



SDI – Sistemas Distribuidos e Internet

ENUNCIADO PRÁCTICA 2 – NodeJS - SW

INFORME

Grupo 2021-105-2021-107

Nombre1:	Alberto
Apellidos1:	Fernández Gutiérrez
Email1:	UO265181@uniovi.es
Cód. ID GIT	2021-105
% Participación	50
Nombre1:	Xurde
Apellidos1:	García Fernández
Email1:	UO271033@uniovi.es
Cód. ID GIT	2021-107
% Participación	50



Índice

RAZONAMIENTO DE LOS PORCENTAJES DE PARTICIPACIÓN	3
INTRODUCCIÓN	3
MAPA DE NAVEGACIÓN	4
ASPECTOS TÉCNICOS Y DE DISEÑO RELEVANTES.....	5
INFORMACIÓN NECESARIA PARA EL DESPLIEGUE Y EJECUCIÓN	6
CONCLUSIÓN	6



Razonamiento de los porcentajes de participación

Dividimos la práctica según los puntos asignados a cada apartado para lograr que ambos realicemos un trabajo equitativo. Durante el transcurso de esta fuimos ayudándonos a implementar las partes que nos resultaban más complejas, por lo que la distribución de apartados según los puntos no refleja el tiempo y esfuerzo dedicado por cada uno. Al final realizamos gran parte de la práctica de forma conjunta y ambos pensamos que nuestro trabajo es igualitario.

De todas formas, esta es una esquematización de nuestra idea inicial:

Aplicación Web:

-Alberto -> W1-5 -> 1'5 puntos

-Xurde -> W6-11 -> 1'8 puntos

Servicios Web:

-Xurde -> S1-4 -> 2 puntos

-Alberto -> C1-3 -> 1,2 puntos

Otros:

-Alberto -> Tests y Documentación, *opcional W12*



Introducción

Wallapop es un sistema que hemos desarrollado para la práctica. Consta de una aplicación web en Node.js, de una serie de servicios web (api rest) y un cliente que se sirve de estos servicios. Wallapop se sirve de una base de datos en MongoDB para la gestión de usuarios, ofertas y chats.

Las credenciales de la cuenta de mongoDB son:

->Email: ferberto136@gmail.com

->Contraseña: sdimongodb1

Mapa de navegación

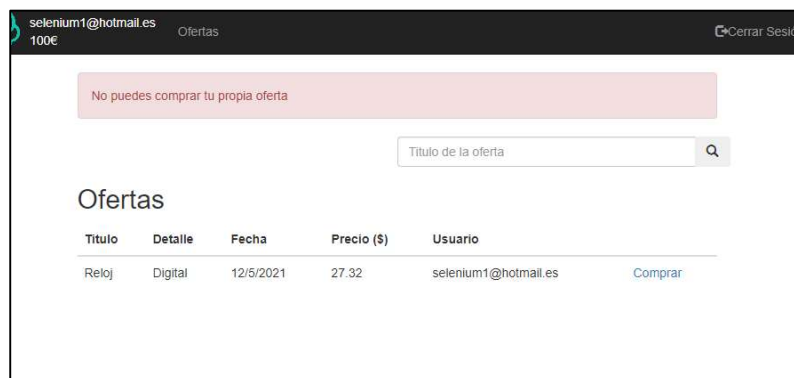




Aspectos técnicos y de diseño relevantes

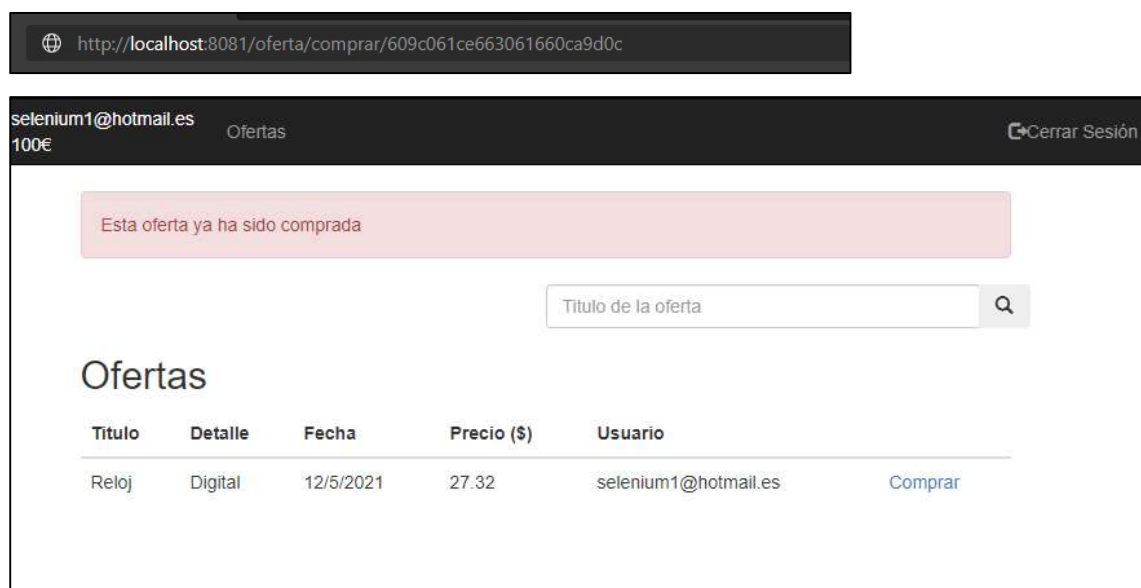
-Sobre comprar ofertas:

En la aplicación web, cuando un usuario identificado accede a buscar ofertas se le muestran todas las ofertas no compradas que figuran en el sistema, incluyendo las que él mismo ha creado. A pesar de esto, no le es posible comprar sus propias ofertas y el sistema se lo notifica:



Las ofertas se muestran para que el usuario pueda comprobar que su oferta está a la vista de todos los usuarios como una oferta más. Así podrá compararla con el resto de las ofertas.

Aunque no se muestren las ofertas compradas, contemplamos el caso en el que un usuario pueda conocer la id de una oferta comprada e intentar realizar una petición a la aplicación:





-Sobre ver usuarios:

Solo los administradores pueden acceder a esta vista ([admin@email.com|admin](#)). Si cualquier otro usuario estándar intenta acceder o borrar un usuario será redirigido. En esta vista no se listan los administradores y tampoco pueden ser eliminados.

También contemplamos que un usuario intente borrar una compra que no es suya.

Información necesaria para el despliegue y ejecución

En el proyecto hemos usado todas (o casi) las librerías usadas durante las prácticas, así que su instalación es necesaria para el correcto funcionamiento del proyecto. Además, también utilizamos:

Loguer log4js: > ***npm install log4js***

Driver de Mongo para java, mongo-java-driver-3.4.2: >



mongo-java-driver-
3.4.2.jar

Al ejecutar las pruebas se elimina toda la información de la base de datos y se introducen unos de prueba. Las pruebas manipulan estos datos. Cuando finalizan los datos persisten hasta que se vuelven a iniciar las pruebas. Cuando las pruebas acaban se puede acceder al sistema con (entre otros):

[admin@email.es](#) | admin

[prueba1@prueba1.com](#) | prueba1

[prueba2@prueba2.com](#) | prueba2

Conclusión

Hemos logrado desarrollar un sistema distribuido aplicando los conocimientos adquiridos en clase, tanto teóricos como prácticos. Nos dimos realmente cuenta de la importancia la seguridad como una característica transversal al sistema. Por último, hemos de decir que pudimos experimentar con mayor cercanía las ventajas y desventajas que suponen.