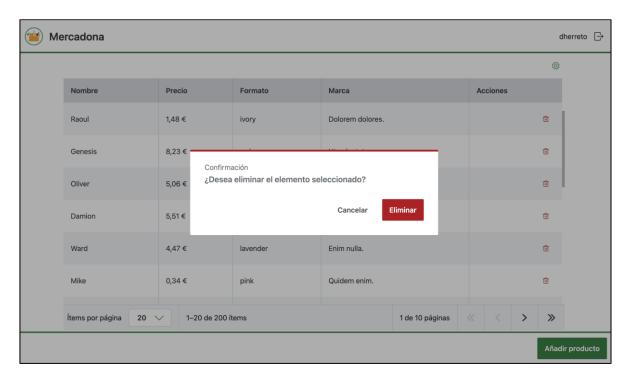
Nota UX: la cabecera de la tabla y el paginador se quedan fijos, solo se hace scroll por los datos.

Modal de alta de tornillo.

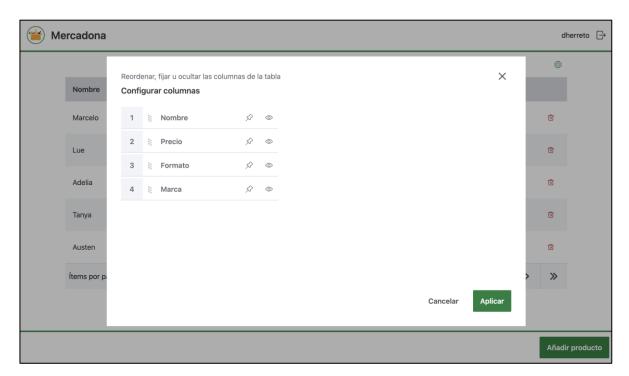


Requisito de negocio: quieren que el precio incremente 0,1 con los botones.

Modal de confirmación para borrar un tornillo.



Modal de ordenación de columnas.



Nota de UX: la columna de eliminar no se gestiona.

Nota: aquellas personas de fuera de Mercadona que opten a este puesto deben hacer igualmente todos los ejercicios, pero sin hacer uso del Framework de desarrollo de Mercadona. En este caso no es necesario usar la guía de estilos de Mercadona, deberá desarrollar algo parecido.

### Criterios de evaluación:

- Terminar la aplicación (2 puntos)
- Uso de componentes FWK:
  - o FWK CNA Angular (2 puntos) (internos)
  - Otro FWK 'open source' (2 puntos) (externos)
- Uso de Git, 'conventional commits' (2 puntos)
- Test unitarios (2 punto)
- Uso de tipos e interfaces (1 punto)
- Buenas prácticas de estilo SCSS (BEM, OOCSS, ...) (1 punto)