



## SDI – Sistemas Distribuidos e Internet

### ENUNCIADO PRÁCTICA 2

#### Node.js y Servicios web

2022/2023

### INFORME

Nombre/Apellidos, ID-GIT#1	Aarón Orozco Fernández, 2223-508, uo281997@uniovi.es
Nombre/Apellidos, ID-GIT#2	Eduardo Blanco Bielsa, 2223-503, uo285176@uniovi.es
Nombre/Apellidos, ID-GIT#3	Santiago López Laso, 2223-506, uo277369@uniovi.es
Nombre/Apellidos, ID-GIT#4	Chen Xin Pan Wang, 2223-510, uo276967@uniovi.es
Nombre/Apellidos, ID-GIT#5	Diego Villa García, 2223-511, uo277188@uniovi.es
Nombre/Apellidos, ID-GIT#6	
Cód. ID EQUIPO	53
Repositorio Github	<a href="https://github.com/gitblanc/sdi2223-entrega2-53">https://github.com/gitblanc/sdi2223-entrega2-53</a>



## Índice

<b>INTRODUCCIÓN .....</b>	<b>3</b>
<b>MAPA DE NAVEGACIÓN .....</b>	<b>4</b>
<b>ASPECTOS TÉCNICOS Y DE DISEÑO RELEVANTES.....</b>	<b>5</b>
<b>INFORMACIÓN NECESARIA PARA EL DESPLIEGUE Y EJECUCIÓN .....</b>	<b>6</b>
<b>CONCLUSIÓN .....</b>	<b>8</b>

