



SDI – Sistemas Distribuidos e Internet

ENUNCIADO PRÁCTICA 2

Node.js y Servicios web

2022/2023

INFORME

Nombre/Apellidos, ID-GIT#1	Álvaro Davila Sampedro, 2223-302, UO284548@uniovi.es
Nombre/Apellidos, ID-GIT#2	Israel Solís Iglesias, 2223-312, UO282162@uniovi.es
Nombre/Apellidos, ID-GIT#3	Omar Teixeira González, 2223-313, UO281847@uniovi.es
Nombre/Apellidos, ID-GIT#4	David Leszek Warzynski Abril, 2223-315, UO278968@uniovi.es
Cód. ID EQUIPO	32
Repositorio Github	https://github.com/Omitg24/SDI2223-entrega2-32





Índice

INTRODUCCIÓN	3
MAPA DE NAVEGACIÓN	4
MODELO DE DOMINIO	5
ASPECTOS TÉCNICOS Y DE DISEÑO RELEVANTES.....	6
INFORMACIÓN NECESARIA PARA EL DESPLIEGUE Y EJECUCIÓN	7
CONCLUSIÓN	7



Introducción

Para la realización del proyecto “*UrWalletPop*”, hemos utilizado el proyecto anterior como base, añadiendo las modificaciones correspondientes vistas en las clases de laboratorio, por lo que, de nuevo, la mayoría de los direccionamientos se producen a través de la barra de navegación. Además, en cuanto a estilos se ha vuelto a tomar la estructura anterior, siguiendo el esquema de colores, rojo, negro, gris y blanco.

Basarnos en el proyecto anterior o en las prácticas ha sido de gran utilidad para agilizar el desarrollo, sobre todo en estos momentos finales del semestre, por lo que, en algunos casos que hubo problemas o dudas ante como enfocar alguna funcionalidad en concreta, contar con esas bases ha sido de bastante ayuda.

El proyecto se diferencia en 2 partes, la parte 1 es la que sigue la estructura del anterior y la parte 2 se basa en lo visto en las prácticas de laboratorio con la API REST.

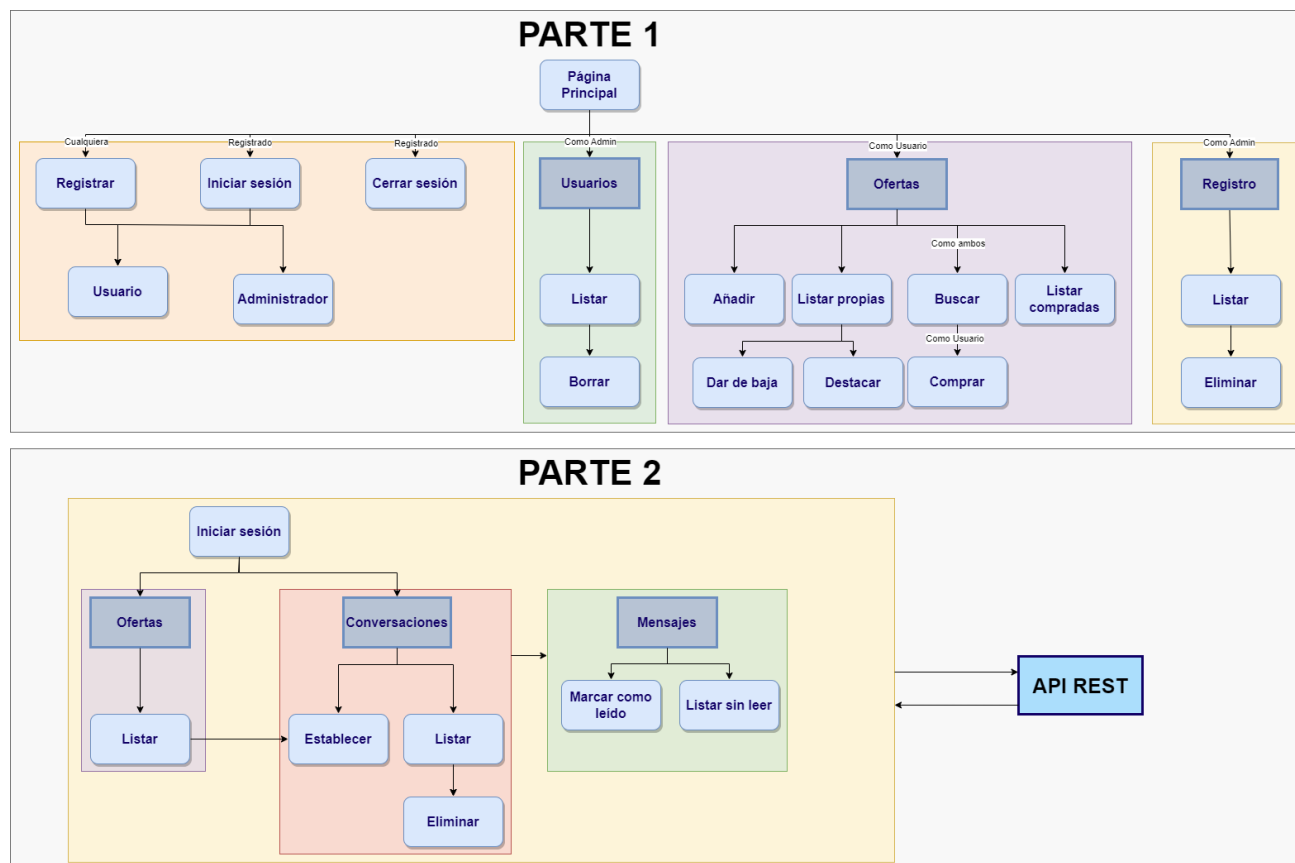
Comenzando con la parte 1, al ser el mismo proyecto (o prácticamente), gran parte de la funcionalidad es idéntica, distinguiendo 2 roles claros, administrador y estándar, donde, el administrador, puede acceder a un listado de usuarios, de forma que puede ver cuantos se encuentran en el sistemas y borrarlos, además, también cuenta con un sistema de registro o logging, en donde se pueden ver las peticiones realizadas por todos los usuarios de la aplicación en cuestión. También se ha decidido que el administrador pueda ver la lista de ofertas publicadas a modo de control de las ofertas que se suben, algo importante en caso de que una oferta viole las políticas de la web.

