



MEMORIA PROYECTO

'Trabajo individual'

Fecha Entrega: 25/05/23

Integrantes:

JIGO

Prohibida la reproducción total o parcial sin autorización escrita de la EOI.

^{©:} Quedan reservados todos los derechos. (Ley de Propiedad Intelectual del 17 de noviembre de 1987 y Reales Decretos).

Documentación elaborada por la EOI.

Probibila la propieda del la EOI.











1. - Memoria Descriptiva

1.1 Organización

El proyecto ha sido desarrollado trabajando primero la parte de maquetación y posteriormente la lógica. Teniendo así una visión global de las acciones que tendrá el usuario final a la hora de interactuar con la página. El equipo de desarrollo es individual realizando todas las tareas en el orden descrito.

A continuación, se muestra la distribución de las tareas:

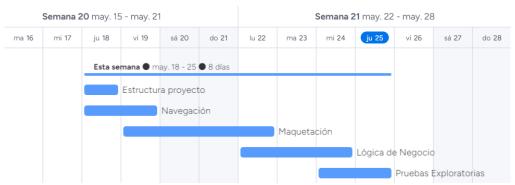


Ilustración 1. Diagrama de Gantt del proyecto

1.2 Investigación

Se ha profundizado en como se producen las llamadas asíncronas a la API y el uso de observables, con el objetivo de evitar errores a la hora de mostrar información que aún no se ha obtenido en la llamada.

En concreto se ha investigado el uso de los observables y de Subject para obtener datos globales en la aplicación y que todos los componentes puedan disponer siempre actualizados. Si queremos que un componente quede actualizado cuando un servicio actualice algún dato de su interior (sin importar quien lo llame) podemos utilizar los observables. Para ello declararemos el servicio como global con povidedIn: 'root' dentro de la etiqueta @Inyectable() y crearemos un objeto del tipo Subject:

`estadisticasSubject = new Subject<any>();

Ahora crearemos dos métodos en el servicio, uno para devolver un observable al componente que lo necesite y otro que envía los datos actualizados a todos los que tengan el observable. Esto se hará aplicando al objeto Subject los siguientes métodos:

.next(datosEnviar); y `.asObservable();`

Para utilizarlo en un componente, inyectaremos el servicio por dependencias y en el ngOnInit() obtendremos el observable y nos suscribiremos, asignando su respuesta a una variable local. Ahora cada vez que se llame a la función de enviar los datos desde cualquier parte todos recibirán los datos actualizados.

1.3 Objetivo y Resultado

El resultado del proyecto cumple satisfactoriamente con los objetivos propuestos y en él tenemos las siguientes funcionalidades terminadas:

- Inicio de sesión que solo deja acceder con un usuario y contraseña correcto.
- Todas las Operaciones CRUD sobre las entidades de Usuarios, Artículos y Pedidos.
- Buscador que realiza búsquedas con cada carácter que introducimos con cada una de las entidades.
- Estadísticas globales del total de cada una de las entidades.
- Cierre de sesión

En cuanto a las funcionalidades sin terminar, queda pendiente añadir mensajes del éxito o fracaso de cada una de las operaciones del CRUD y del inicio de sesión que informen al usuario final. Ya que actualmente estos solo se muestran por consola.





1.4 Conclusión

El proyecto ha sido muy útil para afianzar todos los conocimientos adquiridos en el módulo de frontend, ya que en él se han aplicado todas las cosas vistas en clase.

El tiempo de desarrollo que hemos tenido lo considero adecuado, ya que antes de realizarlo pudimos practicar con ejercicios enfocados a conocimientos parciales del curso que eran necesarios para este proyecto final.

En cuanto a la dificultad, considero que el proyecto tiene una parte más fácil como es la de estructurar el proyecto y crear las operaciones contra la API y otra más complicada. Esta segunda parte sería el filtrado y extracción de estadísticas de las entidades de otras entidades, como por ejemplo, obtener el top 3 de artículos más pedidos por un usuario.