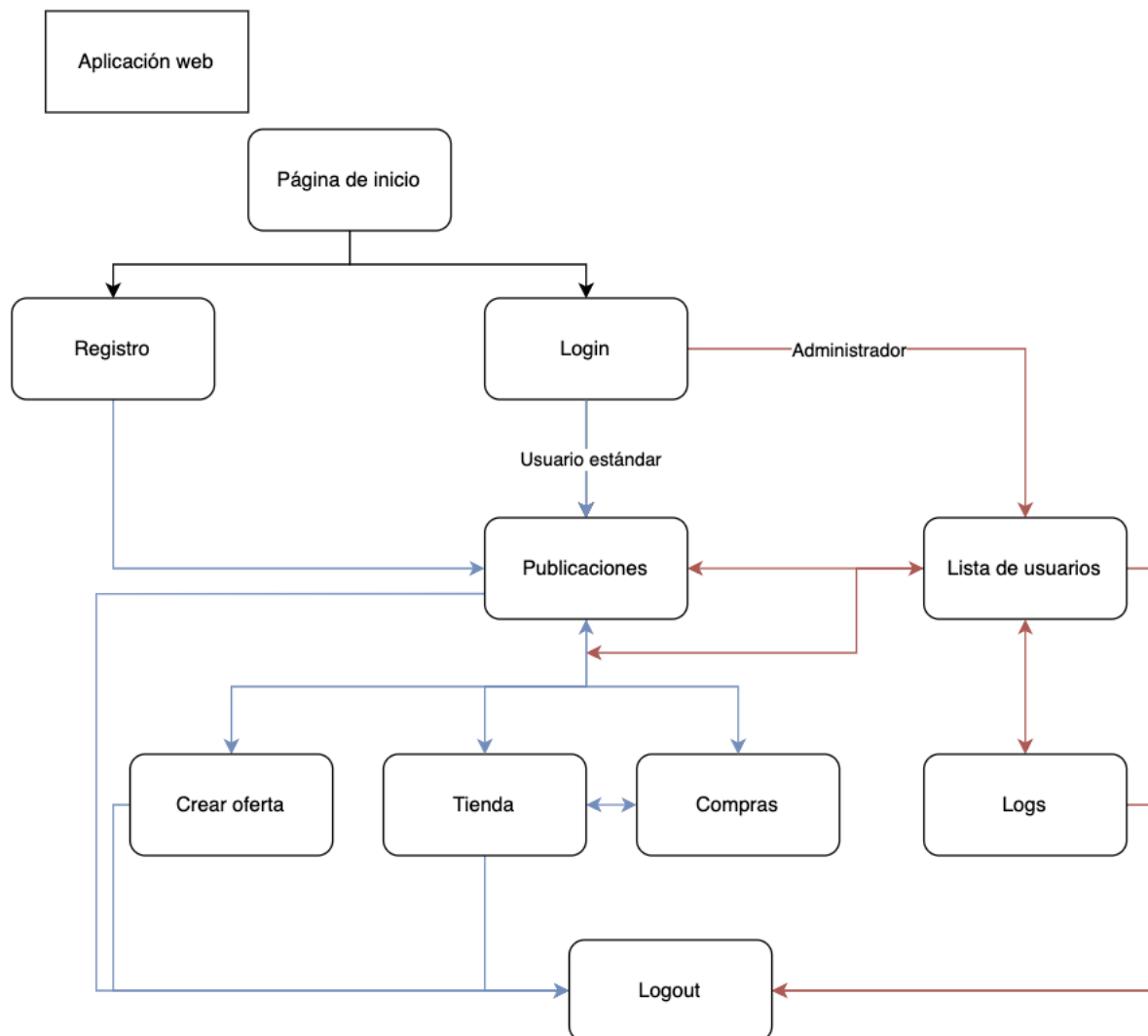


## Introducción

**MyWallapop** es una aplicación basada en Node.js y en servicios web creada para la asignatura de Sistemas Distribuidos e Internet. En esta aplicación se plasma parte de la funcionalidad de la tan conocida aplicación Wallapop, pero en una versión más reducida y funcional. En ella, como usuario, podrás comprar ofertas, ver las existentes, buscar las que te interesen, publicar nuevas ofertas y entablar conversaciones con otros usuarios. Además, se ha desarrollado una **API** que permitirá a un cliente mediante una interfaz gráfica listar las ofertas y entablar conversaciones en función de sus intereses.



## Mapa de navegación

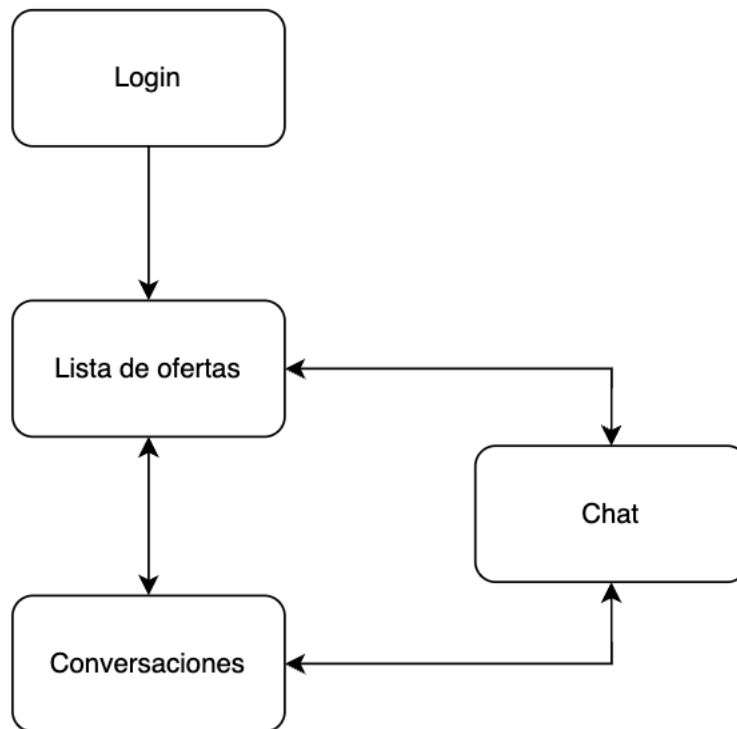


Administrador: → (flecha roja)  
Usuario estándar: → (flecha azul)

**Nota:** el usuario administrador puede realizar todas las funciones y acceder a las mismas vistas que un usuario estándar.