Con respecto a un estándar, este puede añadir ofertas a la aplicación, las cuales puede destacar para otros usuarios, dichas ofertas podrán ser visualizadas desde la vista de ofertas propias, donde, como su propio nombre indica, el usuario verá las ofertas que haya publicado hasta el momento, o desde la pestaña de publicadas, donde verá todas aquellas ofertas que otros usuarios hayan subido. Además de esto, el usuario podrá comprar ofertas subidas por otros, gastando dinero de su saldo inicial (100), lo que producirá un aumento en el saldo del otro.

Pasando a la parte 2, se ha implementado una API Rest, implementando algunas de las funcionalidades de la parte 1 (inicio de sesión o listado de ofertas), algunas del proyecto anterior (conversaciones) y otras nuevas (mensajes leídos). En este caso, no hay una distinción de roles, ya que se utiliza como usuario estándar y lo que se ha implementado corresponde al inicio de sesión con *token*, posteriormente, este usuario podrá ver un listado de las ofertas de otros usuarios y establecer una conversación por una oferta que le interese, dichas conversaciones contendrán mensajes que tendrán un estado indicando si ha sido leído o no, a su vez, el usuario tendrá acceso a un listado de las conversaciones, donde podrá eliminarlas o ver el número de mensajes sin leer de una conversación.

Mapa de navegación

El mapa de navegación se ha dividido en 2 partes, cada una correspondiente a la parte del proyecto, sin embargo ambas tienen en común que el desplazamiento se produce a través de la barra de navegación superior (con alguna excepción).



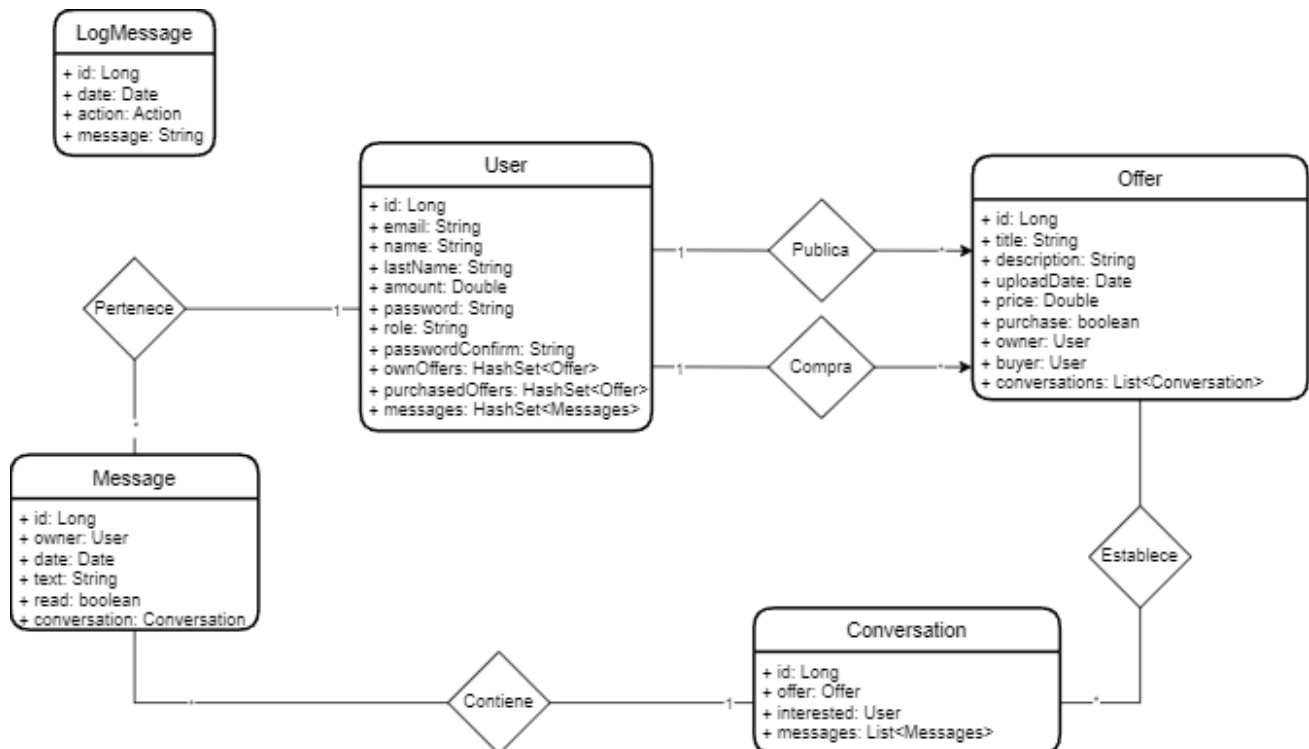
En la parte 1, salvando acciones como el borrado de usuarios o del registro, además de dar de baja, destacar o comprar una oferta, se puede acceder a través del panel de navegación.

Con respecto a la parte 2, únicamente se puede acceder a través de la barra de navegación a la sección de ofertas o listado de conversaciones, el resto de accesos se realiza a través de otros elementos en cada una de las vistas correspondientes.



Modelo de Dominio

En base a lo comentado en la introducción, en la aplicación están presentes las siguientes entidades, siendo la principal el usuario, ya que es para el que se desarrolló la aplicación, un usuario puede publicar ofertas o comprar las publicadas de otros, además de que, a través de una oferta, éste establecerá una conversación con el propietario, la cual contendrá varios mensajes, que estarán relacionados con el usuario que los envió.



Cabe mencionar que hay otra entidad que no se relaciona con ninguna de las anteriores de manera directa, pero que es la encargada de realizar el registro de acciones, por lo que, indirectamente se encuentra relacionado con el resto.



Aspectos técnicos y de diseño relevantes

Para el desarrollo del proyecto, se han tomado diversas decisiones con respecto al diseño o distribución de la aplicación, por ejemplo, para seguir utilizando la estructura del proyecto anterior, se ha utilizado la versión 4.5.2 de Bootstrap, en lugar de la utilizada en las prácticas de laboratorio, pero también se han tomado decisiones en cuanto a aspectos propios de la web, como que el administrador pueda acceder al listado de ofertas (algo que no se indica en el PDF), de forma que pueda llevar un control de dichas ofertas, tal y como se ha explicado en la introducción, otra decisión que se tomó fue la del cambio del listado de ofertas (también hecho en el proyecto anterior), ya que, en clase se hizo a modo de tabla, pero para el proyecto se ha hecho utilizando cards, de forma que la presentación fuese más familiar a otras páginas de compra-venta de productos. También se ha tomado la decisión de, en la parte 2, implementar los validadores empleando Express-Validator, sin embargo, y esto se relaciona directamente con el siguiente párrafo, se intentó hacer esto para la parte 1, pero, tras varios errores y algunos aspectos en los que se han encontrado dificultades (hacer comprobaciones en las que dos campos sea iguales o el acceso a una base de datos), además de que estas validaciones, como en el caso de la validación de datos del registro, ya estaban hechas desde hace 2 semanas.

A lo largo de este desarrollo, se han encontrado ciertas dificultades en algunos aspectos, como el paso de parámetros (usuario, rol, dinero, fecha de nacimiento, etc...) a las vistas, ya que, en el anterior proyecto, el guardado en sesión y utilización de estos en las vistas era instantáneo, sin embargo aquí, en cada petición Get, se debía pasar los parámetros a la vista, algo que a simple vista no da una buena impresión, pero que, sin embargo y tras varios esfuerzos por mejorarlo, ha sido imposible. Sin embargo, también ha habido dificultades en la parte del log, concretamente, detectar cuando un usuario iniciaba sesión con éxito o fallaba, lo que terminó arreglándose añadiendo el log en el post del login, para esos casos mencionados, otra por ejemplo podría ser la realización de tests para pruebas de la parte 2, entre otras.



Información necesaria para el despliegue y ejecución

Para utilizar la aplicación, se debe lanzar la base de datos y posteriormente la aplicación, la cual se conecta en el puerto 8081, de manera que mediante el enlace: <http://localhost:8081>, seremos redirigidos a la vista inicial de éste.

La base de datos se ha desplegado en local (tal y como se ha pedido).

Entre los datos de ejemplo, se puede acceder a la aplicación como administrador o como uno de los otros 15 usuarios comunes, de esta forma:

- Administrador:
 - Email: admin@email.com.
 - Contraseña: admin.
- Usuario:
 - Email: user01@email.com, user02@email.com, user03@email.com, etc...
 - Contraseña: user01, user02, user03, etc...

Para la realización del proyecto, se han utilizado varias tecnologías, cada una con su respectiva versión, estas tecnologías son:

- Node 16.19.1.
- MongoDB 4.1.4.
- JQuery 3.5.1.
- Selenium 3.141.59.
- Bootstrap 4.5.2.
- El resto de versiones se encuentran en el fichero package.json o en el pom.xml del proyecto de tests.

Cabe mencionar que se han empleado las mismas versiones a las utilizadas en la asignatura, de manera que se simplifique el proyecto a la hora de crearlo y que se cumpla con los criterios de uso obligatorio.

Conclusión

De forma general, la práctica nos ha servido para demostrar el conocimiento aprendido en clase, además, también nos ha servido para ser capaces de solucionar errores que nunca antes habíamos visto y que han surgido a lo largo del desarrollo de forma inesperada. Además, se han llevado a cabo varias reuniones para organizar la distribución de tareas y ver como estaba el proyecto en ese momento, además, se ha colaborado para realizar el punto de la seguridad en común, haciendo que cada desarrollador se encargase de la seguridad de sus apartados.

Pasando a las conclusiones individuales:

- Álvaro Davila Sampedro, UO284548:
Para la realización del proyecto, el estudiante se ha encargado de realizar los puntos 10, 11, 12 correspondientes a la compra de ofertas, lista de ofertas compradas y al marcado de una oferta como destacada de la aplicación web. En cuanto a la restapi y el cliente JQuery, el S7 (marcar



mensaje como leído), el C6 (Marcar mensajes como leídos de forma automática) y el C7 (Mostrar el número de mensajes sin leer). Todos estos incluyendo sus tests.

En cuanto a las dificultades encontradas, la mayor fue en el punto C6 a la hora de que se refrescasen los mensajes de forma automática. Con respecto a las ventajas encontradas, se agradece la existencia de una clase solo dedicada a la resolución de dudas sobre el desarrollo del proyecto y además con el desarrollo del mismo mejora el conocimiento de Node.js y de Javascript. En cuanto a las desventajas, la mayor fue a la hora de realizar los tests con Selenium, el uso de XPath para la búsqueda de los elementos y el uso de una restapi ya que hasta esta asignatura nunca había utilizado una.

- Israel Solís Iglesias, *UO282162*:

Durante el desarrollo del proyecto se han realizado distintas tareas como borrados múltiples de usuarios, registro de las acciones del sistema. Además de la parte 2 tanto en la parte de servicios como de cliente se ha realizado las tareas de enviar mensajes a una oferta, obtener los mensajes de una conversación y obtener el listado de conversaciones

Ventajas:

- La creación de un servicio web rest
- A nivel de trabajo en grupo ha sido una ventaja trabajar con gente conocida ya que hay más confianza a la hora de tomar decisiones y conocemos el estilo de trabajo de cada uno.
- El uso de twig para agilizar el trabajo.

Dificultades:

- Conversaciones: A la hora de obtener la información y restringir el acceso a los usuarios
- Test: A la hora de testear alguna prueba que ha dado errores por XPath y de la carga de elementos.
- Test: A la hora de realizar las pruebas de la API por la falta de experiencia.

- Omar Teixeira González, *UO281847*:

Para este proyecto, el estudiante se ha vuelto a encargar de los apartados que lo tocaron anteriormente, es decir, se encargó de la creación del repositorio y del proyecto, así como de la configuración de este, ya sea mediante la creación de las vistas generales (layout, home, inicio, página de error, etc...), además de las configuraciones necesarias en el fichero app.js, ya sea de configuración de la base de datos, encriptación de contraseñas, seguridad de acceso a determinadas áreas de la web o los propios routers que gestionan esto, sin embargo, pasando a los puntos concretos, el estudiante se ha encargado de los puntos 1 al 4 de la parte 1 y de los puntos 1 de la parte 2, todos correspondientes al manejo de usuarios. Cabe mencionar, que de igual forma que el resto de compañeros, también se han realizado los tests correspondientes, siendo estos los tests del 1 al 11 de la parte 1, así como 38, 39 y 40 de la parte 2^a y 48, 49 y 50 de la parte 2^b.

Además, el estudiante se ha encargado de la redacción de este documento, incluyendo los diferentes esquemas (mapa de navegación y modelo de dominio) y la creación de los diseños de la web (ya realizados en la entrega anterior).

Pasando a las ventajas encontradas, ha resultado de gran utilidad hacer el trabajo con el mismo equipo del proyecto anterior, así como también ha sido que la temática fuera la misma (Wallapop), lo que nos ha servido para distribuir el trabajo más rápido y saber cómo proceder la mayoría del tiempo. Además, el hecho de poder ir preguntando las dudas a lo largo del desarrollo



(ya sea en la última clase dedicada a dudas o a lo largo de las otras clases de laboratorio), también resulta de gran utilidad para no ir a última hora con una gran cantidad de trabajo por hacer.

Con respecto a las desventajas, se hubiese agradecido que se comentase en las clases de laboratorio ciertas cosas que iban a afectar al desarrollo, véase la mejor forma de realizar los test para la parte 2 (que se da un ejemplo), u otras cosas que afectaron más directamente, como por ejemplo, el paso de parámetros a las vistas de atributos guardados en sesión, algo que se habló con el profesor pero no se consiguió llegar a nada, por lo que tuvo que dejarse de la forma que se sabía (pasándoselo al render en cada petición Get),

- David Leszek Warzynski Abril, *UO278968*:

Se han realizado las tareas relativas a añadir ofertas, listar ofertas propias, eliminar ofertas propias y buscar ofertas, además del mostrado de ofertas disponibles tanto en la API como en los servicios con AJAX/JQuery, incluyendo para estas sus respectivas pruebas unitarias, por otra parte se han realizado los diferentes pageobjects que se han utilizado para la implementación de las pruebas unitarias, también se ha realizado una búsqueda de documentación y tutoriales enfocados al uso de Bootstrap para el mostrado de las ofertas.

Las desventajas que he encontrado en el desarrollo se han centrado en el desarrollo de las pruebas unitarias debido al uso de XPath para la búsqueda de elementos y la realización de los test para la parte de API que requieren pasar el token obtenido en el login a la petición para la búsqueda de ofertas y la posterior obtención de la lista con las mismas a través de la respuesta.

Por otro lado las ventajas que han encontrado, han sido a la hora de realizar las listas de ofertas aprovechando el uso de Bootstrap para dar una mejor imagen a la estética de la página, este proyecto también ha aportado mayor experiencia con el uso de control de versiones usando GitHub, la consumición y creado de APIs y el manejo de un trabajo en grupo con personas con las que no tengo relación y con las que sí, el proyecto ha servido para encontrar solución a problemas apoyándonos en miembros del equipo, como por ejemplo para la implementación de las conversaciones.