Tanto con el módulo de Back como con el de Front he adquirido una inmensidad de conocimientos sobre la visión global de un servicio. Esto ha sido gracias a los grandes profesionales que hemos tenido como profesores que además de animarnos a seguir demostraban pasión por lo que trasmitían.





2. - Memoria Descriptiva

A continuación, se muestran documentos gráficos del funcionamiento de la aplicación

2.1 Pantallas Angular

2.1.1 - Inicio de sesión



Ilustración 2. Inicio de sesión

2.1.2 - Usuarios

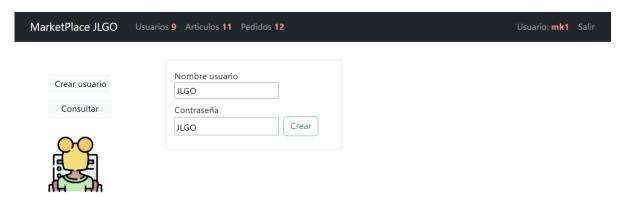


Ilustración 3. Creación de usuarios



Ilustración 4. Consulta de todos los usuarios





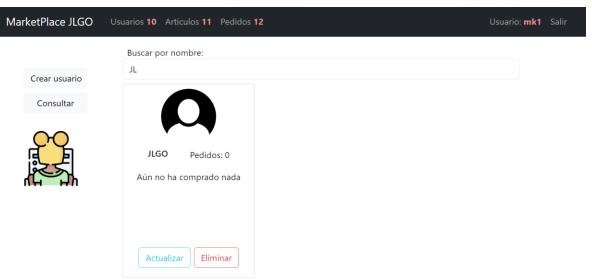


Ilustración 5. Filtrado de usuarios por nombre



Ilustración 6. Panel de actualización de usuario

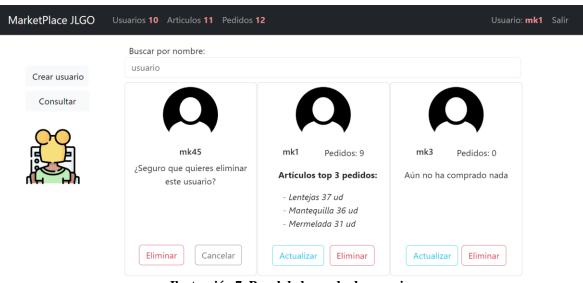


Ilustración 7. Panel de borrado de usuario





2.1.3 - Artículos



Ilustración 8. Creación de artículos