API REST



## Aspectos técnicos y de diseño relevantes

- Aplicación web
  - **Registro de usuario:** se ha realizado un formulario con los datos necesarios para el registro, sean email, nombre, apellidos y fecha de nacimiento. A su vez se han realizado unas series de comprobaciones que los campos deben cumplir. No puede haber otro usuario con el mismo email en la base de datos, la fecha de nacimiento debe ser posterior a la actual y ambas contraseñas deben ser iguales.
  - **Inicio de sesión:** las únicas comprobaciones que se hacen es que los campos no estén vacíos y que nombre de usuario y la contraseña sean correctos.
  - **Listado de usuarios y borrado de usuarios:** una vez se inicia sesión con el administrador, se podrán ver todos los usuarios que se encuentran en la base de datos (excepto el propio administrador). El administrador mediante un checkbox podrá eliminar a los usuarios que quiera. A la hora de eliminar un usuario todo lo relacionado con él será también borrado (desde sus ofertas, compras, chats y mensajes).
  - **Logger** de la aplicación: en un principio se pensó en emplear librerías externas tales como Log4js o Winston, sin embargo, dada la excesiva carga a la aplicación se decidió crear un logger propio y simple con la función necesaria genérica para almacenar los logs de la aplicación. Dicha función, añade un log de un tipo determinado a una colección dentro del repositorio mywallapop llamada “logs” con su mensaje y su fecha correspondientes.




- **Listado y borrado de ofertas:** una vez que se inicia sesión con un usuario que esté previamente registrado, dicho usuario podrá dar de alta ofertas con un título descriptivo de la oferta, una descripción detallada, la fecha de publicación y la cantidad solicitada en euros. Una vez creada dicha oferta también se le tendría que mostrar al usuario una opción de eliminar/ dar de baja esa oferta y se eliminará la oferta y toda la información relacionada con ella.
- **Compra de ofertas:** una vez que se inicia sesión con un usuario que esté previamente registrado también, el usuario podría acceder a la ventana de Tienda y comprar ofertas de todos los usuarios. No podría comprar ofertas que ha dado de alta él mismo, ni aquellas ofertas cuyo coste sea mayor que el saldo que tenga ni ofertas que ya han sido vendidas por otro usuario. Esas compras se listan en otra pestaña donde se puede ver la fecha exacta de su compra.
- Servicios web
  - API de servicios web rest
    - **Identificación de usuario:** se ha hecho un login de forma que cuando el usuario se logué se genere un token que identificara la sesión del usuario.
    - **Listado de ofertas:** comprobando el usuario que se encuentra en sesión, se obtendrán las ofertas que no sean del usuario.
  - Cliente rest
    - **Autenticación del usuario** (widget-login.html): en el caso en el que el usuario inicie sesión correctamente, se enviará a la vista de ofertas. En el caso de que haya un error en el logeo se le mostrará a un usuario con un mensaje.
    - **Ofertas** (widget-offers.html): al iniciar sesión el usuario será redirigido a la vista de ofertas, pero solo se mostrarán las de los demás, nunca las suyas. Se ha añadido una funcionalidad extra que permite que si un usuario ya entabló una conversación con el vendedor de dicha oferta no se generará un chat nuevo, es decir, el botón de chateo se enlazará con el chat existente y no se perderán los mensajes ya escritos.
    - **Chats** (widget-chat.html): un usuario podrá tener un chat para cada oferta que le interese. Se ha implementado una funcionalidad que permite que los mensajes se envíen en tiempo real con un timeout de unos 2 segundos (aproximadamente). Es decir, no es necesario recargar la aplicación para ver los nuevos mensajes. El chat ha sido diseñado de forma que todos los mensajes se ven como una tabla con el usuario que envía el mensaje, el contenido de éste y su fecha.
    - **Conversaciones** (widget-conversations.html): un usuario puede tener múltiples conversaciones (pero sólo una por oferta que le interese). Desde el apartado de conversaciones será capaz de ver las conversaciones que ha entablado y podrá acceder a los chats desde ellas.

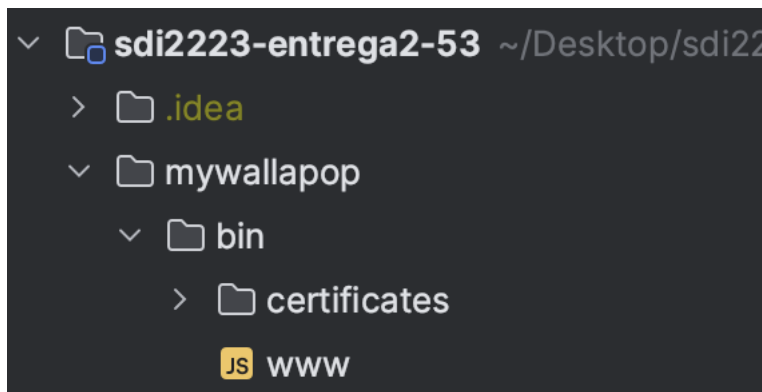
## Información necesaria para el despliegue y ejecución

Versiones de software usados:

- **Node.js:** v 16.20.0
- **Mongodb:** v 4.7.1
- **Mongodb Compass:** última versión
- **Postman:** última versión (exclusivo para realizar pruebas a la API)
- **Express:** última versión
- **Cookie-parser:** última versión

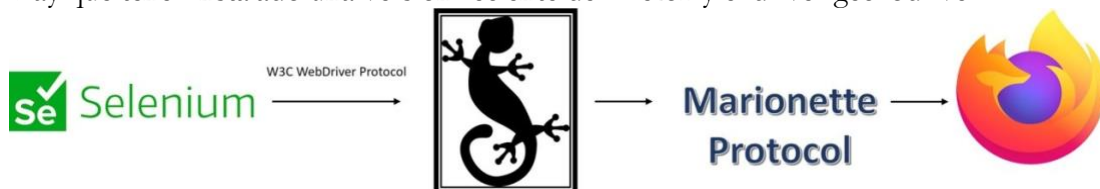
Instrucciones de despliegue de la aplicación:

- Tener instalado una versión razonablemente actual de [IntelliJ Ultimate](#) o cualquier otro editor que permita ejecutar Node.js
- Instalar MongoDB y MongoDB Compass:  <https://www.mongodb.com/docs/manual/installation/>
- Dentro de la carpeta de mywallapop hacer un `$ npm install`
- Con el editor que desees (en este caso haremos uso de IntelliJ) ejecuta el archivo `www` (con la versión de Node **16.20.0** mencionada anteriormente), localizado en la carpeta `/bin` del proyecto:



Instrucciones de despliegue de las pruebas:

- En IntelliJ abrir el proyecto `sdi2223-entrega2-tests-53`
- Vaciar la base de datos myWallapop para que no haya más datos que los datos de prueba que se insertan al principio de los tests
- Ejecutar la aplicación principal (archivo `www`) en la otra ventana de IntelliJ Idea
- Hay que tener instalado una versión reciente de Firefox y el driver `geckodriver`.



- En la clase de tests `Sdi2223Entrega153ApplicationTests` retocar el path de Firefox y `geckodriver` a la localización del tester en cuestión.

```
class Sdi2223Entrega153ApplicationTests {  
    1 usage  
    static String PathFirefox = "C:\\Program Files\\Mozilla Firefox\\firefox.exe";  
    //static String GeckoDriver = "C:\\Users\\uo277369\\Desktop\\PL-SDI-Sesión5-material\\geckodriver-v0  
    1 usage  
    static String GeckoDriver = "C:\\Users\\mines\\Desktop\\wallapop\\geckodriver-v0.30.0-win64.exe";  
    //static String GeckoDriver = "C:\\Users\\aaron\\Desktop\\UNI\\tercero\\SEGUNDO\\SDI\\Practica\\SDI-  
    //static String GeckoDriver = "C:\\Dev\\tools\\selenium\\geckodriver-v0.30.0-win64.exe";  
    //static String PathFirefox = "/Applications/Firefox.app/Contents/MacOS/firefox-bin";  
    //static String GeckoDriver = "/Users/USUARIO/selenium/geckodriver-v0.30.0-macos";  
}
```

- Ejecutar la clase de tests  
`src/test/java/com/uniovi/sdi2223entrega2test/n/Sdi2223Entrega2TestApplicationTests.java`



## Conclusión

Miembro	Aportaciones	Ventajas	Desventajas
Aarón Orozco Fernández	<p>Mi contribución ha sido lo siguiente:</p> <p><b>Aplicación Web</b></p> <p>W1 Registrase como usuario, W2 Iniciar sesión, W3: Fin de sesión, W4: Listado de usuarios del sistema y W5: Borrado múltiple de usuarios.</p> <p><b>Servicios Web</b></p> <p>S1 Identificarse como usuario, S2 Obtener el listado de ofertas disponibles, C1 Autenticación del usuario y C2 Mostrar listado de ofertas disponibles.</p>	<p><b>Nivel técnico:</b> considero que el desarrollo de la aplicación ha ido bien debido a que mi parte del trabajo (siendo en ambas partes el login) era bastante esencial para que mis compañeros pudieran continuar con el trabajo y yo intente realizarlo y acabarlo lo más rápido posible</p> <p><b>Nivel de trabajo en grupo:</b> no hubo ningún problema, hubo una vez comunicación y más o menos fuimos cumpliendo con los plazos que nos fuimos asignando</p>	<p><b>Nivel técnico:</b> he conseguido resolver ciertos problemas como el mostrar en el navbar los datos del usuario, que al principio fue una complicación ya que no era igual que en el anterior proyecto. Por lo que tuve que investigar un poco hasta sacarlo adelante.</p> <p><b>Nivel de trabajo en grupo:</b> puedes ser que algunas tareas no estuvieran justo en el plazo que acordamos todo el equipo, pero en general no hubo nada grave.</p>
Eduardo Blanco Bielsa	<p>Mi principal función en la aplicación fue encargarme del monitoreo (logging) y la seguridad de ésta. Sin embargo, también realicé los puntos C3 y C4 de la API en colaboración con mi compañero Diego.</p> <p>También aporté diversos métodos a los apartados S2 y S3 de la API y ayudé en la corrección de errores generales de la aplicación.</p>	<p><b>Nivel técnico:</b> ha sido una gran ventaja crear un logger propio en lugar de emplear uno ya existente de otra librería, pues de esta forma he reducido severamente el impacto de las dependencias del proyecto, así como aumentado drásticamente el nivel de seguridad de la aplicación (7 vulnerabilidades menos). Por último, quiero mencionar que he podido crear un sistema de chateo en tiempo real con una latencia de unos 2</p>	<p><b>Nivel técnico:</b> tuve ciertas dificultades a la hora de conseguir realizar conversaciones en tiempo real entre vendedor y ofertante debido a cómo vincular un vendedor con el chat del ofertante correspondiente</p> <p><b>Nivel de trabajo en grupo:</b> la única pega mencionable se trata de la tardanza en disponer de la parte de conversaciones a bajo nivel, pero no fue nada realmente exagerado.</p>





		<p>segundos aproximadamente.</p> <p><b>Nivel trabajo de grupo:</b> ha habido una buena comunicación durante todo el transcurso del proyecto, por lo que no se han generado conflictos y nos hemos ayudado mutuamente.</p>	
Santiago López Laso	<p>He completado los requisitos W11, W12, S4 y S5. Además, implementé todas las pruebas funcionales de la parte 1 (desde la 1 hasta la 37). Por otro lado, ayudé a mis compañeros con errores de sus partes y a cómo resolver algunos problemas que les surgían.</p>	<p><b>Nivel técnico:</b></p> <p>Los requisitos W11 y W12 no supusieron ningún problema y gracias a las pruebas automatizadas que implementé conseguimos descubrir y resolver algunos errores en la aplicación.</p> <p><b>Nivel trabajo en grupo:</b></p> <p>Durante todo el proceso de desarrollo nos hemos comunicado para repartirnos el trabajo y resolver problemas, lo que permitió implementar todo lo necesario a tiempo. En general, hubo un buen trabajo de equipo y todos los miembros del grupo ayudaron en el desarrollo.</p>	<p><b>Nivel técnico:</b></p> <p>Me costó hacer los requisitos S4 y S5 porque no sabía cómo localizar un chat a partir de una oferta y un usuario, y de qué forma guardar los mensajes en MongoDB. Al final me basé en la práctica anterior para resolverlo, creando una colección chats con la oferta y el interesado y otra colección messages en el que cada mensaje tiene la id de su chat, además de otros datos como el autor y el texto.</p> <p><b>Nivel trabajo en grupo:</b></p> <p>Como la parte 2B depende de la parte 2A, algunos compañeros tuvieron que esperar a que los demás acabaran su parte para comenzar a trabajar. Aun así, no hubo problemas graves.</p>
Chen Xin Pan Wang	<p>He implementado los requisitos W9 y W10 de la parte 1 de la</p>	<p><b>Nivel técnico:</b></p> <p>Considero que este trabajo ha ido bien por</p>	<p><b>Nivel técnico:</b></p> <p>Al principio tuve un problema con el tema</p>



	<p>Aplicación Web además del requisito S3 de la parte de API de Servicios Web Rest.</p>	<p>mi parte también. En ciertas ocasiones los compañeros me tuvieron que ayudar a implementar alguna funcionalidad que se me había complicado un poco.</p> <p><b>Nivel trabajo en grupo:</b></p> <p>No hubo ningún problema importante, las dudas que teníamos nos lo íbamos consultando como equipo que somos así que todos los miembros del equipo han trabajado en el desarrollo del proyecto.</p>	<p>de comprar las ofertas porque no conseguía que se descontase correctamente las cantidades, pero me ayudó Santiago en clase a resolverlo y era un fallo pequeño.</p> <p>También a la hora de crear una conversación al principio lo intenté hacer de una manera (sin crear el chat) porque pensábamos que eso se hacía cuando se listaban los mensajes, pero al final decidí guiarme por el primer trabajo y hacerlo en función de cómo lo hicimos ahí.</p> <p><b>Nivel trabajo en grupo:</b></p> <p>El único inconveniente grande que vería de este trabajo es que algunos compañeros tuvieron que esperar por otros para complementar o realizar más funcionalidad.</p>
Diego Villa García	<p>Completé los requisitos W6, W7 y W8 de la parte de la aplicación web y, junto con Eduardo, los apartados C3 y C4 de la parte del cliente JQuery. También implementé las pruebas desde la 38 a la 42 y desde la 48 hasta la 54.</p>	<p><b>Nivel Técnico:</b></p> <p>Los requisitos W6, W7, y W8 me resultaron especialmente sencillos porque son los mismos que realicé en la primera entrega.</p> <p><b>Nivel trabajo en grupo:</b></p> <p>Hubo muy buena comunicación entre todos los miembros del equipo, y nos fuimos ayudando cuando teníamos dudas y corrigiendo pequeños</p>	<p><b>Nivel Técnico:</b></p> <p>En los requisitos C3 y C4 se me hizo un poco complicado entender cómo encontrar las conversaciones teniendo en cuenta que cualquiera de los dos usuarios tiene que poder verlas. Hubo que darle muchas vueltas a los métodos que teníamos en la API y sobre todo cómo llamarlos desde el cliente JQuery para</p>



		errores que nos iban quedando.	<p>obtener los resultados que queríamos.</p> <p><b>Nivel trabajo en grupo:</b></p> <p>Como algunas partes de la entrega necesitan que las anteriores estén terminadas y funcionando, había días que algunos miembros del equipo no tenían nada que hacer porque estaban esperando a que terminasen los demás.</p>
--	--	--------------------------------	---