



SDI – Sistemas Distribuidos e Internet

ENUNCIADO PRÁCTICA 2 – NodeJS - SW

INFORME **Grupo 711-708**

Nombre1:	Esther
Apellidos1:	González García de Vega
Email1:	UO269763@uniovi.es
Cód. ID GIT	711
% Participación	50
Nombre1:	Santiago
Apellidos1:	Fidalgo Salles
Email1:	UO265578@uniovi.es
Cód. ID GIT	708
% 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

Se deberá indicar (para equipos de dos personas) de forma razonada cómo se han repartido las tareas, aunque sea de forma general. Y en caso de discrepancia entre los miembros hacerlo constar aquí.

El trabajo ha sido realizado de forma equitativa mediante la realización de varias reuniones online por Teams y mediante la división del trabajo de la forma en la que creíamos que ambos trabajábamos lo mismo.

Introducción

Este trabajo se divide en dos partes, una dedicada a la **aplicación Web** y otra dedicada a los **servicios API web REST y Cliente ligero JQuery/AJAX**.

- Respecto a la parte dedicada a la aplicación Web:

MyWallapop es una aplicación Web de compraventa de artículos entre usuarios. Existirán tres tipos de perfiles de usuario:

- **Usuario público:** es el usuario que no está logueado y solo tiene acceso a la pantalla principal, a la de registro y a la de identificación.
- **Usuario registrado (usuario estándar):** es el usuario estándar que usa la aplicación, tiene acceso a todas las vistas, excepto a la vista del administrador (resetear base de datos y listar usuarios). Cuando un usuario se registra dispondrá de 100 euros en su cuenta, para gastar en ofertas y además habrá ciertas restricciones, como que no puede haber dos usuarios con el mismo email en la aplicación.
- **Usuario registrado (administrador):** es el encargado de administrar los usuarios y, en general, todos los datos de la base de datos. Tiene acceso a todas las partes de la aplicación.

Los requisitos de esta parte que hemos implementado son los siguientes:

1. Registrarse como usuario
2. Iniciar sesión
3. Fin de sesión
4. Listado de usuarios
5. Borrado múltiple de usuarios
6. Dar de alta una nueva oferta
7. Listado de ofertas propias
8. Dar de baja una oferta
9. Buscar ofertas
10. Comprar una oferta
11. Ver el listado de ofertas compradas



- Respecto a la parte del cliente ligero:

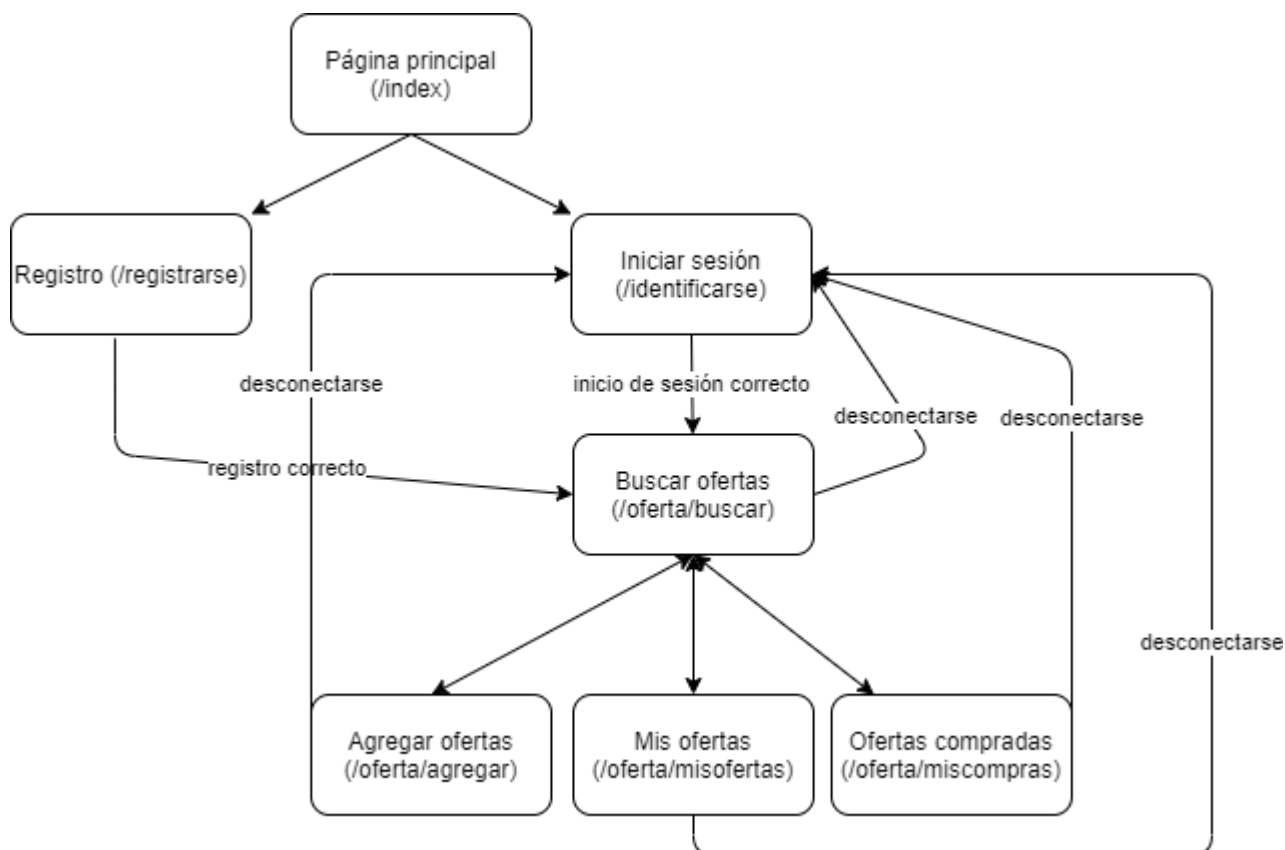
Esta parte incluirá las funcionalidades anteriores y, además, se incluyen nuevas funcionalidades (. Se divide a su vez en dos partes:

- Primera parte, las funcionalidades que hemos implementado son las siguientes:
 1. Identificarse como usuario vía token
 2. Usuario identificado. Mostrar el listado de ofertas disponibles
 3. Usuario identificado. Enviar mensajes a una oferta.
 4. Usuario identificado. Obtener los mensajes de una conversación
 5. Obtener el listado de conversaciones
 6. Eliminar una conversación
 7. Marcar mensaje como leído
- Segunda parte:
 - Autenticación del usuario
 - Mostrar el listado de ofertas disponibles
 - Enviar y mostrar los mensajes de una oferta
 - Ver el listado de conversaciones
 - Eliminar una conversación
 - Marcar mensajes como leídos de forma automática



Mapa de navegación

El mapa de navegación de la **primera parte** de un usuario estándar sería el siguiente:



El mapa de navegación de la **segunda parte** del cliente ligero:





Aspectos técnicos y de diseño relevantes

La distribución del proyecto sigue el siguiente patrón de directorios:

- **Modules**, encontraremos el gestor de la base de datos, en donde se realizarán todas las operaciones de base de datos.
- **Public**, almacenamos aquí los diferentes fragmentos del cliente de la segunda parte, además las imágenes y *cookies.js*
- **Routes**, donde se encuentran los diferentes controladores para las rutas, organizados según su función. Son el punto intermedio entre el acceso a la base de datos y la creación de vistas. Entre ellos está el controlador *radmin.js*, creado solamente para el reseteo de la base de datos en la realización de las pruebas.
- **Views** (vistas), se encuentran aquí las diferentes plantillas de vistas que se usarán para mostrar las páginas finales al usuario.

Además, a la misma altura que los directorios, está *app.js*, es la aplicación en sí.

Información necesaria para el despliegue y ejecución

Para el despliegue necesitaremos únicamente:

- Node.js (+v12).
- Conexión a internet para conectar con la BD Mongo.

El servidor se encontrará disponible en la dirección <https://localhost:8081>

Para ejecutar las pruebas con Selenium necesitaremos JDK 8 y un IDE compatible (STS).

Conclusión

Nos hemos dado cuenta de los conocimientos que hemos ido adquiriendo a lo largo de las distintas tareas realizadas en clase, y hemos sabido aplicarlas a una práctica diferente. A su vez este proyecto nos ha venido bien para mejorar el trabajo en equipo y aprender un poco más sobre las dinámicas de trabajo en grupo, gracias a Git hemos podido repartir el trabajo y trabajar de manera autónoma cuando lo necesitamos. Sin embargo, también hemos echado en falta el trabajo de forma presencial, ya que de esa manera hubiésemos ganado tiempo a la hora de resolver dudas y ayudarnos. En líneas generales estamos contentos con el trabajo realizado.