Ilustración 9. Consulta de todos los artículos

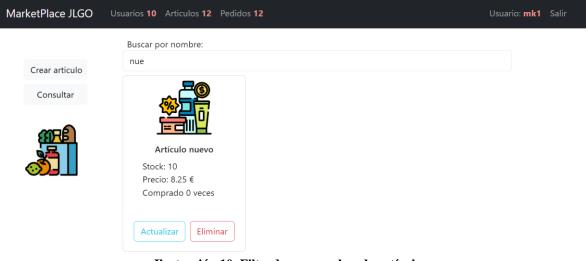


Ilustración 10. Filtrado por nombre de artículo







Ilustración 11. Panel de actualización de artículo



Ilustración 12. Panel de Borrado de artículo

2.1.4 - Pedidos

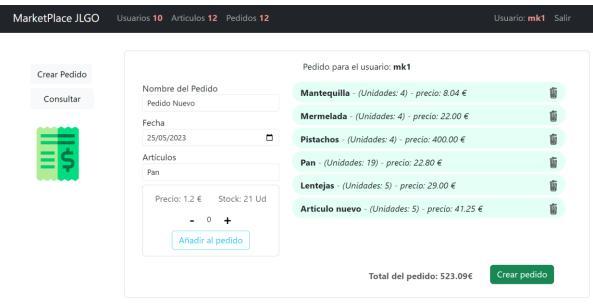


Ilustración 13. Creación de un pedido





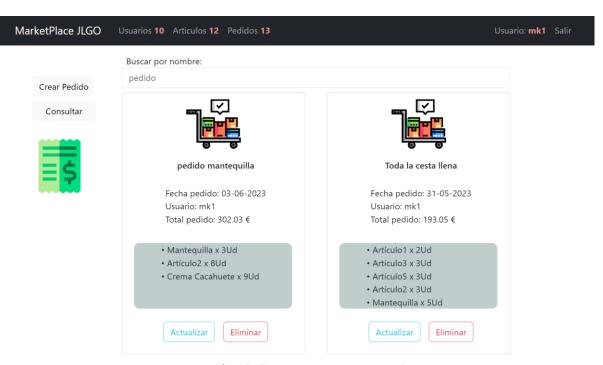


Ilustración 14. Consulta de todos los pedidos

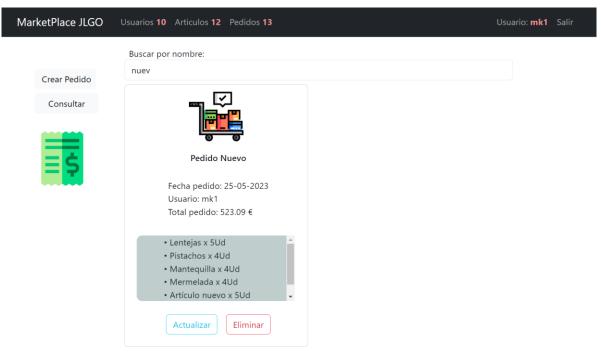


Ilustración 15. Filtrado por nombre





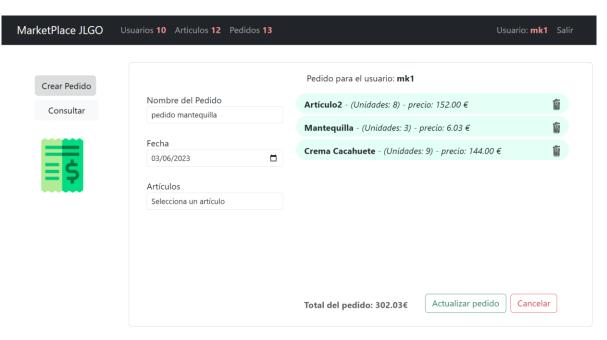


Ilustración 16. Panel de actualización de pedido

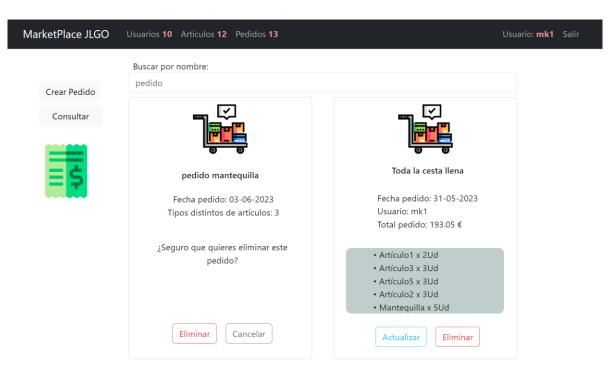


Ilustración 17. Panel de borrado de pedido



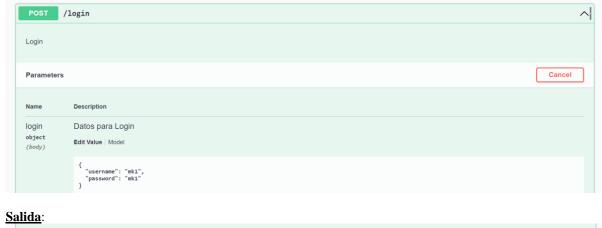




2.2 API REST

2.2.1 Servicio Web - Login

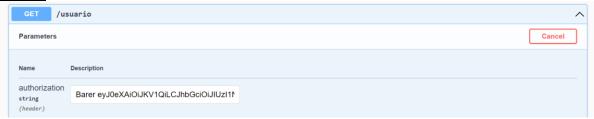
Entrada:





2.2.2 Servicio Web - Obtención de usuarios

Entrada:



Salida:

