



UNIVERSIDAD
POLITECNICA
DE VALENCIA

UNIVERSIDAD POLITÉCNICA DE VALENCIA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA

Red social de reparaciones de coches

Memoria del proyecto final de carrera

Autor:

Luis Mielgo Olcina

Director:

Ismael Torres Boigues

Valencia, 24 de septiembre 2012

Índice

Tabla de imágenes.....	- 4 -
1 Introducción	- 7 -
1.1 Objetivos	- 7 -
2 Estado del arte	- 8 -
2.1 driverside.....	- 8 -
2.2 foodspotting.....	- 9 -
2.3 tallerator.....	- 9 -
3 Especificación	- 11 -
3.1 Introducción	- 11 -
3.2 Descripción General del Sistema.....	- 11 -
3.3 Requisitos de Gestión de Información	- 11 -
3.4 Requisitos Funcionales	- 12 -
3.5 Requisitos de Interfaz de Usuario	- 12 -
3.5.1 Usuarios Anónimos	- 12 -
3.5.2 Usuarios Registrados.....	- 13 -
3.5.3 Gestores del portal.....	- 13 -
4 Requisitos	- 14 -
4.1 Mapa navegacional	- 14 -
4.2 Descripción de las pantallas	- 15 -
4.2.1 Opciones generales	- 15 -
4.2.1.1 Inicio sin una sesión abierta	- 15 -
4.2.1.2 Registro.....	- 16 -
4.2.1.3 Mapa	- 17 -
4.2.1.4 Mejores talleres	- 18 -

4.2.1.5 Mejor valoradas	- 19 -
4.2.2 Opciones privadas	- 20 -
4.2.2.1 Inicio con una sesión abierta.....	- 20 -
4.2.2.2 Ver mis datos.....	- 21 -
4.2.2.3 Modificar mis datos.....	- 22 -
4.2.2.4 Añadir vehículo propio	- 23 -
4.2.2.5 Mis vehículos.....	- 24 -
4.2.2.6 Visualizar vehículos	- 25 -
4.2.2.7 Amigos.....	- 26 -
4.2.2.8 Enviar mensaje	- 27 -
4.2.2.9 Buscar amigos.....	- 28 -
4.2.2.10 Mensajes enviados	- 29 -
4.2.2.11 Mensajes recibidos.....	- 30 -
4.2.2.12 Insertar taller.....	- 31 -
4.2.2.13 Ver mis talleres.....	- 32 -
4.2.2.14 Visualizar taller.....	- 33 -
4.2.2.15 Modificar datos del taller	- 34 -
4.2.2.16 Insertar reparación.....	- 35 -
4.2.2.17 Últimas reparaciones.....	- 36 -
5 Implementación	- 37 -
5.1 Tecnologías y herramientas utilizadas	- 37 -
5.1.1 Lenguajes, BD, servidor,css	- 37 -
5.1.2 Herramientas de desarrollo	- 38 -
5.2 Estructura de la aplicación	- 38 -
5.3 Arquitectura	- 44 -
5.4 Diseño.....	- 47 -

5.4.1 Diagrama de clases.....	- 47 -
5.4.1.1 Descripción de las clases	- 48 -
5.4.2 Diagrama Entidad-Relación.....	- 49 -
5.4.3 Capa de presentación.....	- 50 -
5.4.4 Capa lógica	- 52 -
5.4.5 Capa de persistencia de datos.....	- 57 -
Anexo 1: Registro del dominio	- 69 -
Anexo 2: Modelo de negocio.	- 70 -
Anexo 3: Manual de usuario.....	- 71 -
6 Conclusiones.....	- 88 -
6.1 Fases del proyecto.....	- 88 -
6.2 Dificultades.....	- 88 -
6.3 Experiencia	- 88 -
6.4 Trabajo futuro	- 89 -
7 Bibliografía	- 90 -

Tabla de imágenes

Fig (1) driverside.com	2.1 driverside
Fig (2) foodspotting.com	2.2 foodspotting
Fig (3) tallerator.es	2.3 tallerator
Fig (4) Mapa navegacional	4.1 Mapa navegacional
Fig (5) captura inicio sin una sesión abierta	4.2.1.1 Inicio sin una sesión abierta
Fig (6) Mockup Registro de usuario	4.2.1.2 Registro
Fig (7) Captura Registro de usuario	4.2.1.2 Registro
Fig (8) Mockup Papa	4.2.1.3 Mapa
Fig (9) Captura Mapa	4.2.1.3 Mapa
Fig (10) Mockup Mejores talleres	4.2.1.4 Mejores talleres
Fig (11) Captura Mejores talleres	4.2.1.4 Mejores talleres
Fig (12) Mockup Reparaciones mejor valoradas	4.2.1.5 Mejor valoradas
Fig (13) Captura Reparaciones mejor valoradas	4.2.1.5 Mejor valoradas
Fig (14) Mockup Inicio con sesión abierta	4.2.2.1 Inicio con una sesión abierta
Fig (15) Captura Inicio con sesión abierta	4.2.2.1 Inicio con una sesión abierta
Fig (16) Mockup Ver mis datos	4.2.2.2 Ver mis datos
Fig (17) Captura Ver mis datos	4.2.2.2 Ver mis datos
Fig (18) Mockup Actualizar mis datos	4.2.2.3 Modificar mis datos
Fig (19) Captura Actualizar mis datos	4.2.2.3 Modificar mis datos
Fig (20) Mockup Añadir vehículo de usuario	4.2.2.4 Añadir vehículo propio
Fig (21) Captura Añadir vehículo de usuario	4.2.2.4 Añadir vehículo propio
Fig (22) Mockup Mis vehículos	4.2.2.5 Mis vehículos

Fig (23) Captura Mis vehículos	4.2.2.5 Mis vehículos
Fig (24) Mockup Visualizar vehiculo	4.2.2.6 Visualizar vehículos
Fig (25) Captura Visualizar vehículo	4.2.2.6 Visualizar vehículos
Fig (26) Mockup Amigos	4.2.2.7 Amigos
Fig (27) Captura Amigos	4.2.2.7 Amigos
Fig (28) Captura Enviar mensaje	4.2.2.8 Enviar mensaje
Fig (29) Mockup Buscar amigos	4.2.2.8 Enviar mensaje
Fig (30) Captura Buscar amigos	4.2.2.9 Buscar amigos
Fig (31) Mockup Mensajes enviados	4.2.2.10 Mensajes enviados
Fig (32) Captura Mensajes enviados	4.2.2.10 Mensajes enviados
Fig (33) Mockup Mensajes recibidos	4.2.2.11 Mensajes recibidos
Fig (34) Captura Mensajes recibidos	4.2.2.11 Mensajes recibidos
Fig (35) Mockup Insertar taller	4.2.2.12 Insertar taller
Fig (36) Captura Insertar taller	4.2.2.12 Insertar taller
Fig (37) Mockup Ver listado de mis talleres	4.2.2.13 Ver mis talleres
Fig (38) Captura Ver listado de mis talleres	4.2.2.13 Ver mis talleres
Fig (39) Mockup Visualizar datos taller	4.2.2.14 Visualizar taller
Fig (40) Captura Visualizar datos taller	4.2.2.14 Visualizar taller
Fig (41) Mockup Modificar datos del taller	4.2.2.15 Modificar datos del taller
Fig (42) Captura Modificar datos del taller	4.2.2.15 Modificar datos del taller
Fig (43) Mockup Insertar reparación	4.2.2.16 Insertar reparación
Fig (44) Captura Insertar reparación	4.2.2.16 Insertar reparación
Fig (45) Mockup Ultimas reparaciones	4.2.2.17 Últimas reparaciones
Fig (46) Captura Ultimas reparaciones	4.2.2.17 Últimas reparaciones
Fig (47) Diagrama de clases	5.4.1 Diagrama de clases
Fig (48) Diagrama Entidad-Relación	5.4.2 Diagrama Entidad-Relación

Fig (49) interfaz de usuario no registrado	5.4.3 Capa de presentación
Fig (50) boceto de la interfaz de usuario no registrado	5.4.3 Capa de presentación
Fig (51) interfaz de usuario no registrado	5.4.3 Capa de presentación
Fig (52) boceto de la interfaz de usuario no registrado	5.4.3 Capa de presentación
Fig (53) Ejemplo registro de dominio	Anexo 1: Registro del dominio
Fig (54) Menú usuario No registrado	Anexo 3: Manual de usuario.
Fig (55) Mapa	Anexo 3: Manual de usuario.
Fig (56) Mejores talleres	Anexo 3: Manual de usuario.
Fig (57) Mejor valoradas	Anexo 3: Manual de usuario.
Fig (58) Pantalla inicio Usuario Registrado	Anexo 3: Manual de usuario.
Fig (59) Perfil de usuario	Anexo 3: Manual de usuario.
Fig (60) Formulario añadir vehículo de usuario	Anexo 3: Manual de usuario.
Fig (61) Listado vehículos de usuario	Anexo 3: Manual de usuario.
Fig (62) Sección mis amigos	Anexo 3: Manual de usuario.
Fig (63) Enviar mensaje	Anexo 3: Manual de usuario.
Fig (64) Mensajes recibidos/enviados	Anexo 3: Manual de usuario.
Fig (65) Formulario para insertar taller	Anexo 3: Manual de usuario.
Fig (66) Listado de talleres dados de alta por el usuario	Anexo 3: Manual de usuario.
Fig (67) Modificar datos del taller	Anexo 3: Manual de usuario.
Fig (68) Formulario para insertar una reparación	Anexo 3: Manual de usuario.

1 Introducción

Este documento describe el trabajo realizado en el proyecto: “RED SOCIAL DE REPARACIONES DE COCHES”. A fin de aprobar el proyecto final de carrera para la obtención del título de Ingeniero Técnico en Informática de Gestión, expedido por la Universidad Politécnica de Valencia.

El objetivo principal de este proyecto es plasmar una idea, en este caso una red social sobre reparaciones, haciendo uso de los conocimientos adquiridos durante la carrera, así como cómo aprender sobre el uso y manejo de las tecnologías/aplicaciones necesarias para llevar a cabo el proyecto.

Con la aparición de la web 2.0 **(1)** hemos podido ver en relativamente poco tiempo, como las redes sociales han pasado de ser un fenómeno apenas conocido, a convertirse en un elemento importante en la vida de muchas personas.

Las Redes Sociales **(2)** han revolucionado la forma en la que vivimos, en la que trabajamos, la forma de hacer negocios, de comunicarnos, de vivir el ocio y ha supuesto un cambio en la forma en la que los alumnos pueden aprender.

Cuando mencionamos el término Web 2.0 nos referimos a una serie de aplicaciones y páginas de Internet que proporcionan servicios interactivos en red dando al usuario el control de sus datos y la posibilidad de compartirlos con muchísimos otros usuarios.

Así, podemos entender como 2.0 "todas aquellas utilidades y servicios de Internet que se sustentan en una base de datos, la cual puede ser modificada por los usuarios del servicio, ya sea en su contenido (añadiendo, cambiando o borrando información o asociando datos a la información existente), bien en la forma de presentarlos, o en contenido y forma simultáneamente."

1.1 Objetivos

En este proyecto se pretende diseñar y desarrollar una red social de reparaciones de coches. Las redes sociales han proliferado en los últimos años y permiten compartir la información de diferentes usuarios y de sus mismos gustos/aficiones entre ellos, en este caso información de los usuarios y de los distintos coches que han reparado.

Esta red social tiene el fin de que tanto los usuarios como los responsables del sitio puedan añadir y actualizar información cómodamente y en cualquier momento de una forma automatizada.

2 Estado del arte

En mi búsqueda de Webs de referencia que cumpliera con las funcionalidades que buscaba, si bien es cierto hay web/foros dedicados al automóvil, véase el conocidísimo forocoches.com, páginas para compartir/alquilar vehículos o páginas de rutas, pero apenas hay redes sociales dedicados a la reparación y valoración por parte de usuarios.

Ahora expondré brevemente 3 páginas de temática similar o que puedan haber servido de referencia en el diseño de la web de este proyecto.

2.1 driverside

Driverside **(3)** es un portal web dedicado a nuestro vehículo donde conseguir información de talleres y reparaciones, apartado para tasar el precio de nuestro coche, piezas y accesorios, análisis y reportajes de coches, todo esto según el año, marca y modelo que seleccionemos. La ventaja de este servicio es poder conseguir un presupuesto para determinadas reparaciones sin coste alguno o tasar nuestro modelo actual, haciendo un baremo de otras ventas de dicho modelo en función de estado y km.

¿Cuál es la pega? Que es una web íntegramente en inglés y solo disponible para residentes en USA, puesto que nos pide a parte del año, marca y modelo el código zip (equivalente al cp), en el cual se basa para buscar talleres próximos, siendo válido únicamente en dicho continente.



Fig (1) driverside.com

2.2 foodspotting

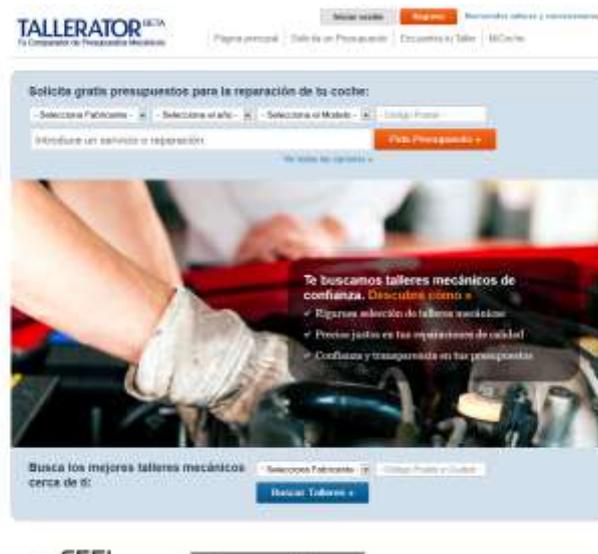
He tomado como referencia foodspotting.com (4) que aunque su temática no tenga nada que ver con el ámbito que ocupa mi proyecto, trata sobre platos de cocina, tiene un diseño simple y sencillo de usar, justo lo que busco para mi red social. Por lo que aprovecharemos la visión general de un mapa para localizar los talleres, y la sencillez para la distribución del sitio web.



Fig (2) foodspotting.com

2.3 tallerator

Tallerator.es (5) es un marketplace web y móvil para conseguir mantenimientos y reparaciones económicas, un comparador de presupuestos mecánicos donde puedes ahorrar en las reparaciones y mantenimientos de tu coche. Su funcionamiento es muy sencillo, cuando necesitas una reparación o mantenimiento de tu coche, solicitas a través de Tallerator.es el servicio que necesitas y los talleres reciben tu solicitud, en muy poco tiempo tendrás en tu panel de control una buena cantidad de presupuestos donde poder elegir, cuando selecciones uno de ellos podrás ver los datos de contacto del taller que has seleccionado e ir al taller sabiendo cuanto te costará y ahorrándote dinero, y lo mejor, a ti no te cuesta nada.



Fig(3) tallerator.es

3 Especificación

3.1 Introducción

En el siguiente apartado vamos a describir el funcionamiento y la implementación de una red social sobre talleres y mecánica en general.

3.2 Descripción General del Sistema

El proyecto a desarrollar consistirá en un sistema de información de reparaciones y talleres donde han sido efectuadas. Para ello, el sistema deberá mantener tanto información sobre talleres, reparaciones y las reparaciones/mantenimientos en dichos talleres. Este sistema permite la recomendación y la votación online de talleres en distintas ciudades, aprovechando información sobre preferencias, gustos y valoraciones de tus amistades y usuarios con características de vehículos parecidas.

Al sistema se podrán conectar tanto usuarios anónimos como registrados. Los anónimos sólo podrán visualizar información general sobre talleres y puntuaciones de los talleres, pero no tendrán acceso a definir relaciones de amistad ni a comentar reparaciones. Por su parte, los registrados podrán comentar y valorar las reparaciones y/o mantenimientos de los vehículos en los talleres, así como visualizar cualquier comentario y las valoraciones de las relaciones de amistad u otros usuarios, también podrán gestionar relaciones de amistad con otros usuarios registrados, así como dar de alta talleres nuevos, vehículos o reparaciones.

Además, existirá el perfil Gestor del portal, cuyo objetivo es obtener informes de los comentarios diarios, las reparaciones realizadas y se encargará de mantener el catálogo de talleres actualizado.

El sistema debe ofrecer la posibilidad de recomendar talleres a los usuarios registrados.

3.3 Requisitos de Gestión de Información

El sistema deberá almacenar información sobre talleres y reparaciones, con el objetivo de realizar valoraciones o ranking de talleres. En concreto, se podrán valorar reparaciones de TALLERES, y en base a una media de estas reparaciones se asignará una puntuación al taller correspondiente. Cada reparación tendrá un precio, descripción de la reparación o mantenimiento, información de la duración

aproximada, etc. Además, las reparaciones estarán etiquetadas en TIPOS (ej. Mantenimiento, Cambio de aceite, Cambio de ruedas, etc.), de manera que podrá pertenecer a uno o varios tipos. El sistema mantendrá información sobre los USUARIOS REGISTRADOS, que contendrá información identificativa, y las valoraciones/comentarios emitidos. De cada usuario registrado el sistema debe mantener el histórico de valoraciones/comentarios, el PERFIL DE PREFERENCIAS o gustos (vehículos, zonas, marcas), y un listado de usuarios AMIGOS, y un listado de PETICIONES DE AMISTAD pendientes de comprobar.

3.4 Requisitos Funcionales

A efectos del caso de estudio, se pedirá que se implemente al menos la siguiente funcionalidad:

- Mapa mundi con la localización de los talleres (presentes en la BD) desde donde acceder a la web/ficha del taller en cuestión.
- Registrar un usuario.
- Petición de amistad.
- Enviar mensaje privado a otro usuario/amigo.
- Añadir comentario a la reparación.
- Valorar (puntuar) reparación.
- Dar de alta un taller.
- Dar de alta una reparación.
- Dar de alta un/varios vehiculo/s del usuario.
- Dar de alta tipos de reparaciones (Gestor del portal).
- Añadir marcas y modelos de vehículos (Gestor del portal).

3.5 Requisitos de Interfaz de Usuario

Se debe desarrollar una interfaz web que permita que el sistema sea usado por: Anónimos, Usuarios Registrados y Gestores del portal. Cada uno de estos usuarios tiene unos objetivos diferentes con el sistema, que se describen a continuación.

3.5.1 Usuarios Anónimos

Estos usuarios podrán ver información sobre los talleres y la valoración general del mismo (incluyendo promociones disponibles). No podrán comentar, ni ver los comentarios, ni valorar, ni explorar las amistades existentes.

3.5.2 Usuarios Registrados

Estos usuarios podrán consultar el catálogo de talleres buscando directamente por zonas, o por tipos de reparaciones. En cualquier momento debe poder comentar una reparación, bien porque han realizado dicha reparación en ese taller, o bien por las recomendaciones de amigos. También podrán dar de alta talleres y/o reparaciones si no existen actualmente. En cualquier momento podrán acceder a su histórico de comentarios (para ver en qué talleres y en qué reparaciones ha comentado/creado con anterioridad) o a su perfil de preferencias. Además, también podrán acceder al listado de sus amistades, pudiendo, para cada uno de ellos, conocer qué talleres y reparaciones han comentado recientemente, o qué tipos de reparaciones tienen en su perfil. También podrán acceder al listado de peticiones de amistad pendientes, pudiendo aceptarlas o rechazarlas.

3.5.3 Gestores del portal

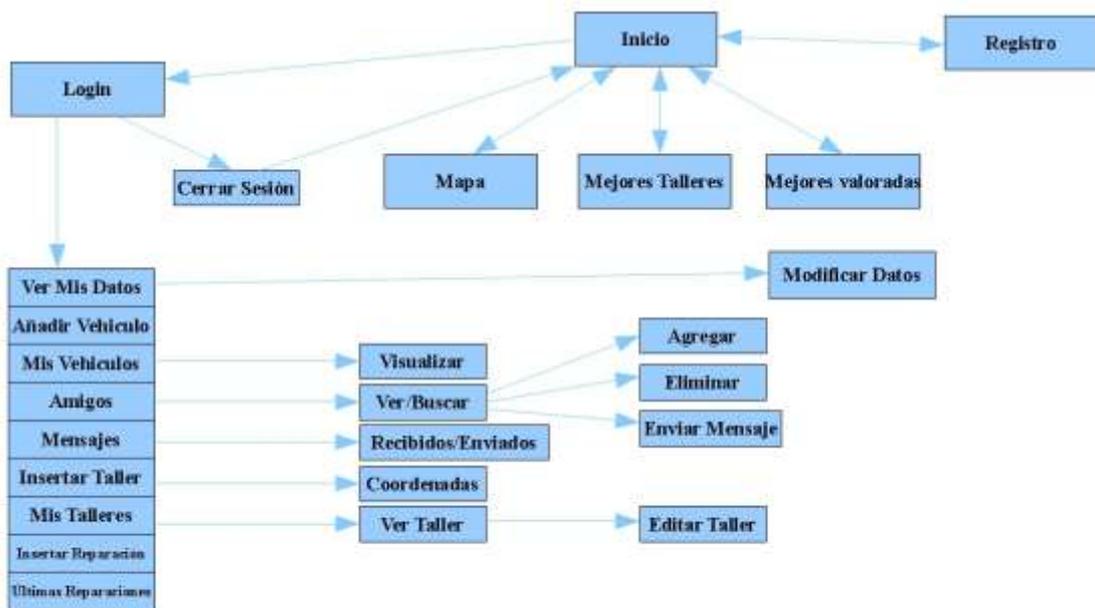
Estos usuarios son los encargados de mantener el catálogo de talleres actualizado. Deberán dar de alta reparaciones/ operaciones de mantenimiento, tipos de vehículos, marcas y modelos.

Podrán consultar los listados actualizados de los comentarios diarios realizados, y el volumen de valoraciones realizadas.

4 Requisitos

4.1 Mapa navegacional

- Desde la página de inicio tenemos acceso a la barra del menú superior con opción de ir a:
 - Inicio, mapas, mejores talleres y mejor valoradas (reparaciones)
- Una vez logeado tenemos acceso al menú privado de usuario.



Desde cualquier página de nuestra web podemos acceder (Inicio)

Una vez has iniciado sesión tenemos acceso al menú privado de usuario

Fig (4) Mapa navegacional

4.2 Descripción de las pantallas

4.2.1 Opciones generales

Estas opciones serán accesibles en cualquier momento, no será necesario que un usuario haya iniciado sesión para poder verlas.

4.2.1.1 Inicio sin una sesión abierta

Esta pantalla nos muestra una breve descripción acerca de las posibilidades de la web y nos permite acceder al resto de opciones y al formulario de inicio de sesión.

Captura de pantalla

The screenshot displays the home page of the 'Reparaweb' website. On the left, there is a login section titled 'Iniciar sesión' with input fields for 'Usuario' and 'Contraseña', and a blue 'Iniciar sesión' button. Below this is a light blue box with the text '¿Aún no formas parte de nuestra comunidad?' and a '¡Regístrate!' link. The top right features a navigation menu with links for 'Inicio', 'Mapa', 'Mejores talleres', and 'Mejor valoradas'. Above the menu are promotional banners: a green circle with 'Hasta un 70% de descuento', a 'VALES DESCUENTO RESTAURANTES*' banner, a 'GROUPON' logo, and a red 'Descuento del día' button. The main content area is titled 'Bienvenidos a Reparaweb' and contains four sections: 'GESTIONA TUS REPARACIONES' (managing repairs), 'BUSCA TALLERES' (finding workshops), 'CREA TU COMUNIDAD' (creating a community), and 'COMUNICATE' (communication).

Fig (5) captura inicio sin una sesión abierta

4.2.1.2 Registro

Esta pantalla posibilita que los usuarios puedan registrarse en nuestra aplicación tras introducir su información personal.

Mockup

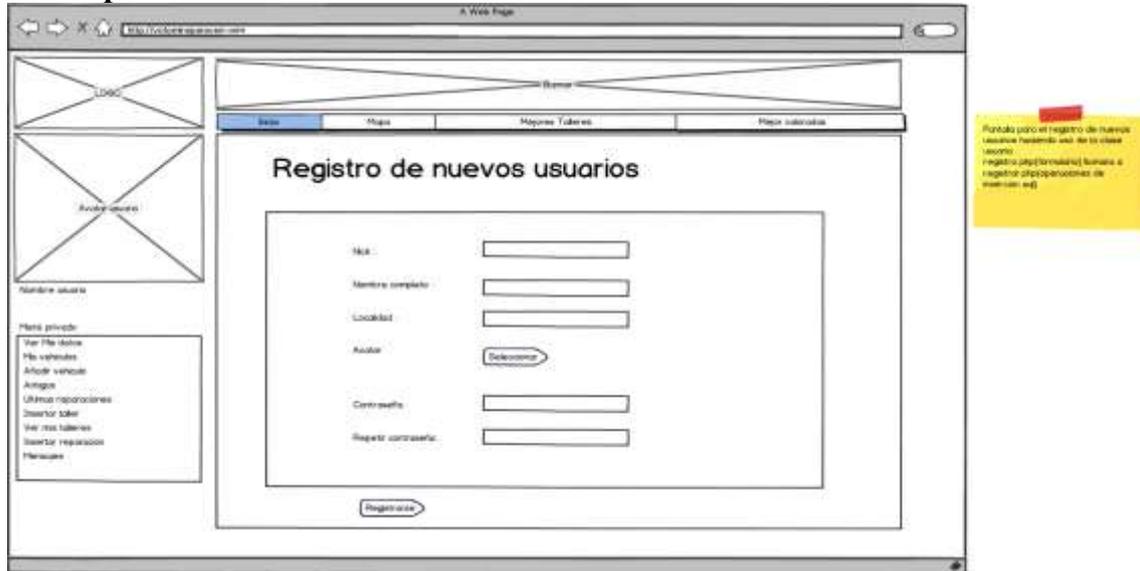


Fig (6) Mockup Registro de usuario

Captura de pantalla



Fig (7) Captura Registro de usuario

4.2.1.3 Mapa

Esta pantalla nos muestra un mapa (6) (7) donde mostramos los talleres introducidos en el sistema. De esta forma posibilitamos la búsqueda geográfica de los talleres que tenemos insertados en la base de datos.

Mockup

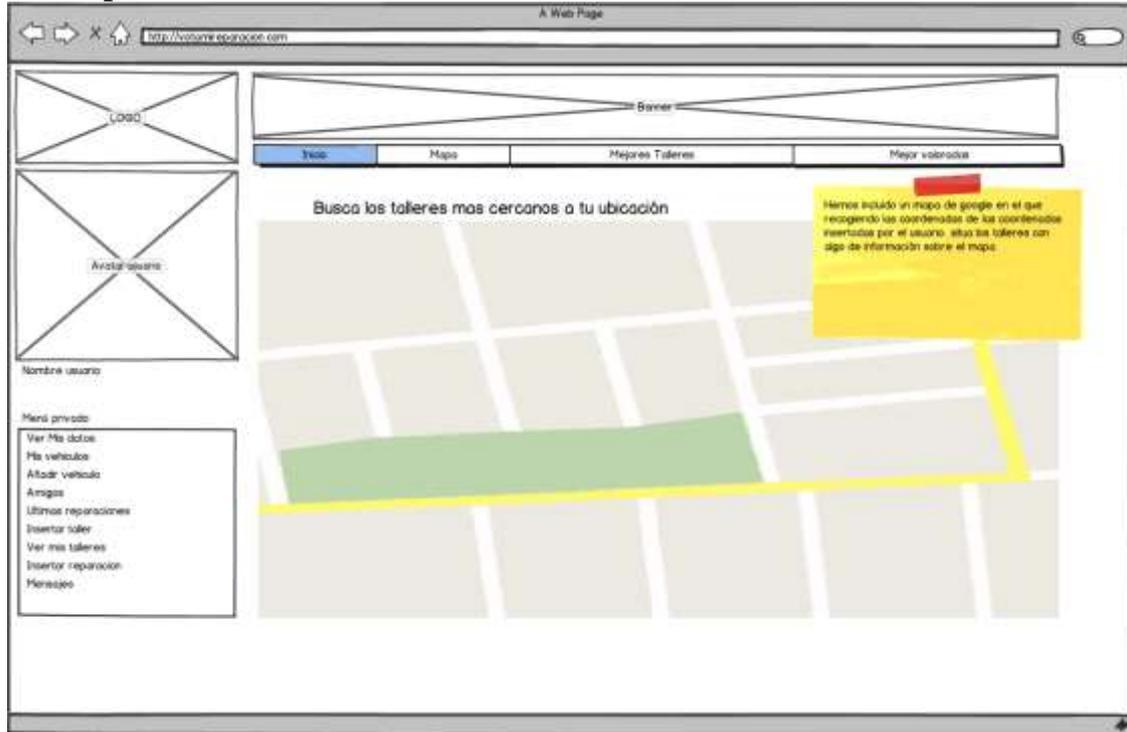


Fig (8) Mockup Mapa

Captura de pantalla



Fig (9) Captura Mapa

4.2.1.4 Mejores talleres

En esta pantalla mostramos los talleres mejor valorados dentro de nuestro sistema y para ello hemos hecho una consulta que calcula la media de las valoraciones de cada reparación agrupadas por nombre de taller.

Mockup



Fig (10) Mockup Mejores talleres

Captura de pantalla



Fig (11) Captura Mejores talleres

4.2.1.5 Mejor valoradas

En esta pantalla mostramos las reparaciones mejor valoradas dentro de nuestro sistema. Este listado nos ofrece la posibilidad de twittear (8) cada reparación así como conocer detalles de cada reparación.

Mockup



Fig (12) Mockup Reparaciones mejor valoradas

Captura de pantalla



Fig (13) Captura Reparaciones mejor valoradas

4.2.2 Opciones privadas

Estas opciones sólo serán accesibles cuando un usuario haya iniciado sesión.

4.2.2.1 Inicio con una sesión abierta

Esta pantalla modifica la información visualizada cuando no hemos iniciado, dejaremos de ver el formulario de acceso para ver un menú con acciones personalizadas para los usuarios.

Mockup

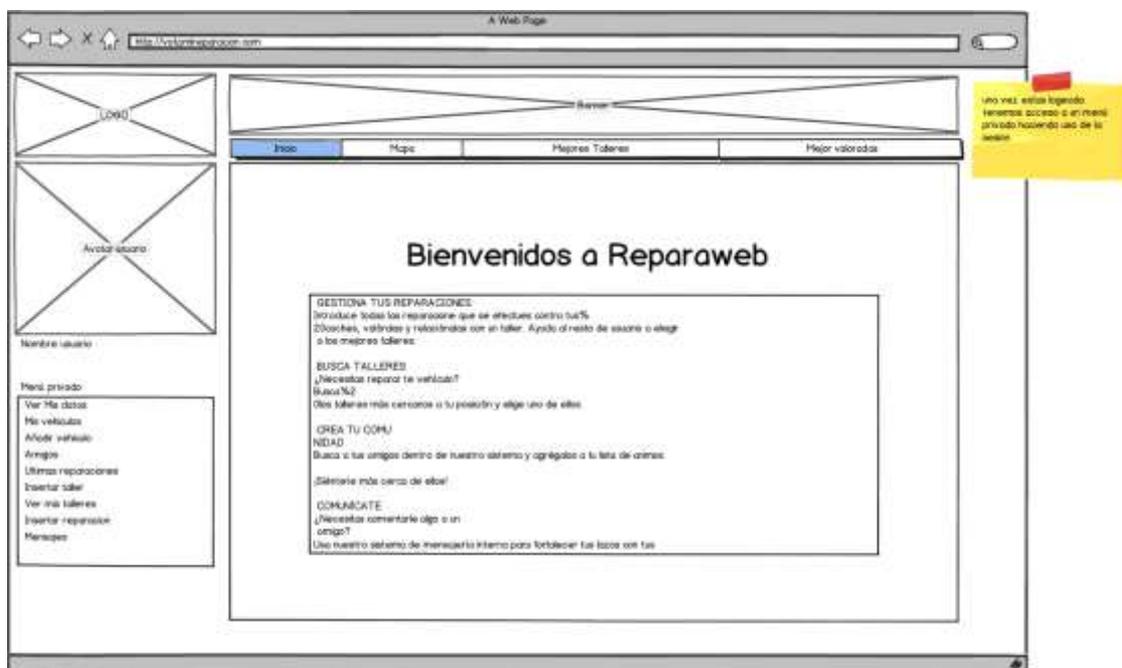


Fig (14) Mockup Inicio con sesión abierta

Captura de pantalla



Fig (15) Captura Inicio con sesión abierta

4.2.2.2 Ver mis datos

Esta pantalla muestra a un usuario registrado la información que se está almacenando sobre él. Ofrece un acceso a la pantalla de edición de usuarios.

Mockup

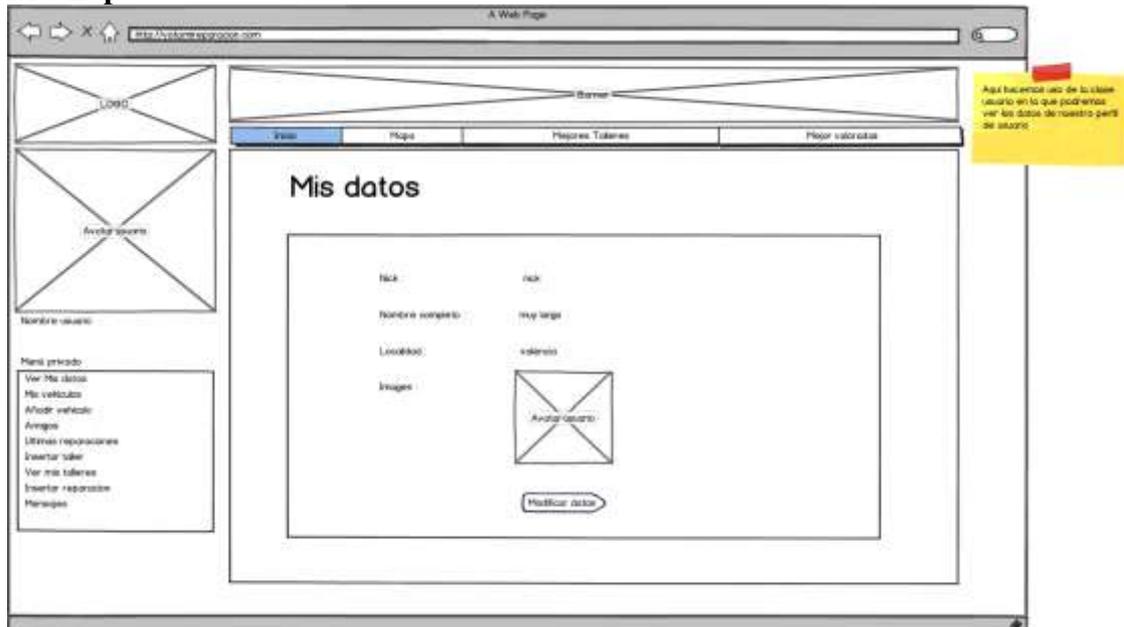


Fig (16) Mockup Ver mis datos

Captura de pantalla



Fig (17) Captura Ver mis datos

4.2.2.3 Modificar mis datos

Esta pantalla ofrece la posibilidad de editar la información que se está almacenando sobre ellos.

Mockup



Fig (18) Mockup Actualizar mis datos

Captura de pantalla



Fig (19) Captura Actualizar mis datos

4.2.2.4 Añadir vehículo propio

Desde esta pantalla podremos ir introduciendo los distintos vehículos que vaya adquiriendo un usuario registrado.

Mockup



Fig (20) Mockup Añadir vehículo de usuario

Captura de pantalla



Fig (21) Captura Añadir vehículo de usuario

4.2.2.5 Mis vehículos

Esta pantalla ofrece a los usuarios registrados un listado con todos los vehículos introducidos por el. También ofrece la posibilidad de acceder a la pantalla de visualización completa de la información introducida.

Mockup

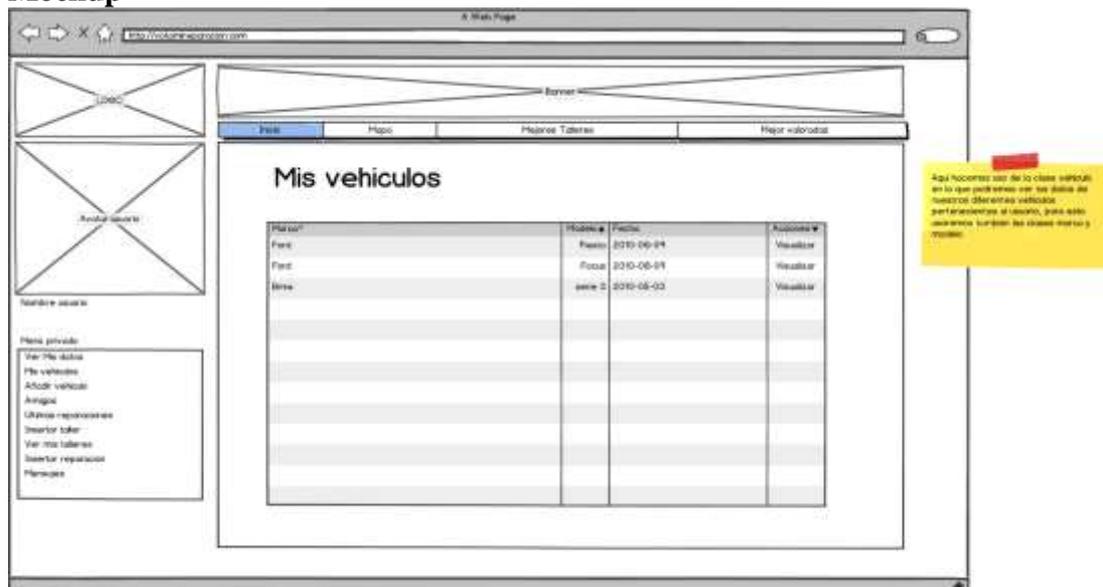


Fig (22) Mockup Mis vehículos

Captura de pantalla



Fig (23) Captura Mis vehículos

4.2.2.6 Visualizar vehículos

Desde esta pantalla podremos acceder a la visualización completa de la información introducida del vehículo.

Mockup



Fig (24) Mockup Visualizar vehiculo

Captura de pantalla



Fig (25) Captura Visualizar vehículo

4.2.2.7 Amigos

Desde esta pantalla podremos acceder al listado completo de amigos del usuario. Desde ella daremos acceso a la búsqueda de nuevos amigos, ofreceremos la posibilidad de eliminarlos y/o de enviarles un mensaje.

Mockup

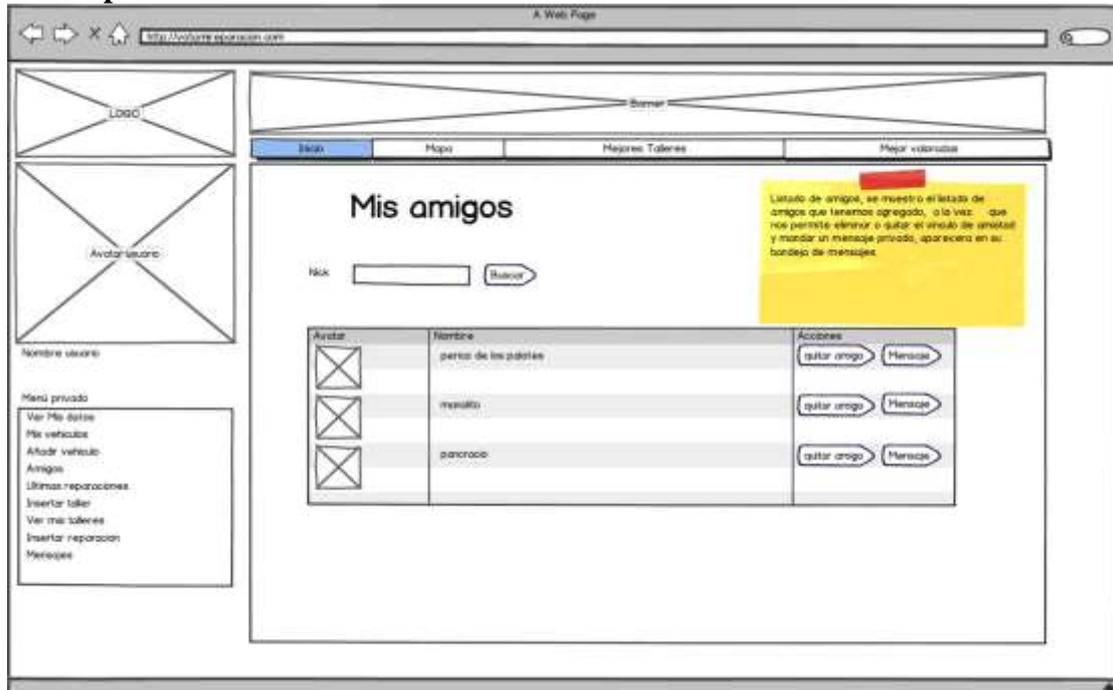


Fig (26) Mockup Amigos

Captura de pantalla



Fig (27) Captura Amigos

4.2.2.8 Enviar mensaje

Desde esta pantalla podremos ponernos en contacto con otros amigos mediante mensajes de texto.

Captura de pantalla

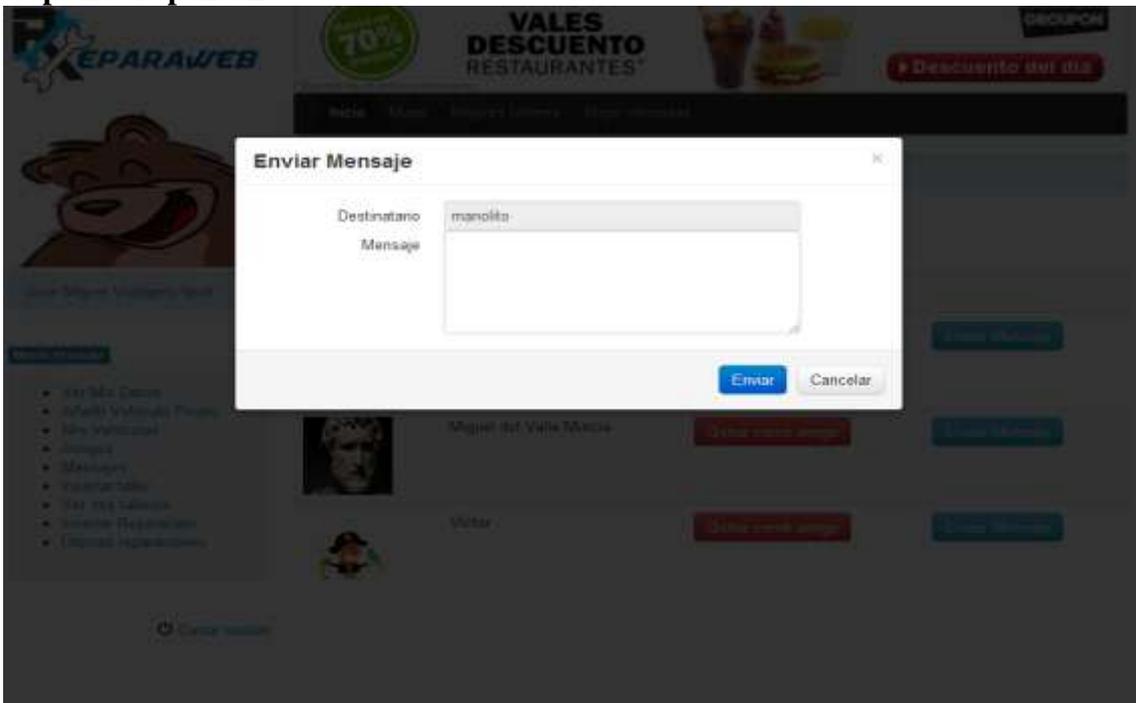


Fig (28) Captura Enviar mensaje

4.2.2.9 Buscar amigos

Accederemos a esta pantalla desde la pantalla del listado de amigos. Mostrará los usuarios existentes en el sistema que cuadren con el filtro especificado.

Mockup



Fig (29) Mockup Buscar amigos

Captura de pantalla

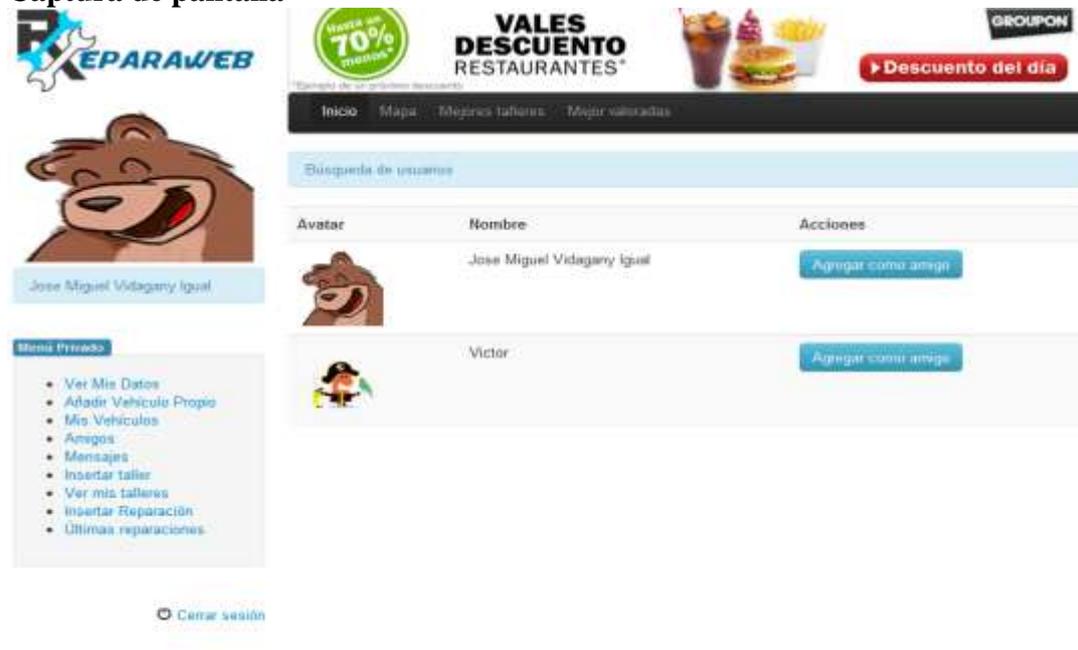


Fig (30) Captura Buscar amigos

4.2.2.10 Mensajes enviados

Desde esta pantalla visualizaremos un listado con los distintos mensajes enviados desde el sistema.

Mockup

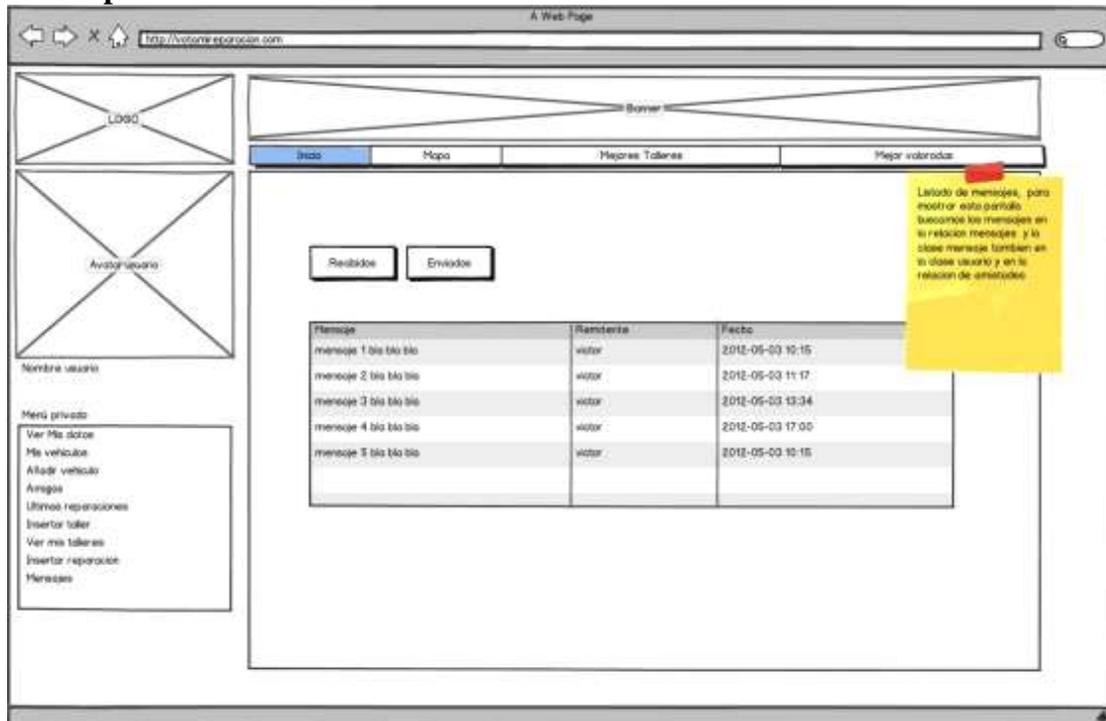


Fig (31) Mockup Mensajes enviados

Captura de pantalla



Fig (32) Captura Mensajes enviados

4.2.2.11 Mensajes recibidos

Desde esta pantalla visualizaremos un listado con los distintos mensajes recibidos desde el sistema.

Mockup

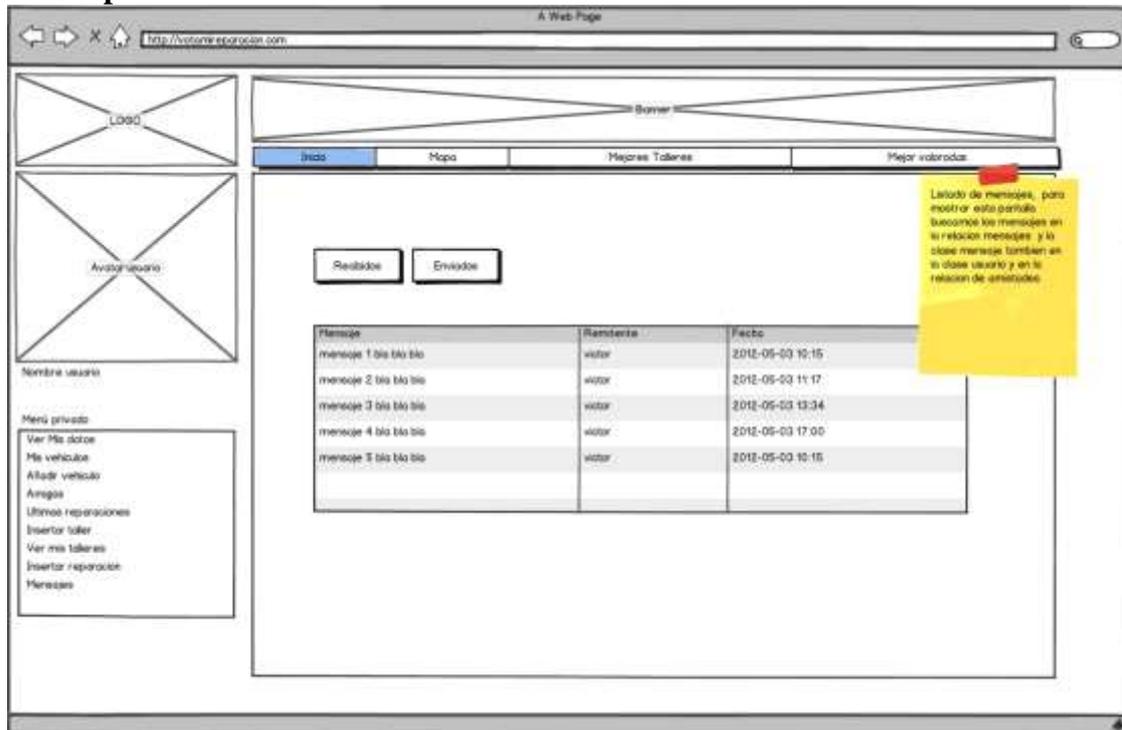


Fig (33) Mockup Mensajes recibidos

Captura de pantalla



Fig (34) Captura Mensajes recibidos

4.2.2.12 Insertar taller

Desde esta pantalla permitiremos la inserción de talleres por parte de los usuarios. Desde esta pantalla ofreceremos acceso a una web basada en Google Maps que nos facilitará la labor de agregar la latitud y la longitud en la que se encuentra situado un mapa.

Mockup



Fig (35) Mockup Insertar taller

Captura de pantalla



Fig (36) Captura Insertar taller

4.2.2.13 Ver mis talleres

Desde esta pantalla permitiremos visualizar el listado de talleres introducidos por el usuario, y de aquí acceder a la visualización de alguno de estos talleres.

Mockup

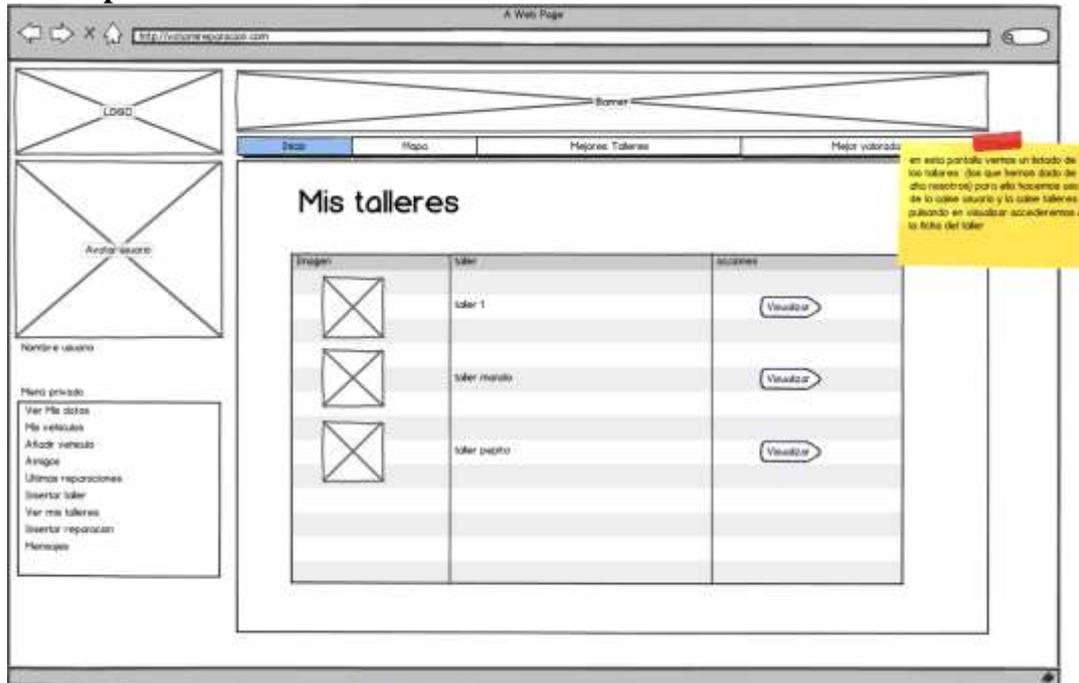


Fig (37) Mockup Ver listado de mis talleres

Captura de pantalla



Fig (38) Captura Ver listado de mis talleres

4.2.2.14 Visualizar taller

Desde esta pantalla veremos los datos de un taller en concreto introducidos por el usuario.

Mockup

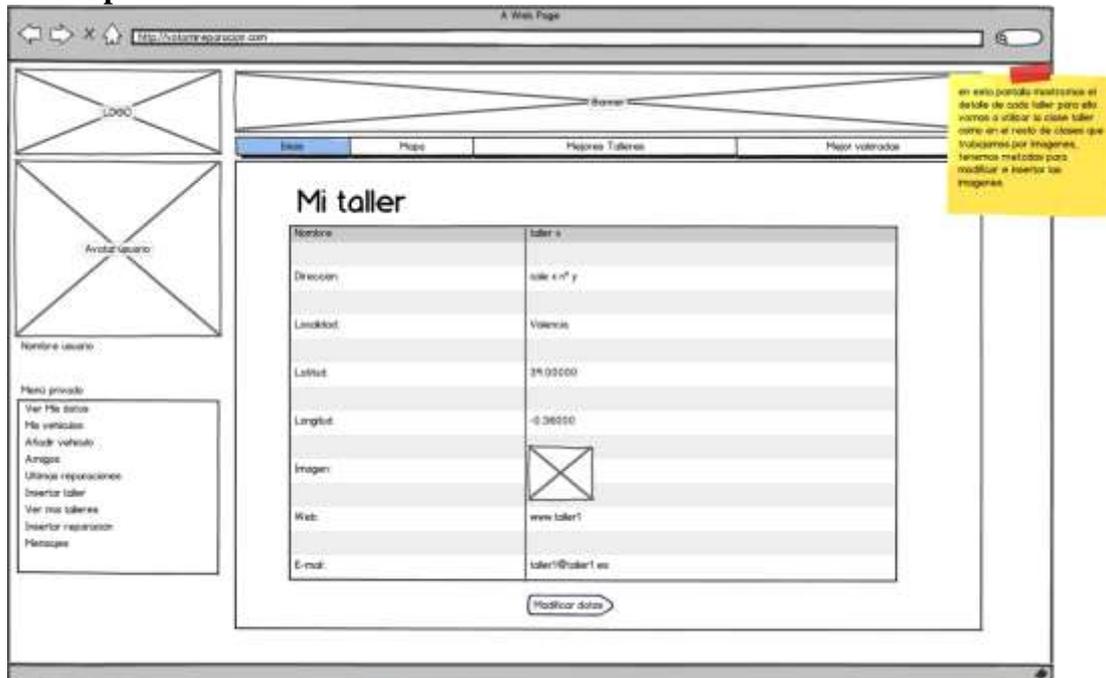


Fig (39) Mockup Visualizar datos taller

Captura de pantalla



Fig (40) Captura Visualizar datos taller

4.2.2.15 Modificar datos del taller

Desde esta pantalla permitiremos modificar los datos del taller.

Mockup



Fig (41) Mockup Modificar datos del taller

Captura de pantalla

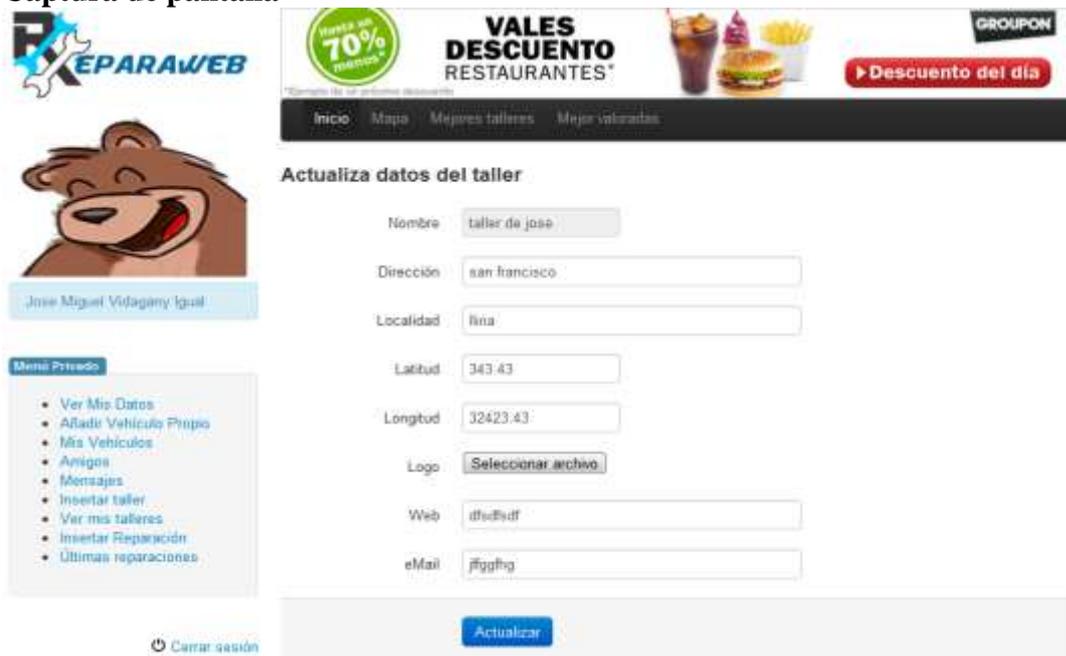


Fig (42) Captura Modificar datos del taller

4.2.2.16 Insertar reparación

Desde esta pantalla un usuario podrá ir añadiendo las distintas reparaciones que vaya realizando contra sus vehículos. Sobre las mismas podrá introducir una valoración.

Mockup



Fig (43) Mockup Insertar reparación

Captura de pantalla



Fig (44) Captura Insertar reparación

4.2.2.17 Últimas reparaciones

Esta pantalla nos ofrecerá un listado con las últimas reparaciones añadidas al sistema por el resto de miembros de la comunidad. Este listado nos ofrece la posibilidad de twittear cada reparación así como conocer detalles de cada reparación.

Mockup



Fig (45) Mockup Últimas reparaciones

Captura de pantalla

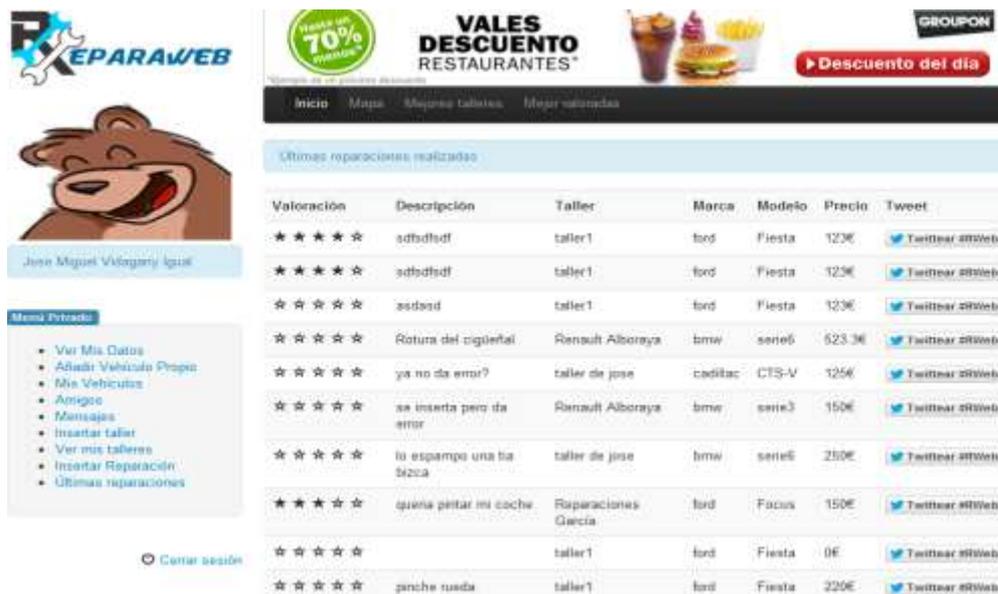


Fig (46) Captura Últimas reparaciones

5 Implementación

5.1 Tecnologías y herramientas utilizadas

5.1.1 Lenguajes, BD, servidor,css

Lenguaje de programación:

- [PHP](#): Como lenguaje de programación **(9)** en el servidor.

Es un lenguaje de programación interpretado o framework para HTML, diseñado originalmente para la creación de aplicaciones para servidores, o creación páginas web dinámicas. PHP es un acrónimo recursivo que significa "PHP Hypertext Preprocessor" (inicialmente PHP Tools, o, Personal Home Page Tools).

- [Javascript \(jQuery\)](#): Como lenguaje de programación **(10)** en el cliente.

Consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. Su característica principal es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

Base de datos:

- [MySQL](#): Es un sistema de gestión de bases de datos relacional (SGDBR), multihilo y multiusuario.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

- [phpMyAdmin \(3.5.1\)](#): Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet.

Servidor:

- [Apache Http Server Project](#): Como servidor web que aloja nuestra web.

El servidor Apache es un software (libre) servidor http de código abierto multiplataforma que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

Css:

- [Bootstrap](#): Como framework CSS (11).

Bootstrap, un framework html/css de Twitter para desarrolladores, no contiene flash, javascript ni imágenes innecesarias, solo CSS.

5.1.2 Herramientas de desarrollo

Diagramas ER:

- [MySQL Workbench](#): Para la realización del diagrama ER y las modificaciones en base de datos.

5.2 Estructura de la aplicación

Paso a describir la estructura de los archivos que componen la aplicación para el usuario dando una breve descripción de su funcionamiento.

- Index.php
 - datos.php
 - clases.php
 - (usuario, mensaje, taller, vehículo, reparación, marca, modelo)
 - multi.php
 - barra.php

Este esquema nos indica que *index.php*, además de su código, contiene otros archivos que pueden contener a su vez otros.

index.php contiene los archivos *datos.php*, *clases.php* (*clases.php* a su vez contiene a las clases *usuario.php*, *mensaje.php*, *taller.php*, *vehiculo.php*, *reparación.php*, *marca.php* y *modelo.php*), también contiene *barra.php* y *multi.php* (encargado de cargar los ficheros de traducción *lang_es.php* y *lang_en.php*) según el idioma del usuario.

Notas:

Siempre navegaremos sobre la página de *index.php* que mantendrá la *barra.php* y todas las paginas por las que nos desplazemos se cargaran en un marco central.

Todos los ficheros *.php* de la aplicación contienen *multi.php* para poder adaptar el idioma según el usuario que se conecte.

Archivos comunes:

Index.php

Página de inicio de nuestra aplicación, dentro de esta página se cargaran el resto de páginas por las que nos desplazemos en un marco central.

barra.php

Es el menú de la parte superior, con el que el usuario se puede desplazar a las diferentes secciones de la web: inicio, mapa, mejores talleres y reparaciones mejor valoradas.

multi.php

Archivo que nos carga el idioma según las preferencias del perfil de usuario (opción modificable que por defecto está en inglés)

clases.php

Este archivo nos carga los *.php* de todas las clases del sistema (*usuario.php*, *mensaje.php*, *taller.php*, *vehiculo.php*, *reparación.php*, *marca.php* y *modelo.php*)

datos.php

Carga los datos necesarios para realizar la conexión con la BD

Archivos de navegación:

Inicio.php

Página de bienvenida y de presentación de cara a los nuevos usuarios.

registro.php

Formulario de registro para nuevos usuarios.

registrar.php

Funciones necesarias para llevar a cabo el registro y posterior modificación de usuarios y almacenarlos en la BD.

ver_misdatos.php

Página a modo de tabla donde ver los datos de usuario, con opción a modificarlos en caso de crearlo conveniente.

misdatos.php

Formulario con opción a modificar los datos de usuario.

amigos.php

Se muestra una página con nuestra lista de amigos y sobre ellos la opción de eliminarlos o mandarles un mensaje.

todos.php

Para realizar búsquedas de usuarios, en caso de buscar y querer agregar a algún amigo.

Enviarsolicitud
Contiene la funcionalidad para añadir a nuestra lista de amigos el amigo solicitado mediante una llamada a la clase usuario.
quitaramigo.php
Contiene la funcionalidad para eliminar de nuestra lista de amigos el amigo solicitado mediante una llamada a la clase usuario
mensajes.php
Desde aquí el usuario puede consultar su lista de mensajes recibidos/enviados.
nuevomensaje.php
Para redactar y enviar un mensaje privado de texto a uno de nuestros amigos.
mapa.php
Muestra un mapa mundi (en el que podemos hacer zoom o movernos libremente) para buscar en el talleres próximos existentes en nuestra BD
login.php (logingoogle.php, logintwitter.php, loginyahoo.php)
Funcionalidad para que un usuario registrado en nuestra web inicie sesión, así como para usuarios no registrados que lo deseen puedan iniciar sesión desde otros servicios conocidos como google, twitter y yahoo.
logout.php
Desde aquí cerramos nuestra sesión de usuario.

sesioniniciada.php
Muestra un menú para usuarios registrados con opciones a: (ver mis datos, añadir vehículo propio, mis vehículos, amigos, mensajes, insertar taller, ver mis talleres, insertar reparación, ultimas reparaciones)
registro_taller.php
Formulario para dar de alta un nuevo taller.
registrar_taller.php
Funciones necesarias para dar de alta y modificar los datos del nuevo taller en la BD.
mejorestalleres.php
Tabla resumen de los mejores talleres, la valoración de los talleres va en función de las valoraciones de las reparaciones en dichos talleres, siendo la valoración del taller una media de las valoraciones de las reparaciones en dicho taller.
mistalleres.php
Página para la visualización de los distintos talleres que podamos haber dado de alta
modificartaller.php
Formulario con opción a modificar los datos del taller seleccionado.
ver_taller.php
Página para ver la ficha con los datos del taller seleccionado.
registro_vehiculo.php
Formulario para dar de alta un vehículo de usuario.

registrar_vehiculo.php
Funciones necesarias para dar de alta un vehículo en la BD.
mivehiculo.php
Página donde se muestra una lista de los vehículos de usuario dados de alta en la BD.
mostrar_vehiculo.php
Página para visualizar los datos del vehículo seleccionado.
registro_reparacion.php
Formulario para dar de alta una nueva reparación, donde seleccionaremos el vehículo de usuario (dado de alta previamente), el taller donde lo hemos reparado, tipo de reparación realizada y una valoración personal.
registrar_reparacion.php
Funciones necesarias para dar de alta una reparación en la BD.
mejorvaloradas.php
Tabla con las reparaciones mejor valoradas de nuestra BD.
ultimasreparaciones.php
Tabla con las ultimas reparaciones realizadas, independientemente de su valoración, de cara al usuario para que pueda hacerse una idea de las ultimas reparaciones más comunes del resto de usuarios.
votar.php
Funcionalidad para almacenar en la BD las distintas valoraciones de los usuarios referentes a reparaciones.

5.3 Arquitectura

En este apartado veremos el tipo de arquitectura utilizada en el diseño además de las capas o niveles que aparecen (interfaz de usuario o presentación, lógica de negocio, persistencia o de datos).

La arquitectura es la de cliente-servidor ampliamente usada en aplicaciones Web.

Esta arquitectura es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios (servidores) , y los (clientes). Un (cliente) realiza peticiones a otro programa, (servidor), que le da respuesta.

La capacidad de proceso está repartida entre los clientes y los servidores, son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que hace más fácil y claro el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus funcionalidades cambien de unos servicios a otros, la arquitectura básica es la misma.

La *arquitectura cliente-servidor* sustituye a la *arquitectura monolítica* en la que no hay distribución, tanto a nivel físico como a nivel lógico.

Una disposición muy común son los *sistemas multicapa* en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

Características Cliente:

- Es quien solicita las peticiones, tienen un papel activo en la comunicación (dispositivo maestro o amo).
- Espera y recibe las respuestas del servidor.
- Normalmente, puede conectarse a varios servidores a la vez.
- Casi siempre interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

características Servidor:

- Esperan a que lleguen las solicitudes de los clientes, tienen un papel pasivo en la comunicación (dispositivo esclavo).
- Tras recibir una solicitud, la procesan y luego envían la respuesta al cliente.
- Normalmente, aceptan conexiones desde un gran número de clientes (a veces el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

Ventajas

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, y sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente).
- Existen tecnologías diseñadas para asegurar la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de uso.

Desventajas

- La congestión del tráfico ha sido siempre un problema en esta arquitectura. Cuando muchos clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). Al contrario, en las redes P2P como cada nodo en la red hace también de servidor, cuantos más nodos hay, mejor es el ancho de banda que se tiene.
- La arquitectura Cliente-Servidor no tiene la robustez de una red P2P. Cuando un servidor está *caído*, las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes P2P, los recursos están generalmente distribuidos en varios nodos de la red. Aunque algunos salgan o abandonen la descarga; otros pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red.

- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el coste.

Algunas redes disponen de tres tipos de nodos:

- Clientes que interactúan con los usuarios finales.
- Servidores de aplicación que procesan los datos para los clientes.
- Servidores de la base de datos que almacenan los datos para los servidores de aplicación.

Esta configuración se llama una **arquitectura de tres-capas** (como ya habíamos dicho usamos la separación en capas [presentación,lógica, datos]).

Este punto está basado **(12)**

5.4 Diseño

5.4.1 Diagrama de clases

Los diagramas de clases son un tipo de diagramas estáticos que describen la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

Utilizados durante el proceso de diseño de los sistemas informáticos, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

Para la realización de este diagrama de clases se ha utilizado la herramienta [MOSKitt](#). (13)

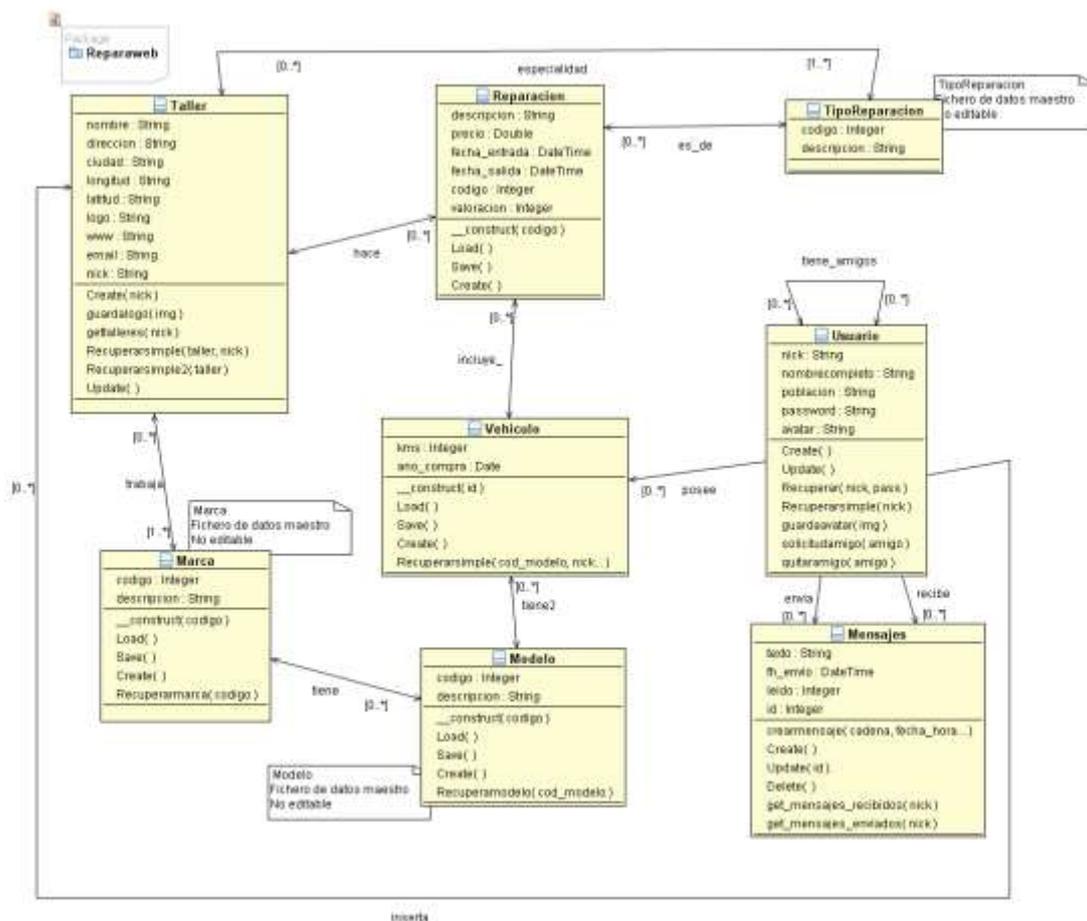


FIG (47) Diagrama de clases

5.4.1.1 Descripción de las clases

- **Usuario**

La clase usuario es la más importante de todas pues todo gira en torno al usuario, que será el que podrá usar toda la funcionalidad de la web en caso de estar registrado.

- **Mensajes**

La clase mensaje representa a los mensajes que podrá enviar/recibir un usuario registrado de otro usuario.

- **Vehículo**

La clase vehículo representa a un vehículo del usuario con referencias a la marca y modelo.

- **Marca**

La clase marca representa la marcas disponibles para los distintos vehículos.

- **Modelo**

La clase modelo representa los distintos modelos que tiene cada marca en concreto.

- **Taller**

La clase taller representa a los distintos talleres que se puedan dar de alta en el sistema por un usuario o administrador, estos talleres serán usados para consulta de sus datos o seleccionado cuando se dé el alta de una nueva reparación.

- **Reparación**

La clase reparación representa a las distintas reparaciones que pueda efectuar un usuario sobre alguno de sus vehículos.

- **Tipo Reparación**

Esta clase representa el listado de los posibles tipos de reparaciones disponibles cuando un usuario da de alta una reparación para alguno de sus vehículos, esta lista es fija y solo es modificable por un administrador de la web.

5.4.2 Diagrama Entidad-Relación

Para la realización de este diagrama Entidad-relación se ha hecho uso de la herramienta [MySQL Workbench](#). (14)

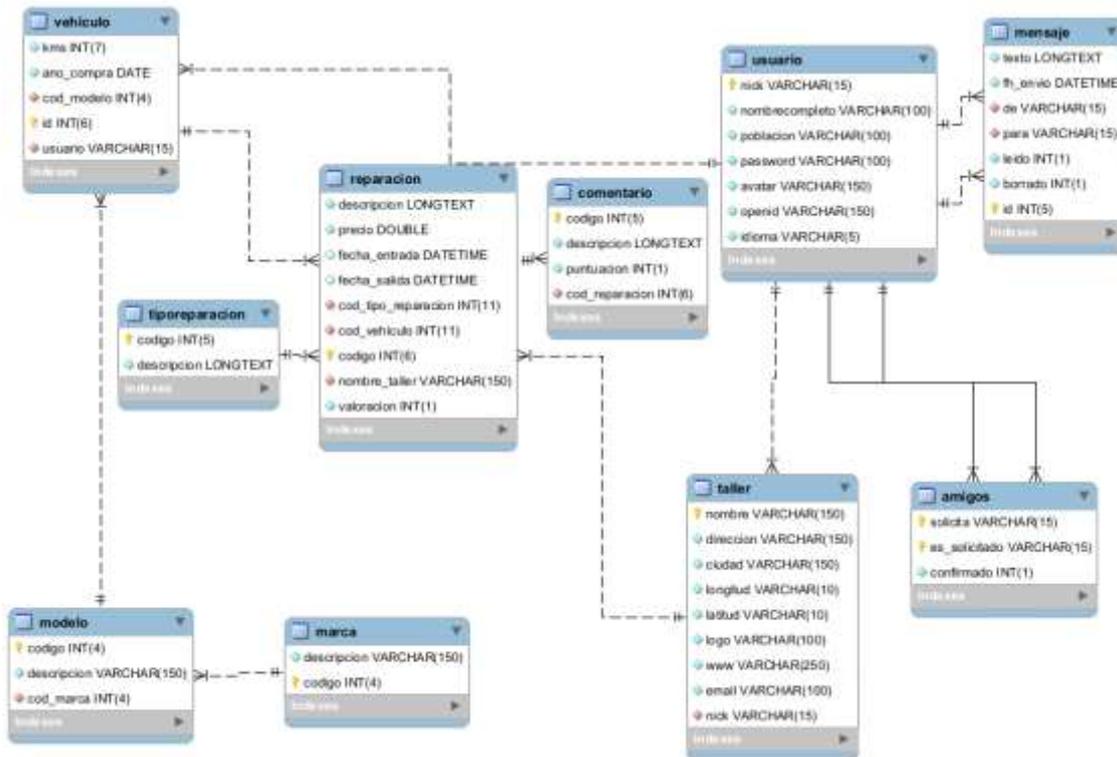


FIG (48) Diagrama Entidad-Relación

5.4.3 Capa de presentación

Para esta capa , la de presentación o interfaz de usuario, se ha usado el framework [Bootstrap](#) de twitter. Conjunto de Css y estilos para hacer más amigable y sencilla la interfaz de usuario.

El formato utilizado para la interfaz de usuario no registrado.



FIG (49) interfaz de usuario no registrado

Su mockup



FIG (50) boceto de la interfaz de usuario no registrado

Formato utilizado en la interfaz de usuario registrado



FIG (51) interfaz de usuario no registrado

Su mockup

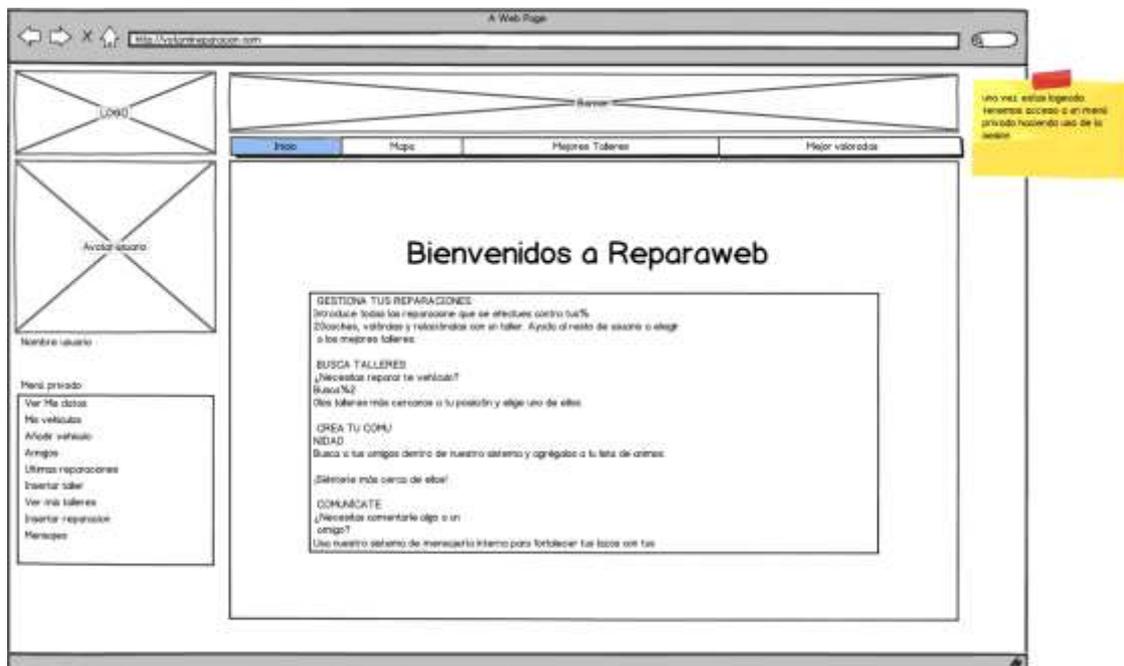


FIG (52) boceto de la interfaz de usuario no registrado

5.4.4 Capa lógica

El nivel lógico está formado por múltiples archivos en su mayoría *.php y algún *.js que implementan la funcionalidad necesaria para el correcto funcionamiento de la aplicación.

Gracias a la independencia entre el nivel de interfaz de usuario y el nivel de persistencia de datos es posible realizar cambios significativos en dicho nivel, por ejemplo, cambiar el sistema de gestión de base de datos o incluso sustituir la propia base de datos por otro sistema de almacenamiento sin que esto afectase al nivel de interfaz, sencillamente cambiando la parte de acceso a la base de datos en nuestro fichero *datos.php*.

Este nivel contiene distintos tipos de funciones que se diferencian según su propósito: funciones en php, funciones que son una mezcla de *php* y *html* o acceso a base de datos.

Un ejemplo de funciones php serían los distintos archivos de registrar: registrar.php, registrar_vehiculo.php, registra_taller.php, registrar_reparacion.php , puesto que su código es únicamente php, de cara a programar una funcionalidad.

Por otro lados los formularios son mezcla de php y html por ejemplo (los formularios que hacen uso de las funciones php mencionadas arriba): registro.php, registro_vehiculo.php, registro_taller.php, registro_reparación.php.

A continuación voy a adjuntar el código de un archivo íntegramente php y otro de su formulario, mezcla de php y html.

registrar.php y registro.php

Registrar.php

```
<?php
$error="";
if (isset($_POST["enviarregistro"])){
    $usuario = new usuario(1);
    $img=$_FILES["archivo"];
    $nick=$_POST['txtUsuario'];
    $pass=$_POST['txtContraseña'];
    $nombre=$_POST['txtNCompleto'];
    $localidad=$_POST['txtLocalidad'];
    $idioma=$_POST['idioma'];

    $usuario->guardaavatar($img);
    $usuario->setnombrecompleto($nombre);
    $usuario->setnick($nick);
    $usuario->setpoblacion($localidad);
    $usuario->setpassword($pass);
    $usuario->setidioma($idioma);
    if($usuario->Create()){

        include("registroom.php");
    }
    else{
        $error= 3;
        include("registro.php");
    }
}elseif (isset($_POST["enviarmodificaciones"])){
    $usuario = new usuario(1);
    $estaelfichero = $_FILES["archivo"]['name'];
    if ($estaelfichero== "") {
        $img=$_POST['archivoactual'];
        $usuario->setavatar($img);
    }else{
        $img=$_FILES["archivo"];
        //ojo! solo hacemos esto si el fichero es
distinto
        $usuario->guardaavatar($img);
    }
    if (strcmp($_POST['txtUsuario'], $_POST['user_prev'])
!= 0)
    {
        $_SESSION['user_new'] = $_POST['txtUsuario'];
        $nick=$_POST['user_prev'];
    }
    else
    {
```

```
        $nick = $_POST['txtUsuario'];
    }
    $pass=$_POST['txtContrasena'];
    $nombre=$_POST['txtNCompleto'];
    $localidad=$_POST['txtLocalidad'];
    $idioma=$_POST['idioma'];

    $usuario->setnombrecompleto($nombre);
    $usuario->setnick($nick);
    $usuario->setpoblacion($localidad);
    $usuario->setidioma($idioma);

    if(strlen($pass)>1){
        $usuario->setpassword($pass);
    }else{
        //si no queremos cambiar el password
        comprobaremos que al hacer el update no envíe un false
        $usuario->setpassword(false);
    }
    if($usuario->Update())
    {
        include("registrook.php");
        //para que se actualicen los datos
        $_SESSION['usuario']=$usuario->getnick();
        $_SESSION['poblacion']=$usuario->getpoblacion();
        $_SESSION['nombrecompleto']=$usuario-
>getnombrecompleto();
        $_SESSION['avatar']=$usuario->getavatar();
        $_SESSION['idioma']=$usuario->getidioma();
    }
    else{
        $error= 3;
        include("misdatos.php");
    }

}else{
    include("registro.php");
}
?>
```

Registro.php

```
<div>
    <h3>Registro de nuevos usuarios</h3>
    <br/>
    <form class="form-horizontal"
action="index.php?page=registrar" method="post"
enctype="multipart/form-data">
    <? if ($error) {?>
        <div class="alert alert-warning"><? echo
$tipoerror[$error]; ?></div>
    <? } ?>
        <div class="control-group">
            <label class="control-label"
for="txtUsuario">Nick</label>
            <div class="controls">
                <input type="text" class="span2"
id="txtUsuario" name="txtUsuario">
            </div>
        </div>
        <div class="control-group">
            <label class="control-label"
for="txtNCompleto">Nombre Completo</label>
            <div class="controls">
                <input name="txtNCompleto" type="text"
class="span4" id="txtNCompleto" size="50">
            </div>
        </div>
        <div class="control-group">
            <label class="control-label"
for="txtLocalidad">Localidad</label>
            <div class="controls">
                <input type="text" class="span4"
id="txtLocalidad" name="txtLocalidad">
            </div>
        </div>
        <div class="control-group">
            <label class="control-label"
for="archivo">Avatar</label>
            <div class="controls">
                <input name="archivo" type="file"
id="archivo" size="35">
            </div>
        </div>
        <div class="control-group">
            <label class="control-label"
for="txtContrasena">Contrase&ntilde;a</label>
            <div class="controls">
```

```
        <input type="password" class="span2"
id="txtContrasena" name="txtContrasena">
        </div>
    </div>
    <div class="control-group">
        <label class="control-label"
for="txtContrasena">Repetir contrase&ntilde;a</label>
        <div class="controls">
            <input type="password" class="span2"
id="txtContrasena2">
            </div>
        </div>
        <div class="form-actions">
            <input type="submit" class="btn btn-primary"
value="Registrarse" name="enviarregistro"/>
        </div>
    </form>
</div>
```

5.4.5 Capa de persistencia de datos

En esta capa se ha hecho uso de un sistema de gestión de bases de datos relacional (SGDBR) [MySQL](#) administrado vía Web desde [phpMyAdmin](#).

TABLA **usuario** (nick: VARCHAR(15), nombrecompleto: VARCHAR(100), poblacion: VARCHAR(100), password: VARCHAR(100), avatar: VARCHAR(150), openid: VARCHAR(150), idioma: VARCHAR(5))

CP: {nick}

VNN:{nick}

nick: Alias e identificador del usuario.

nombrecompleto: nombre completo del usuario.

poblacion: Poblacion del usuario.

password: Contraseña del usuario.

avatar: Foto del usuario.

openid: Identificador openid del usuario.

idioma: Idioma que el usuario tiene definido en la web.

TABLA **amigos** (solicita: VARCHAR(15), es_solicitado: VARCHAR(15), confirmado: INT(1))

CP: {solicita}

CAJ: {solicita, es_solicitado} -> USUARIO

VNN:{solicita, es_solicitado, confirmado}

solicita: nick identificador del usuario que solicita la amistad.

es_solicitado: nick identificador del usuario que es solicitado la amistad.

confirmado: valor 0 ó 1 para si esta confirmada la amistad o no.

TABLA **tiporeparacion** (codigo: INT(5), descripcion: LONGTEXT)

CP: {codigo}

VNN:{codigo, descripcion}

codigo: Identificador del tipo de reparacion.

descripcion: descripcion del tipo de reparacion.

TABLA **marca** (codigo: INT(4), descripcion: VARCHAR(150))

CP: {codigo}

VNN:{codigo, descripcion}

codigo: Identificador de la marca.

descripcion: descripcion de la marca.

TABLA **modelo** (codigo: INT(4), descripcion: VARCHAR(150), cod_marca: INT(4))

CP: {codigo}

CAJ: {cod_marca} -> MARCA

VNN:{codigo, descripcion, cod_marca}

codigo: Identificador del modelo.

descripcion: descripcion del modelo.

cod_marca:Codigo identificador de la marca del modelo.

TABLA **vehiculo** (id: INT(6), kms: INT(7), ano_compra: DATE, cod_modelo: INT(4), usuario: VARCHAR(15))

CP: {id}

CAJ: {usuario} -> USUARIO

CAJ: {cod_modelo} -> MODELO

VNN:{id, kms, ano_compra, cod_modelo, usuario}

id: Identificador del vehiculo.

kms: Kilometros del vehiculo.

ano_compra: Año de compra del vehículo.

cod_modelo: Código identificador del modelo del vehiculo.

usuario: nick propietario del vehiculo.

TABLA **taller** (nombre: VARCHAR(150), direccion: VARCHAR(150), ciudad: VARCHAR(150), longitud: VARCHAR(10), latitud: VARCHAR(10), logo: VARCHAR(100), www: VARCHAR(250), email: VARCHAR(100), nick: VARCHAR(15))

CP: {nombre}

CAJ: {nick} -> USUARIO

VNN:{nombre, direccion, ciudad, longitud, latitud, logo, www, email, nick}

nombre: Identificador del nombre del taller.

direccion: Dirección del taller.

ciudad: Ciudad del taller.

longitud: Coordenada cartográfica del taller.

latitud: Coordenada cartográfica del taller.

logo: Imagen de marca del taller.

www: Dirección web del taller.

email: Correo electrónico del taller.

nick: Identificador de usuario que registro el taller.

TABLA **reparacion** (descripcion: LONGTEXT, precio: DOUBLE, fecha_entrada: DATETIME, fecha_salida: DATETIME, cod_tipo_reparacion: INT(11), cod_vehiculo: INT (11), codigo: INT(6), nombre_taller: VARCHAR(150), valoracion: INT(1))

CP: {codigo}

CAJ: {cod_tipo_reparacion} -> TIPOREPARACION

CAJ: {cod_vehiculo} -> VEHICULO

CAJ: {nombre_taller} -> TALLER

VNN:{codigo, descripcion, precio, fecha_entrada, fecha_salida, cod_tipo_reparacion, cod_vehiculo, nombre_taller, valoracion}

codigo: Identificador de la reparacion.

descripcion: Descripcion de la reparacion.

precio: Precio de la reparacion.

fecha_entrada: Fecha de entrada del vehiculo al taller.

fecha_salida: Fecha de salida del vehiculo al taller.

cod_tipo_reparacion:Codigo del tipo de reparaci3n.

cod_vehiculo:Codigo del vehiculo reparado.

nombre_taller: Nombre del taller donde se ha hecho esta reparacion.

valoracion: Numero entre 0 y 9 valorando la reparacion.

TABLA **comentario** (codigo: INT(5), descripcion: LONGTEXT, puntuacion: INT(1), cod_reparacion: INT(6))

CP: {codigo}

CAJ: {cod_reparacion} -> REPARACION

VNN: {codigo, descripcion, puntuacion, cod_reparacion}

codigo: Identificador del comentario.

descripcion: Texto del comentario en cuestion.

puntuacion: Valor entre 0 y 9 del comentario.

cod_reparacion: Identificador de la reparacion comentada.

TABLA **mensaje** (texto: LONGTEXT, fh_envio: DATETIME, de: VARCHAR(15), para: VARCHAR(15), leído: INT(1), borrado: INT(1), id: INT(5))

CP: {id}

CAJ: {de} -> USUARIO

CAJ: {para} -> USUARIO

VNN: {texto, fh_envio, de, para, leído, borrado, id}

id: Identificador del mensaje.
texto: Texto del mensaje.
fh_envio: Fecha en la que se ha enviado el mensaje.
de: nick del usuario que ha enviado el mensaje
para: nick del usuario que recibe el mensaje.
leido: Valor 0 o 1, por si el usuario destinatario del mensaje, lo ha leído o no.
borrado: Valor 0 o 1, por si el usuario destinatario del mensaje, lo ha borrado o no.

Voy a adjuntar a modo de ejemplo el código de la clase **usuario.php**

```
<?php
class usuario {

    private $nick;
    private $nombrecompleto;
    private $poblacion;
    private $password;
    private $img;
    private $openid;
    private $idoma;

    //función del constructor que nos crea un objeto de tipo
    usuario
    public function Create() {
        $dblink = null;
        try {
            $dblink =
mysql_connect(DB_HOST,DB_USER,DB_PASS);
            mysql_select_db(DB_BASE,$dblink);
        } catch(Exception $ex) {
            echo "Could not connect to " . DB_HOST
. ":" . DB_BASE . "\n"; echo "Error: " . $ex->message;
            exit;
        }
        $query ="INSERT INTO usuario
(`nick`,`nombrecompleto`,`poblacion`,`password`,`avatar`,`o
penid`,`idioma`) VALUES ('" .
mysql_real_escape_string($this->getnick(),$dblink) . "','"
. mysql_real_escape_string($this-
>getnombrecompleto(),$dblink) . "','" .
```

```

mysql_real_escape_string($this->getpoblacion(), $dblink) .
"', '" . md5(mysql_real_escape_string($this-
>getpassword(), $dblink)) ."', '" .
mysql_real_escape_string($this->getavatar(), $dblink) .
"', '" . mysql_real_escape_string($this-
>getopenid(), $dblink) . "', '" .
mysql_real_escape_string($this->getidioma(), $dblink) .
"'");";

        if (mysql_query($query, $dblink)) {
            if(is_resource($dblink))
mysql_close($dblink);
            return true;
        }else{
            if(is_resource($dblink))
mysql_close($dblink);
            return false;
        }

        if(is_resource($dblink))
mysql_close($dblink);
    }

    //update para actualizar los datos del usuario
    public function Update() {
        $dblink = null;
        try {
            $dblink =
mysql_connect(DB_HOST, DB_USER, DB_PASS);
            mysql_select_db(DB_BASE, $dblink);
        } catch(Exception $ex) {
            echo "Could not connect to " . DB_HOST
. ":" . DB_BASE . "\n"; echo "Error: " . $ex->message;
            exit;
        }
        if($this->getpassword())
        {
            if (isset($_SESSION['user_new']))
            {
                $query="UPDATE `usuario` SET
`nick`='" .
mysql_real_escape_string($_SESSION['user_new'], $dblink) ."'
, `nombrecompleto`='" . mysql_real_escape_string($this-
>getnombrecompleto(), $dblink) ."' , `poblacion`='" .
mysql_real_escape_string($this->getpoblacion(), $dblink) ."'
, `password`='" . md5(mysql_real_escape_string($this-
>getpassword(), $dblink)) ."' , `avatar`='" .
mysql_real_escape_string($this->getavatar(), $dblink) ."' ,
`openid`='" . mysql_real_escape_string($this-
>getopenid(), $dblink) ."' , `idioma`='" .

```

```

mysql_real_escape_string($this->getidioma(), $dblink)."
WHERE `nick` ='" . mysql_real_escape_string($this-
>getnick(), $dblink) . "' LIMIT 1 ";
    }
    else
    {
        $query="UPDATE `usuario` SET
`nick`='" . mysql_real_escape_string($this-
>getnick(), $dblink)." , `nombrecompleto`='" .
mysql_real_escape_string($this-
>getnombrecompleto(), $dblink)." , `poblacion`='" .
mysql_real_escape_string($this->getpoblacion(), $dblink) ."
, `password`='" . md5(mysql_real_escape_string($this-
>getpassword(), $dblink)). "' , `avatar`='" .
mysql_real_escape_string($this-
>getavatar(), $dblink)." , `openid`='" .
mysql_real_escape_string($this->getopenid(), $dblink)." ,
`idioma`='" . mysql_real_escape_string($this-
>getidioma(), $dblink)." WHERE `nick` ='" .
mysql_real_escape_string($this->getnick(), $dblink) . "'
LIMIT 1 ";
    }
}
else
{
    if (isset($_SESSION['user_new']))
    {
        $query="UPDATE `usuario` SET
`nick`='" .
mysql_real_escape_string($_SESSION['user_new'], $dblink)."
, `nombrecompleto`='" . mysql_real_escape_string($this-
>getnombrecompleto(), $dblink)." , `poblacion`='" .
mysql_real_escape_string($this->getpoblacion(), $dblink) .
"' , `avatar`='" . mysql_real_escape_string($this-
>getavatar(), $dblink)." , `openid`='" .
mysql_real_escape_string($this->getopenid(), $dblink)." ,
`idioma`='" . mysql_real_escape_string($this-
>getidioma(), $dblink)." WHERE `nick` ='" .
mysql_real_escape_string($this->getnick(), $dblink) ."
LIMIT 1";
    }
    else
    {
        $query="UPDATE `usuario` SET
`nick`='" . mysql_real_escape_string($this-
>getnick(), $dblink)." , `nombrecompleto`='" .
mysql_real_escape_string($this-
>getnombrecompleto(), $dblink)." , `poblacion`='" .

```

```

mysql_real_escape_string($this->getpoblacion(), $dblink) .
" , `avatar`='". mysql_real_escape_string($this-
>getavatar(), $dblink)." , `openid`='".
mysql_real_escape_string($this->getopenid(), $dblink)." ,
`idioma`='". mysql_real_escape_string($this-
>getidioma(), $dblink)." WHERE `nick`='".
mysql_real_escape_string($this->getnick(), $dblink) ."
LIMIT 1";
    }
}

    if (mysql_query($query, $dblink)) {
        if(is_resource($dblink))
mysql_close($dblink);
        return true;
    }else{
        if(is_resource($dblink))
mysql_close($dblink);
        return false;
    }

        if(is_resource($dblink))
mysql_close($dblink);
    }

//function para recuperar los datos del usuario pasando
nick y pass (function necesaria para la acción de login)
    public function Recuperar($nick, $pass) {
        $dblink = null;
        try {
            $dblink =
mysql_connect(DB_HOST, DB_USER, DB_PASS);
            mysql_select_db(DB_BASE, $dblink);
        } catch(Exception $ex) {
            echo "Could not connect to " . DB_HOST
. ":" . DB_BASE . "\n"; echo "Error: " . $ex->message;
            exit;
        }
        $query ="select * from usuario where nick
like '".mysql_real_escape_string($nick, $dblink) . "' and
password like '".
md5(mysql_real_escape_string($pass, $dblink))."'";
        $result=mysql_query($query);
        if(mysql_num_rows($result)>0){
            $objeto=mysql_fetch_object($result);

```

```

        $this->setnick($nick);
        $this->setnombrecompleto($objeto-
>nombrecompleto);
        $this->setpoblacion($objeto-
>poblacion);
        $this->setavatar($objeto->avatar);

        $this->setopenid($objeto->openid);
        $this->setidioma($objeto->idioma);

        return true;
    }else{
        return false;
    }
}

//variante de la function anterior que recupera los datos
de un usuario unicamente pasandole su nick
    public function Recuperarsimple($nick) {
        $dblink = null;
        try {
            $dblink =
mysql_connect(DB_HOST,DB_USER,DB_PASS);
            mysql_select_db(DB_BASE,$dblink);
        } catch(Exception $ex) {
            echo "Could not connect to " . DB_HOST
. ":" . DB_BASE . "\n"; echo "Error: " . $ex->message;
            exit;
        }
        $query ="select * from usuario where nick
like '".mysql_real_escape_string($nick,$dblink) ."'";
        $result=mysql_query($query);
        if(mysql_num_rows($result)>0){
            $objeto=mysql_fetch_object($result);
            $this->setnick($nick);
            $this->setnombrecompleto($objeto-
>nombrecompleto);
            $this->setpoblacion($objeto-
>poblacion);
            $this->setavatar($objeto->avatar);

            $this->setopenid($objeto->openid);
            $this->setidioma($objeto->idioma);

            return true;
        }else{
            return false;
        }
    }
}

```

```
    }

    //conjunto de funciones GET y SET (para consultar y
    //modificar variables del objeto usuario)
    public function setnick($nick) {
        $this->nick = $nick;
        return true;
    }
    public function getnick() {
        return $this->nick;
    }
    public function
setnombrecompleto($nombrecompleto) {
        $this->nombrecompleto = $nombrecompleto;
        return true;
    }
    public function getnombrecompleto() {
        return $this->nombrecompleto;
    }
    public function setpoblacion($poblacion) {
        $this->poblacion = $poblacion;
        return true;
    }
    public function getpoblacion() {
        return $this->poblacion;
    }
    public function setpassword($password) {
        $this->password = $password;
        return true;
    }
    public function getpassword() {
        return $this->password;
    }
    public function setopenid($openid) {
        $this->openid = $openid;
        return true;
    }
    public function getopenid() {
        return $this->openid;
    }
    public function setidioma($idioma) {
        $this->idioma = $idioma;
        return true;
    }
    public function getidioma() {
        return $this->idioma;
    }
}
```

```
//función para guardar la imagen del usuario (avatar) en
una carpeta dentro de nuestro servidor
    public function guardaavatar($img) {
        $tamano = $img['size'];
        $tipo = $img['type'];
        $archivo = $img['name'];
        $prefijo = substr(md5(uniqid(rand())),0,6);

        if ($archivo != "") {
            // guardamos el archivo a la carpeta
            $destino =
URL_AVATARES.$prefijo."_".$archivo;
            if
(copy($_FILES['archivo']['tmp_name'],$destino)) {
                $this->img =
$prefijo."_".$archivo;

                return true;
            } else {
                return false;
            }
        } else {
            return false;
        }

        $this->img = $img;
        return true;
    }
    public function setavatar($img) {

        return $this->img=$img;
    }
    public function getavatar() {
        return $this->img;
    }
    public function getamigos() {
        $dblink = null;
        try {
            $dblink =
mysql_connect(DB_HOST,DB_USER,DB_PASS);
            mysql_select_db(DB_BASE,$dblink);
        } catch(Exception $ex) {
            echo "Could not connect to " . DB_HOST
. ":" . DB_BASE . "\n"; echo "Error: " . $ex->message;
            exit;
        }

        $query ="select * from amigos where solicita
like '".mysql_real_escape_string($this->getnick(),$dblink)
. "'";

        $result=mysql_query($query);
```

```

        if(mysql_num_rows($result)>0){

            while($objeto=mysql_fetch_object($result)){
//lo hacemos asi por que si no creamos en cada posicion del
//array un usuario nuevo, y creamos un usuario que va
//cambiando, lo que hace es cambiar el valor de todas las
//posiciones dinamico vs estatico
                $amigos[$objeto->
>es_solicitado]=new usuario(1);
                $amigos[$objeto->es_solicitado]-
>Recuperarsimple($objeto->es_solicitado);
            }
            return $amigos;
        }else{
            return false;
        }
    }

//funciones para añadir y eliminar un amigo

    public function solicitudamigo($amigo) {
        $dblink = null;
        try {
            $dblink =
mysql_connect(DB_HOST,DB_USER,DB_PASS);
            mysql_select_db(DB_BASE,$dblink);
        } catch(Exception $ex) {
            echo "Could not connect to " . DB_HOST
. ":" . DB_BASE . "\n"; echo "Error: " . $ex->message;
            exit;
        }
        $query ="select * from amigos where solicita
like '".mysql_real_escape_string($this->getnick(),$dblink)
.'" and es_solicitado like
'".mysql_real_escape_string($amigo,$dblink) ."'" ;
        $result=mysql_query($query);
        if(mysql_num_rows($result)==0){

            $query="INSERT INTO amigos
(`solicita`,`es_solicitado`,`confirmado`) VALUES ('" .
mysql_real_escape_string($this->getnick(),$dblink) . "',"'"
. mysql_real_escape_string($amigo,$dblink) . "', 0)";
            $result=mysql_query($query);
            return true;
        }
    }
}

```

```

        public function quitaramigo($amigo) {
            $dblink = null;
            try {
                $dblink =
mysql_connect(DB_HOST,DB_USER,DB_PASS);
                mysql_select_db(DB_BASE,$dblink);
            } catch(Exception $ex) {
                echo "Could not connect to " . DB_HOST
. ":" . DB_BASE . "\n"; echo "Error: " . $ex->message;
                exit;
            }

            $query ="select * from amigos where
solicita like '".mysql_real_escape_string($this-
>getnick(),$dblink) .' and es_solicitado like
' ".mysql_real_escape_string($amigo,$dblink) ."'";
            $result=mysql_query($query);
            if(mysql_num_rows($result)>0){
                //deberia ser un update a
solicitado=false
                $query ="delete from amigos where
solicita like '".mysql_real_escape_string($this-
>getnick(),$dblink) .' and es_solicitado like
' ".mysql_real_escape_string($amigo,$dblink) ."'";
                $result=mysql_query($query);
                return true;
            }
        }
    }
}
?>

```

De la misma forma el resto de clases tiene su correspondiente archivo.php con: constructor, update, funciones para recuperar el objeto, GETS y SETS, y las funciones de consulta y modificación de los datos de su clase pertinentes.

Anexo 1: Registro del dominio

Para poder acceder a la web necesitaremos un dominio para no tener que acceder a la web mediante una dirección IP y un hosting para albergar la web y su base de datos.

Adjuntamos captura del proceso de registro del hosting y el dominio en RedCoruna.

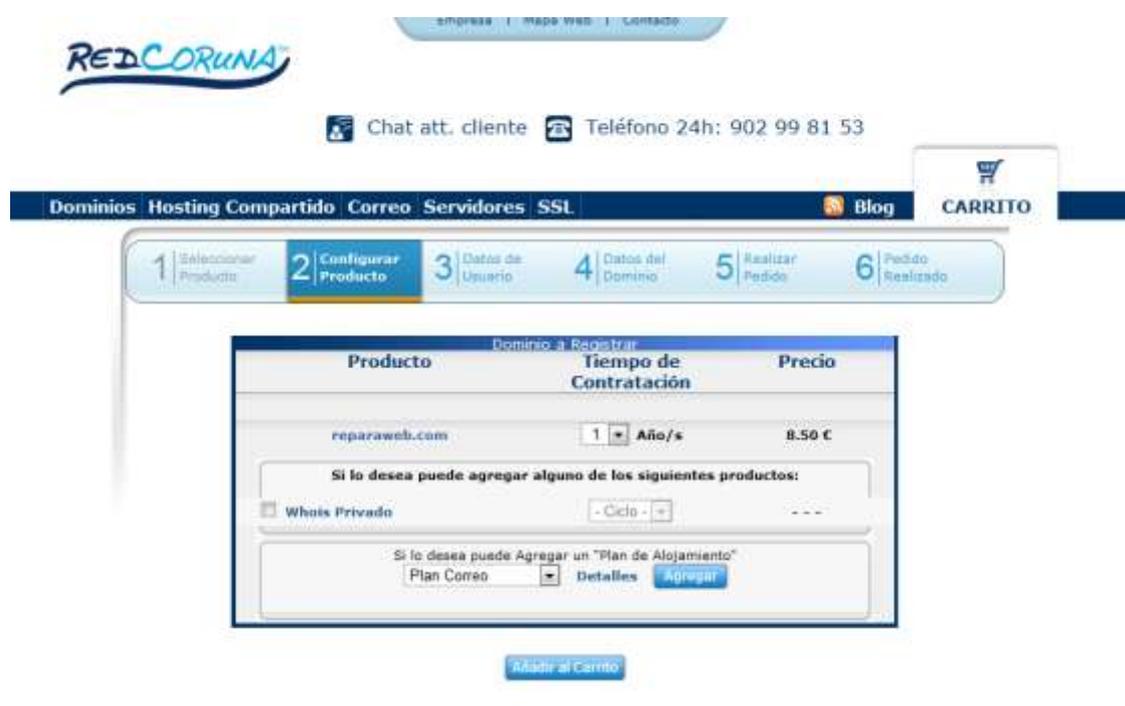


Fig (53) Ejemplo registro de dominio

Anexo 2: Modelo de negocio.

Aquí se detallarán de forma breve distintos modelos de negocio o de rentabilizar nuestra web.

- Una opción a tener en cuenta es la publicidad de Google (adsense). Lo cual nos generara ingresos cuando los usuarios visiten dicha publicidad.
- Una segunda opción base a tu geo localización IP, se te mostraran unas publicidades que estén próximas a tu zona, distintas publicidades de patrocinadores del mundo del motor. La publicidad será contratada por agencia o directamente desde los talleres que quieran contratar con nosotros.
- Se pueden ofrecer servicios de pago a modo de cuentas Premium, por ejemplo para los dueños de posibles talleres que quieran darse a conocer, tener la opción mediante pago, de crear una ficha más detallada y opciones adicionales como promoción de ofertas de dicho taller.
- Siempre está la opción de habilitar un servicio de donaciones, para que los usuarios que lo deseen hagan donaciones como muestra de agradecimiento por nuestro servicio ofrecido.

Anexo 3: Manual de usuario.

ÍNDICE

[Introducción](#) 2

[Menú De Usuario \(anónimo\)](#) 3

[Mapa](#) 4

[Mejores talleres](#) 5

[Mejor valoradas](#) 6

[Registro de usuario](#) 7

[Menú De Usuario](#) 7

[Ver Mis Datos](#) 8

[Añadir Vehículo](#) 8

[Mis Vehículos](#) 9

[Amigos](#) 10

[Mensajes](#) 11

[Insertar Taller](#) 12

[Ver Mis Talleres](#) 13

[Modificar Taller](#) 13

[Insertar Reparación](#) 14

INTRODUCCIÓN

El manual de usuario es un documento que tiene como objetivo dar apoyo al usuario final sobre las características principales que poseen cada módulo o proceso del sistema Web

“Red social de reparaciones de coches” entre los procesos tenemos consultar/añadir reparaciones, consultar/añadir los talleres, registrarse, buscar/añadir amigos y enviar/recibir mensajes.

MENÚ DEL USUARIO

Al ingresar a la página Web se nos presentara o visualizara la página principal que siendo usuario anónimo podrá hacer las opciones de la barra de color negro.

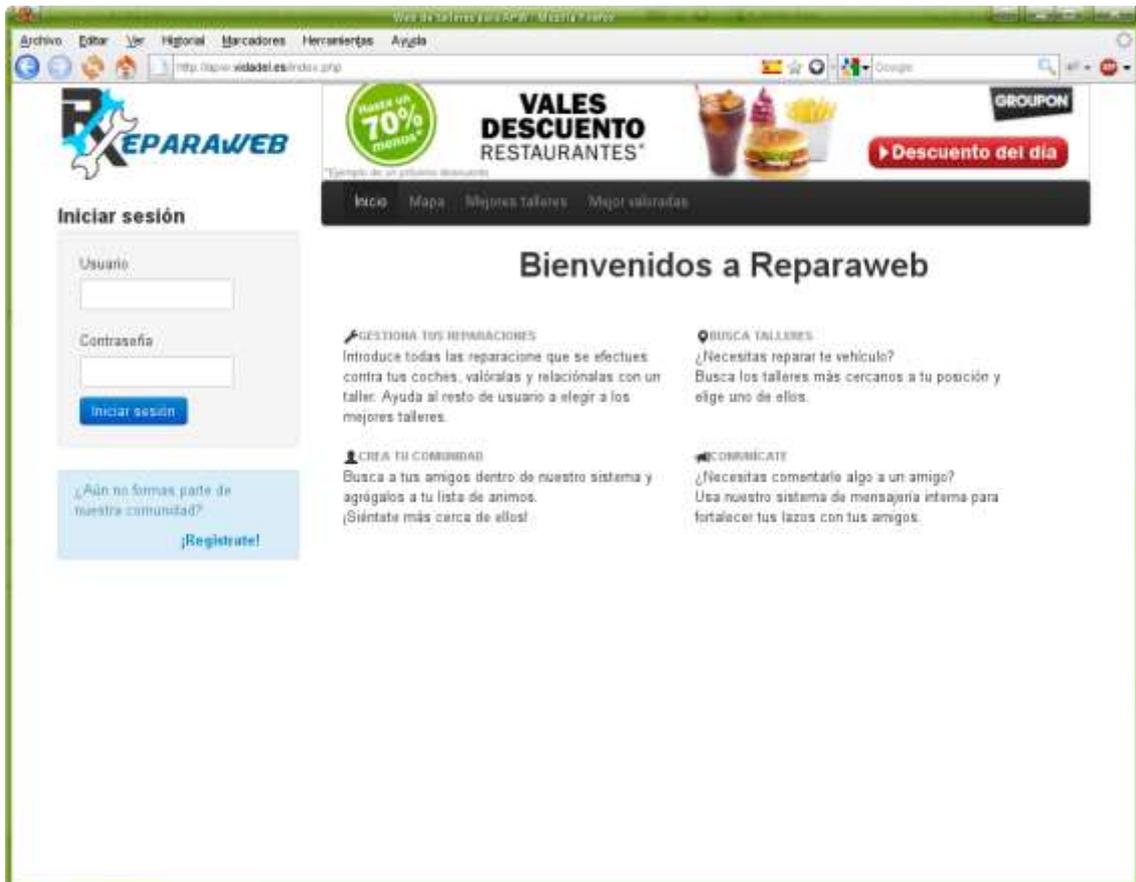


Fig (54) Menú usuario No registrado

Mapa: Ir al Mapa del mundo, con los talleres dados de alta en la web.

Mejores talleres: Ir a los talleres con mejor valoración.

Mejor valoradas: Ir a las reparaciones con mejor valoración.

Si optamos por ir a la opción **MAPA**, del menú, podremos ver el mapa y moviéndonos por el para ver los talleres situados en el mapa.

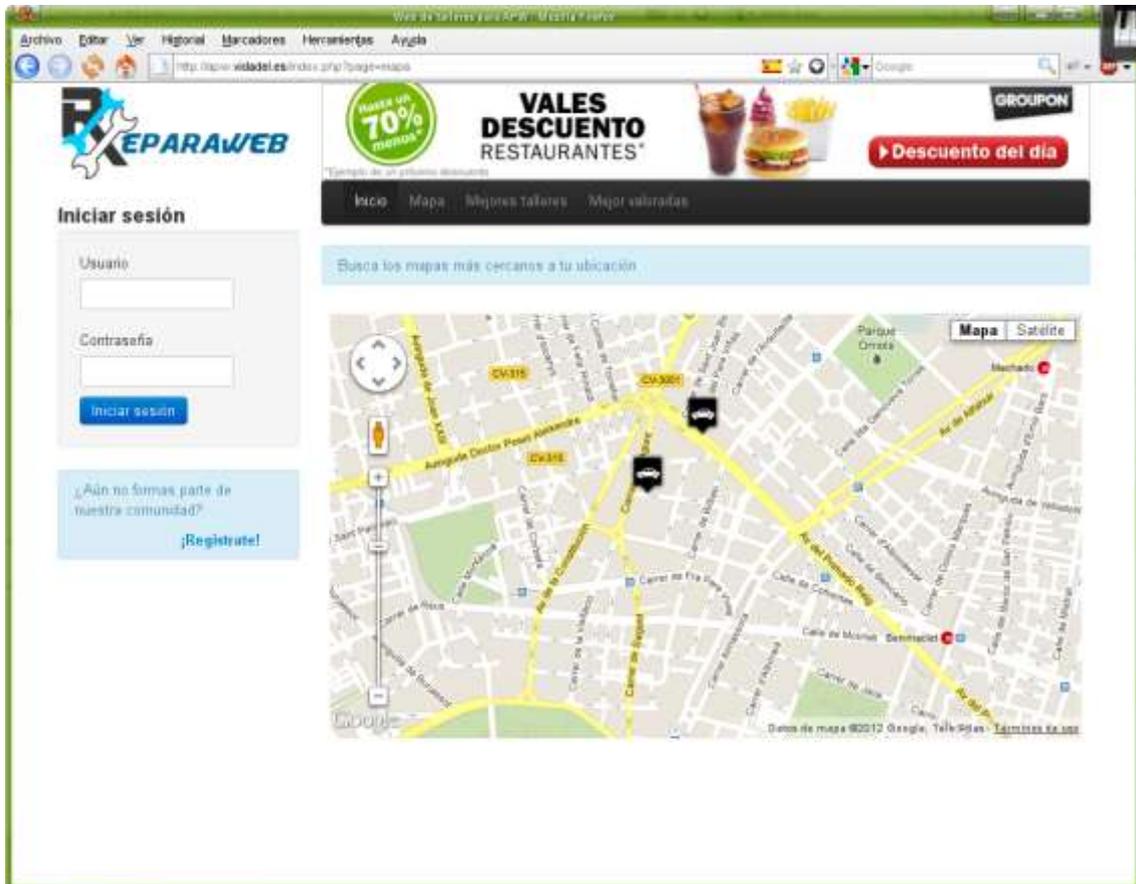


Fig (55) Mapa

Podemos, arrastrar el ratón sobre el mapa para poder movernos. También, podremos con el + y el – acercarnos o alejarnos para ver con más detalle o más porción de mapa.

Si se desea, arrastrando el muñeco amarillo  a una zona del mapa, se podrá ver a vista de calle la zona que se desee.

Si optamos por ir a la opción **MEJORES TALLERES**, del menú, podremos ver la lista de talleres mejores valorados, viendo la puntuación o valoración en formato de estrellas.

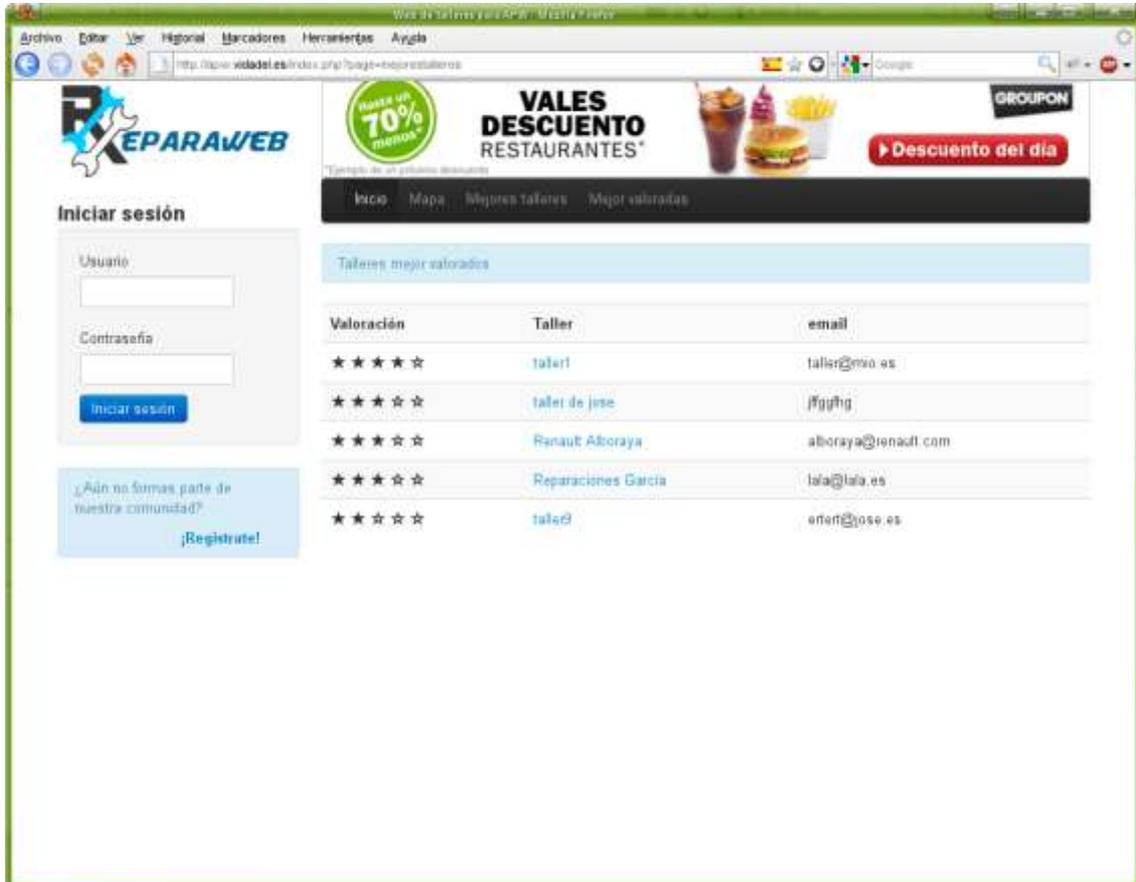


Fig (56) Mejores talleres

Si se desea ver la información de ese taller, como dirección etc, haciendo “click”, sobre el nombre del taller nos llevará a mostrar la información del taller.

Si optamos por ir a la opción **MEJOR VALORADAS**, del menú, podremos ver la lista de las reparaciones mejor valorados, viendo la puntuación o valoración en formato de estrellas.

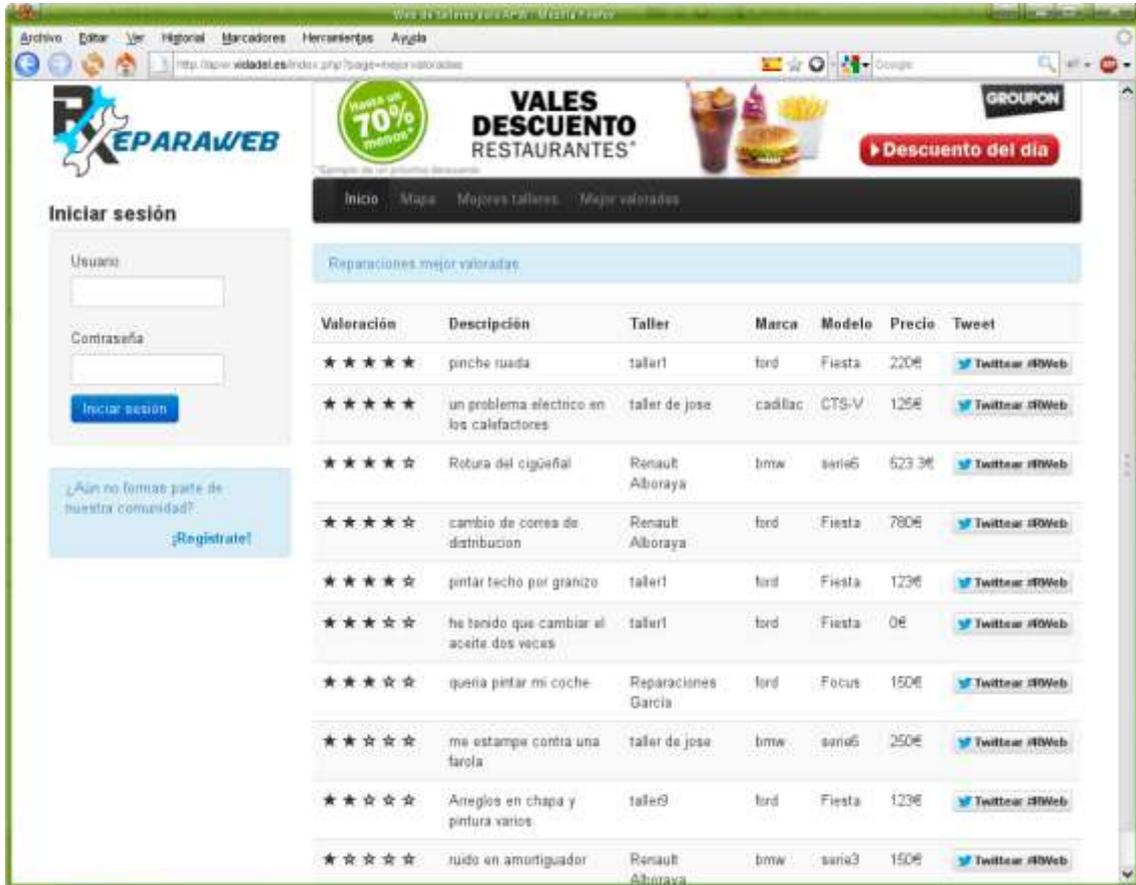


Fig (57) Mejor valoradas

Si se desea se puede usar su cuenta de “Twitter”, para twittear esa reparación.

Si en la página principal, optamos por registrarnos nos mostrara un formulario donde deberemos rellenar cada uno de los apartados para formar parte de la página web, y así ser un usuario registrado.



Fig (58) Pantalla inicio Usuario Registrado

Una vez registrado, como usuario aparecerá un menú con las distintas posibilidades:

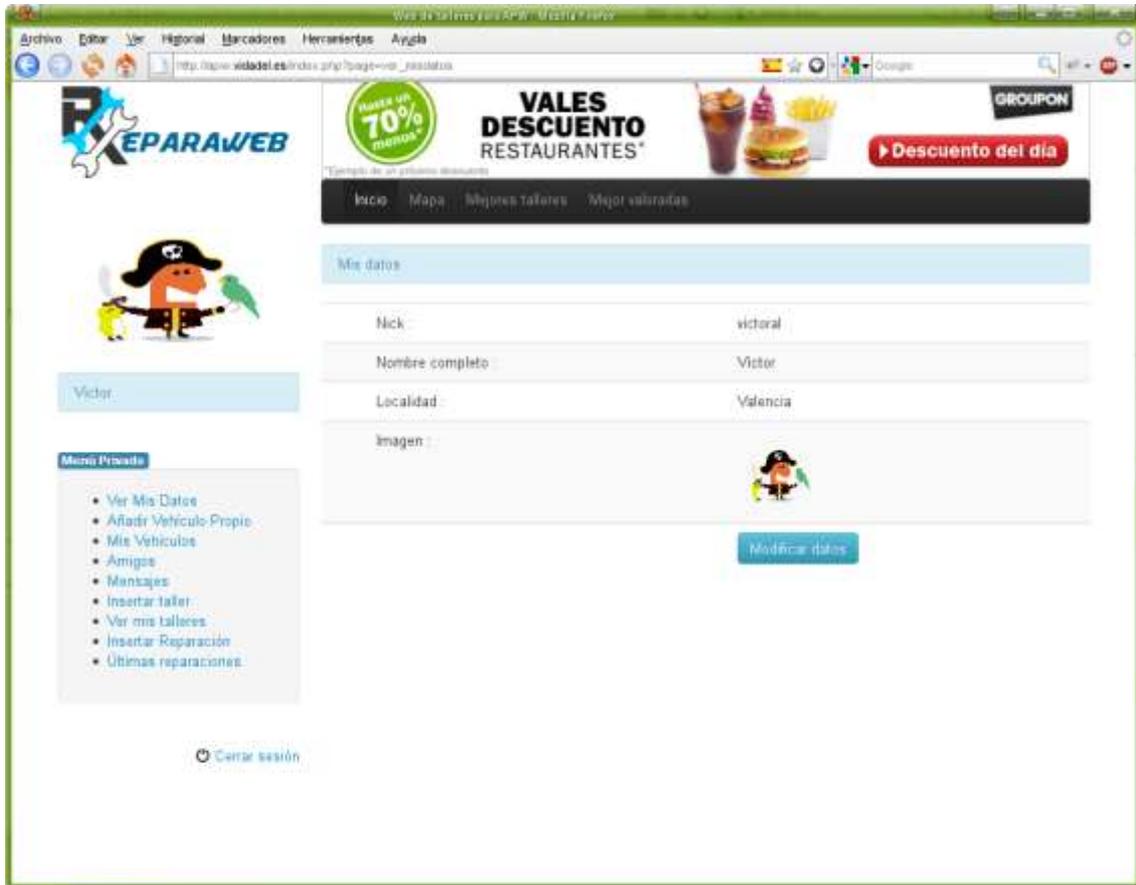
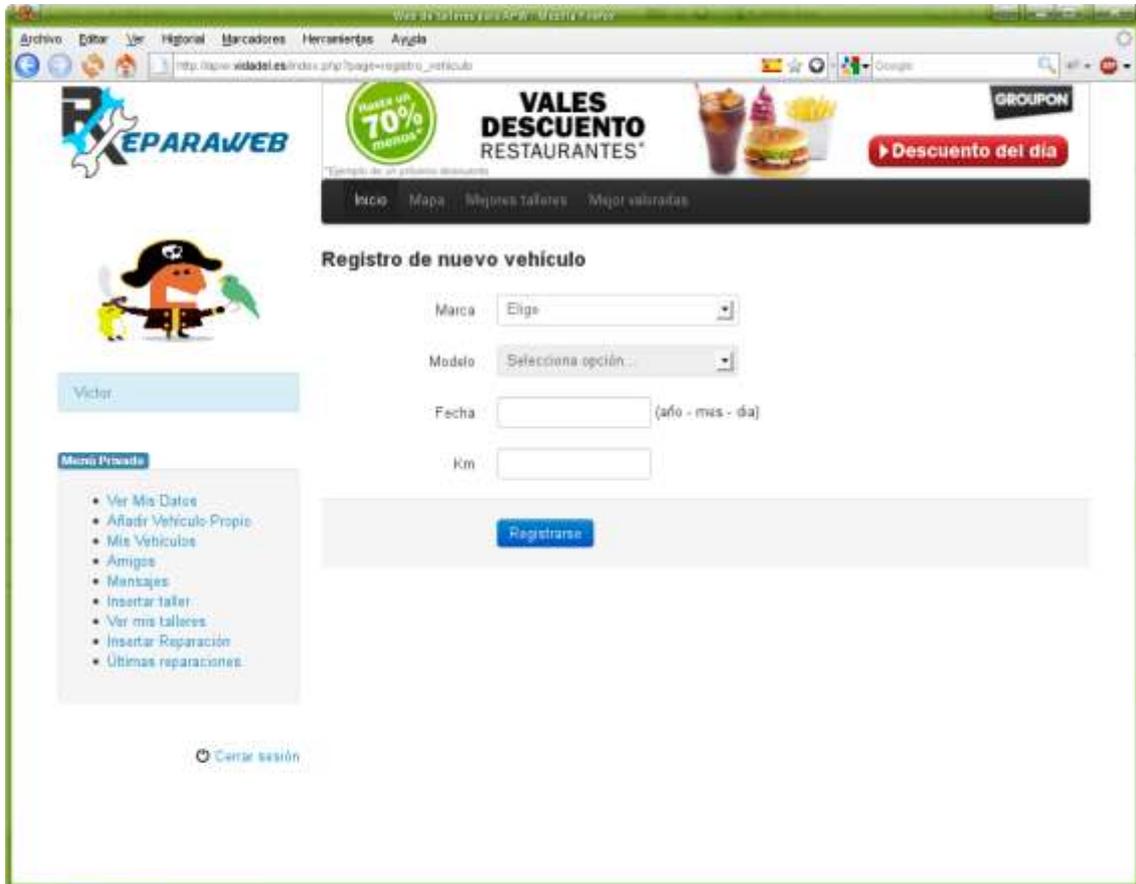


Fig (59) Perfil de usuario

Si pulsáramos **Modificar datos**, nos llevaría a otra página, similar a la página de registro, pero con las casillas con nuestros datos para poder modificarlos.

Añadir Vehículo Propio: Añadir un vehículo a tu cuenta de usuario.



The image shows a screenshot of a web browser displaying the 'Registro de nuevo vehículo' (New Vehicle Registration) form on the REPARAWEB website. The browser's address bar shows the URL 'http://www.viajes.es/index.php?page=registro_vehiculo'. The website header includes the REPARAWEB logo, a '70% menos' discount badge, and a 'VALES DESCUENTO RESTAURANTES' promotion. The main content area features a 'Registro de nuevo vehículo' section with the following fields: 'Marca' (a dropdown menu with 'Elige' selected), 'Modelo' (a dropdown menu with 'Selecciona opción...' selected), 'Fecha' (a text input field with '(año - mes - día)' as a placeholder), and 'Km' (a text input field). A blue 'Registrarse' button is located below the form. On the left side, there is a user profile section with a cartoon pirate character and a 'Ver' button. Below this is a 'Menú Privado' section with a list of links: 'Ver Mis Datos', 'Añadir Vehículo Propio', 'Mis Vehículos', 'Amigos', 'Mantajes', 'Insertar taller', 'Ver mis talleres', 'Insertar Reparación', and 'Últimas reparaciones'. At the bottom left, there is a 'Cerrar sesión' button.

Fig (60) Formulario añadir vehículo de usuario

En esta página, podremos añadir nuestro vehículo, con su marca, modelo, antigüedad del vehículo, y kilometraje. Una vez registrado el vehículo, podrás añadir reparaciones del vehículo.

Mis Vehículos: Ver los vehículos registrados en tu usuario.



Fig (61) Listado vehículos de usuario

Pulsando sobre el icono Visualizar, podrás visualizar todos los detalles del vehículo.

No existe, de momento la posibilidad de modificar los datos del vehículo.

Amigos: Ver o añadir amigos (otros usuarios de la web) a tu cuenta de usuario.

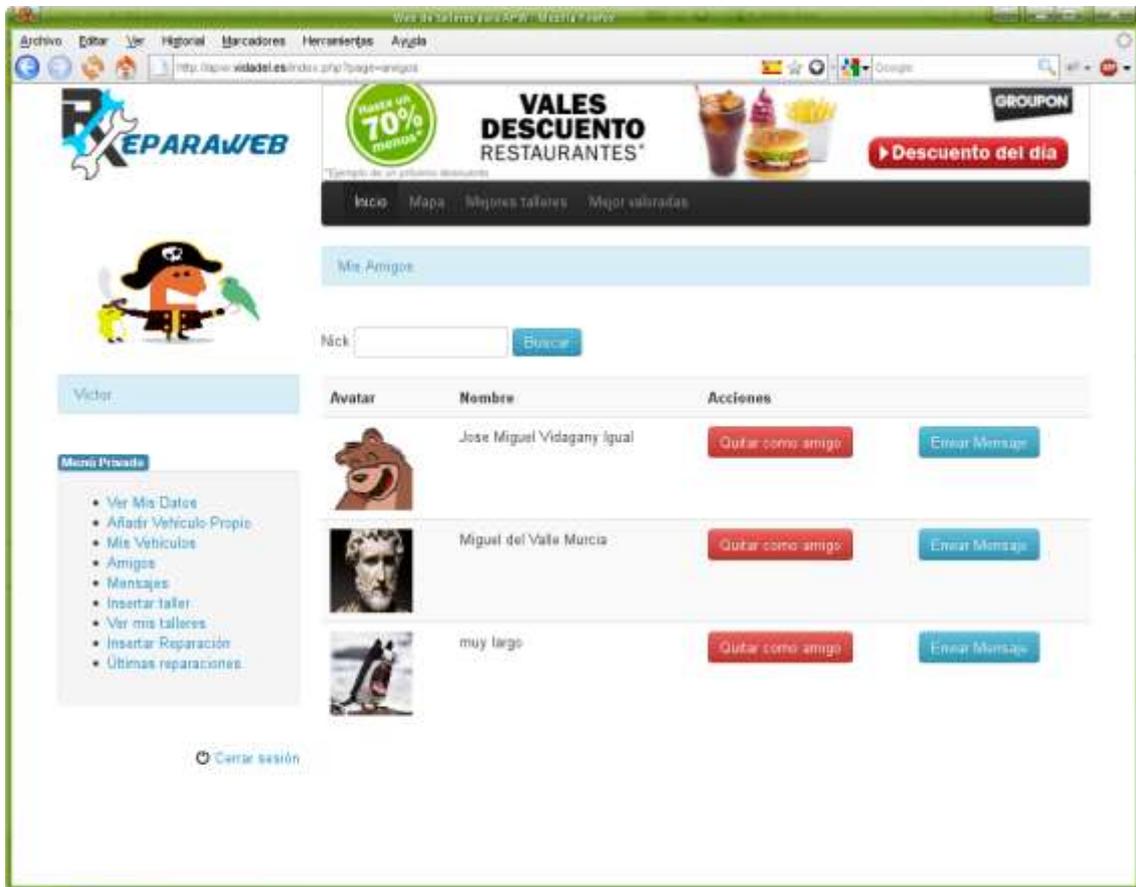


Fig (62) Sección mis amigos

Se pueden buscar nuevos usuarios en el programa, introduciendo un nombre en el recuadro, y pulsando sobre el botón Buscar.

Si pulsas sobre el botón **Quitar como amigo**, lo eliminaras de tu lista de amigos. Y si deseas enviarle un mensaje privado, puedes hacerlo a través del botón **Enviar Mensaje**.

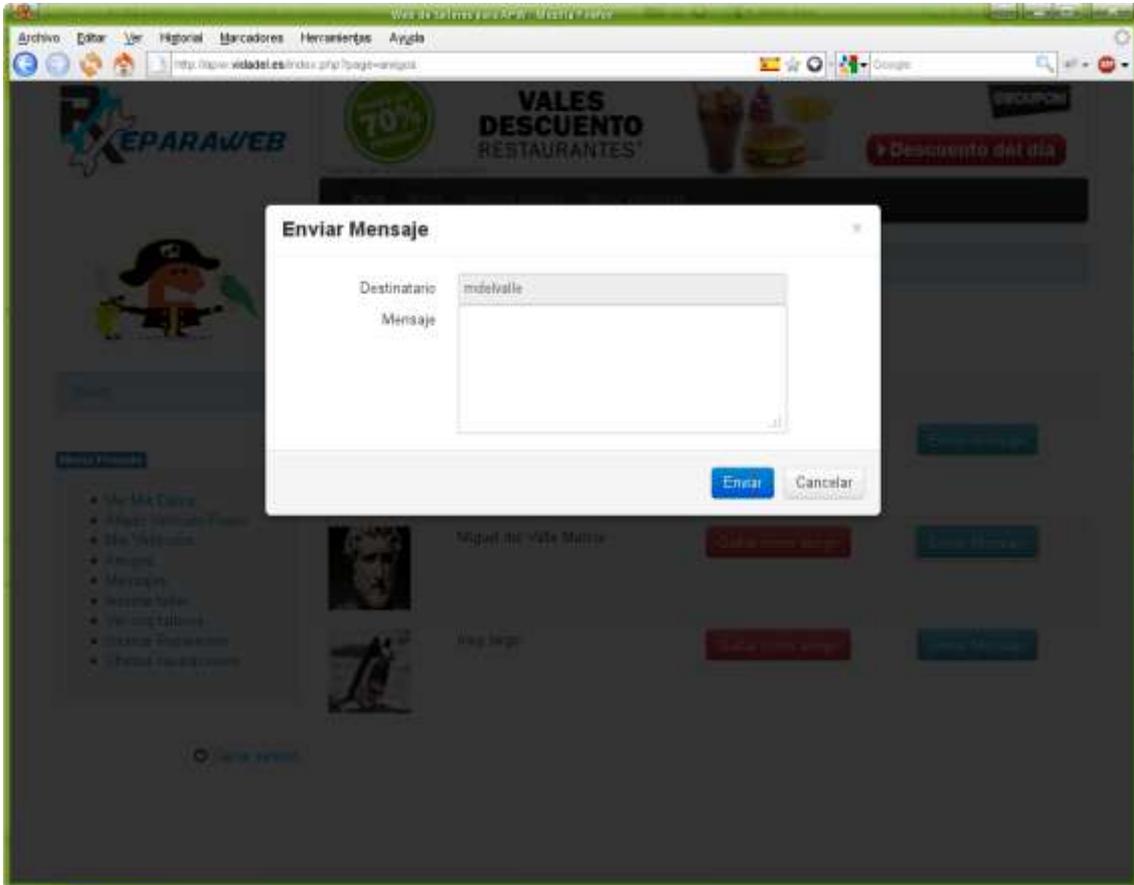


Fig (63) Enviar mensaje

Escribiendo lo que se desee enviar en el recuadro Mensaje, y seguidamente pulsando el botón de Enviar.

Si se desea ver los mensajes que has enviado o que te han enviado lo podrás hacer desde la opción del Menú Privado **Mensajes**.

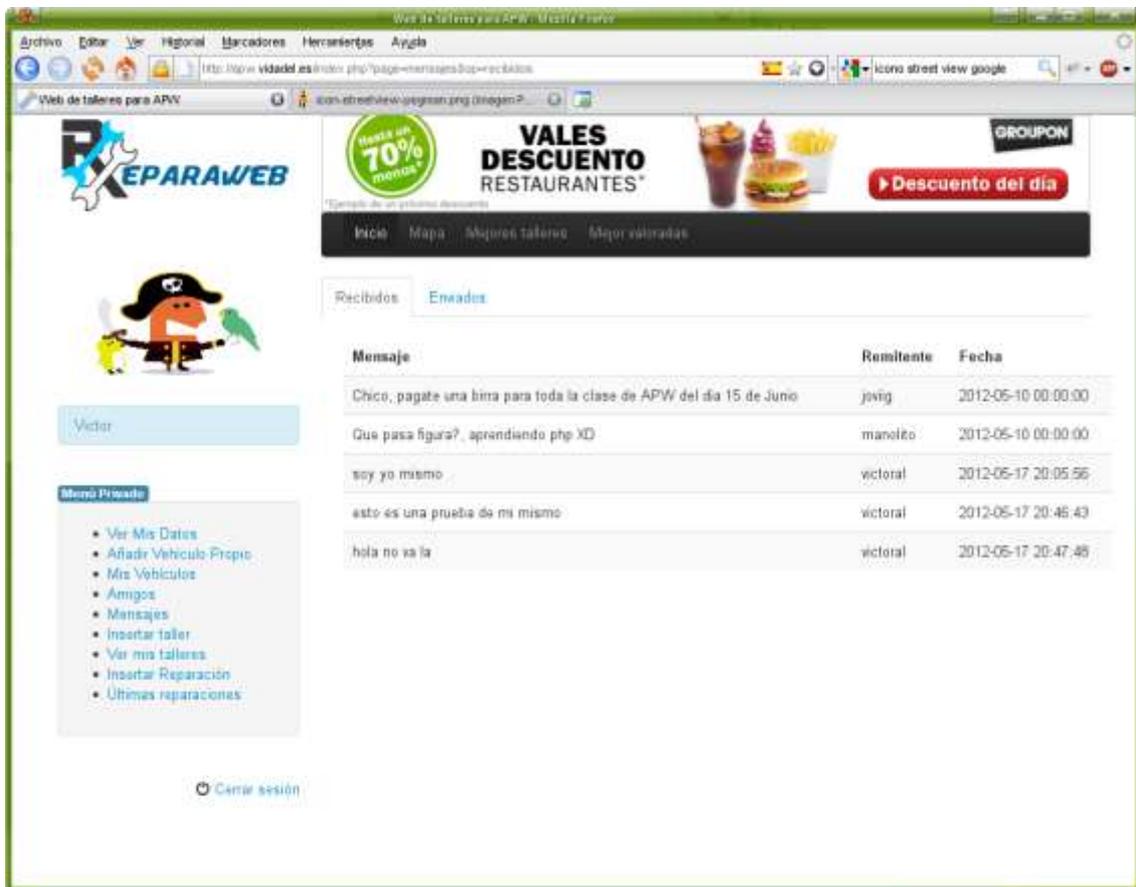
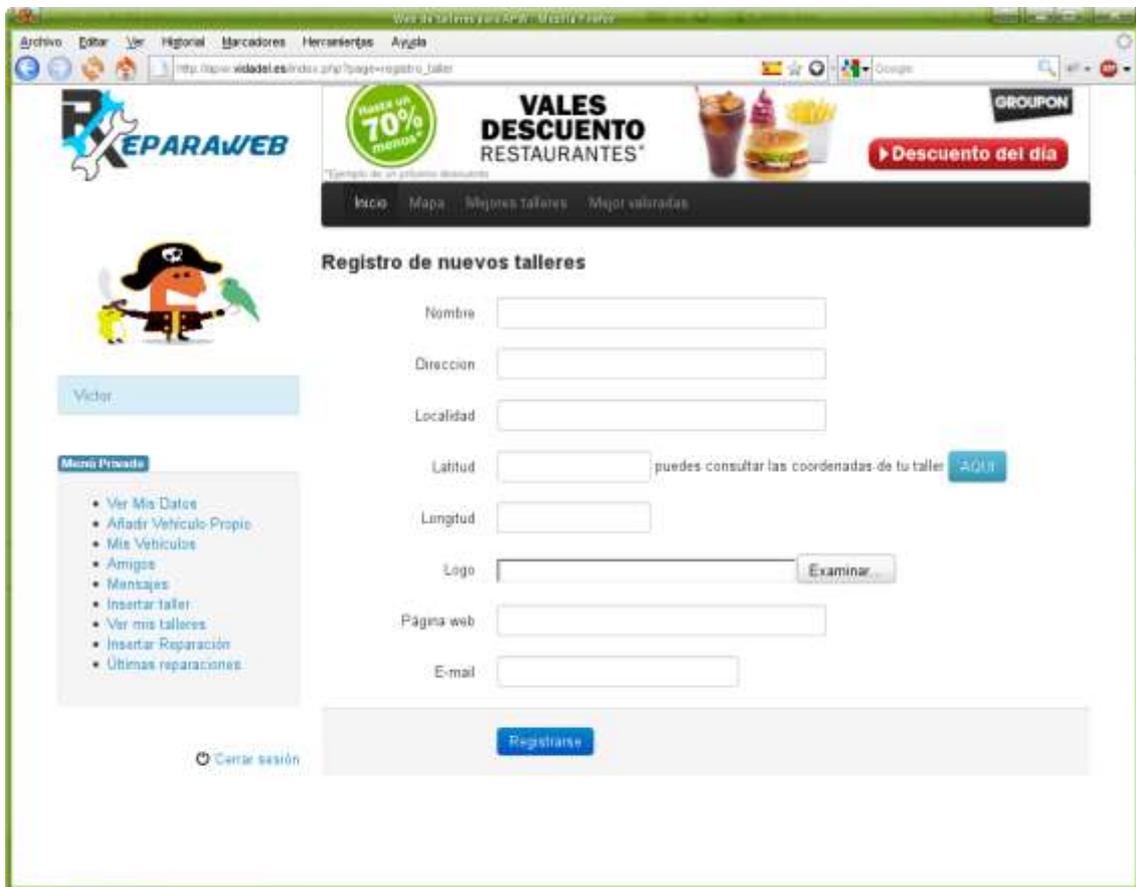


Fig (64) Mensajes recibidos/enviados

Podrá ver dos pestañas, **Recibidos y Enviados**, desde la que podrá alternar la vista de mensajes que ha enviado y los que ha recibido.

Si desea dar de alta un taller, al cual añadir más tarde una reparación, podrá hacerlo desde la opción **Insertar Taller**.



The image shows a web browser window displaying the 'Registro de nuevos talleres' (New Workshop Registration) form on the REPARAWEB website. The browser's address bar shows the URL 'http://www.vivadades/index.php?page=registro_taller'. The website header includes the REPARAWEB logo, a '70% menos' discount badge, and a 'VALES DESCUENTO RESTAURANTES' promotion. The main content area features a 'Registro de nuevos talleres' section with the following fields: 'Nombre', 'Direccion', 'Localidad', 'Latitud', 'Longitud', 'Logo', 'Página web', and 'E-mail'. A blue button labeled 'Registrar' is at the bottom of the form. To the right of the 'Latitud' field, there is a link that says 'puedes consultar las coordenadas de tu taller AQUÍ'. A sidebar on the left contains a 'Mi Perfil' menu with options like 'Ver Mis Datos', 'Añadir Vehículo Propio', 'Mis Vehículos', 'Amigos', 'Mantajes', 'Insertar taller', 'Ver mis talleres', 'Insertar Reparación', and 'Últimas reparaciones'. A 'Cerrar sesión' link is also visible at the bottom left of the sidebar area.

Fig (65) Formulario para insertar taller

Solo tendrá que rellenar los datos del taller.

Para una correcta ubicación, podrá ir al botón **AQUI**, que le permitirá obtener la longitud y latitud (coordenadas cartográficas), de la posición exacta del taller.

Desde la opción **Ver mis talleres**, podrá acceder a los talleres que usted ha dado de alta en la Web, así como poder Visualizar los datos del taller, mediante el botón **Visualizar**.

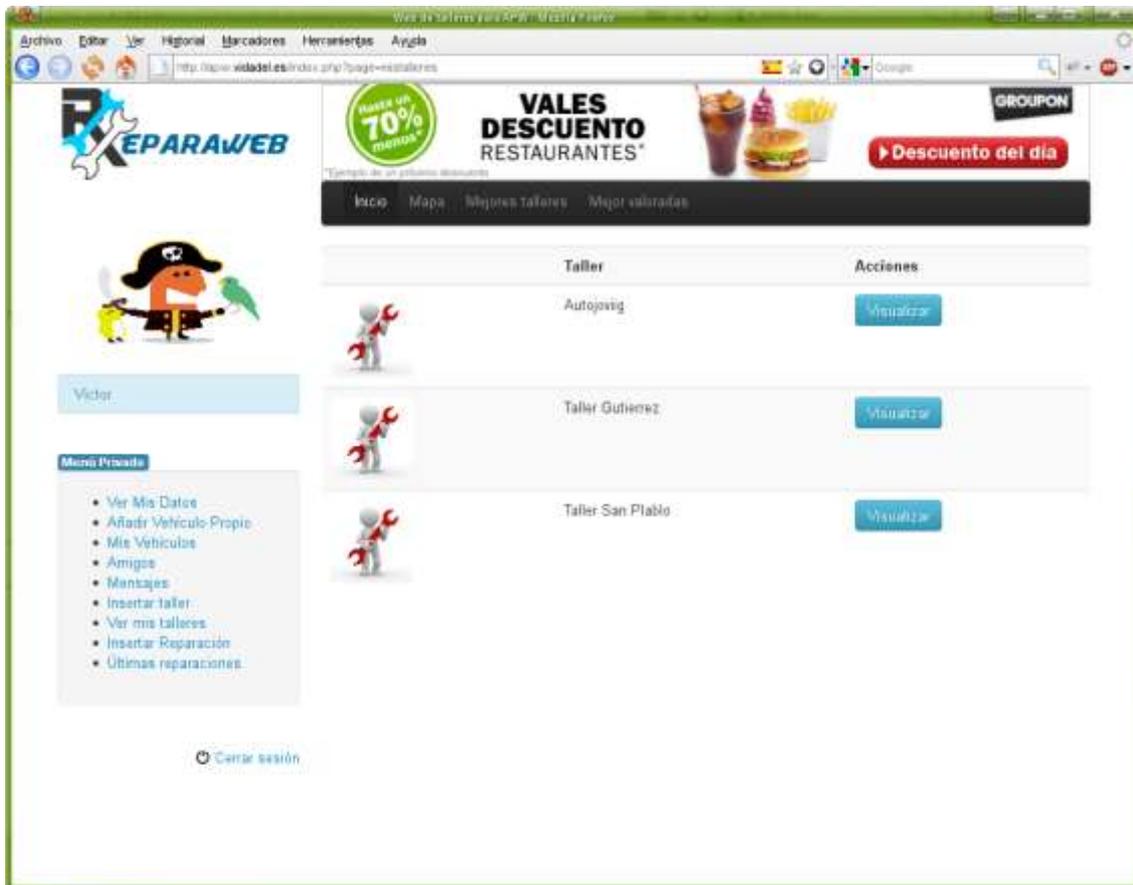


Fig (66) Listado de talleres dados de alta por el usuario

Desde esa página, podrá visualizar y **modificar los datos** del taller

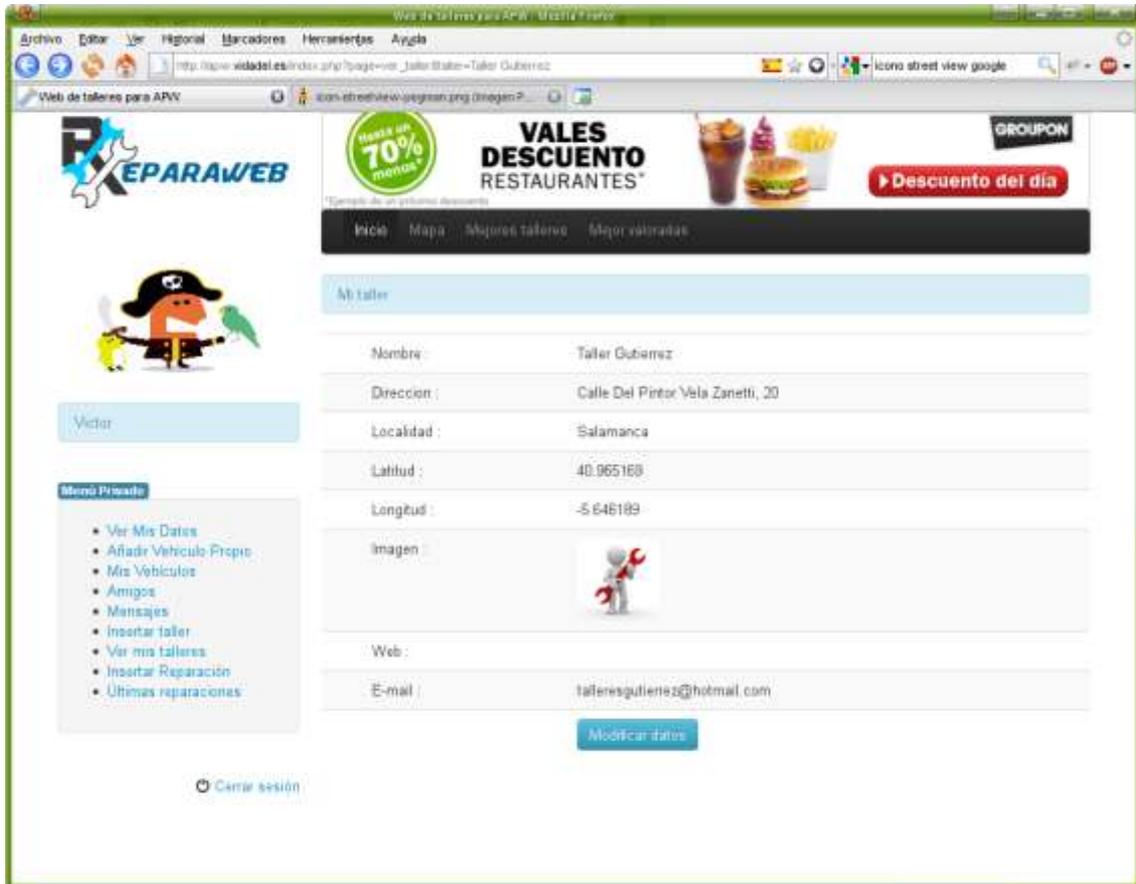
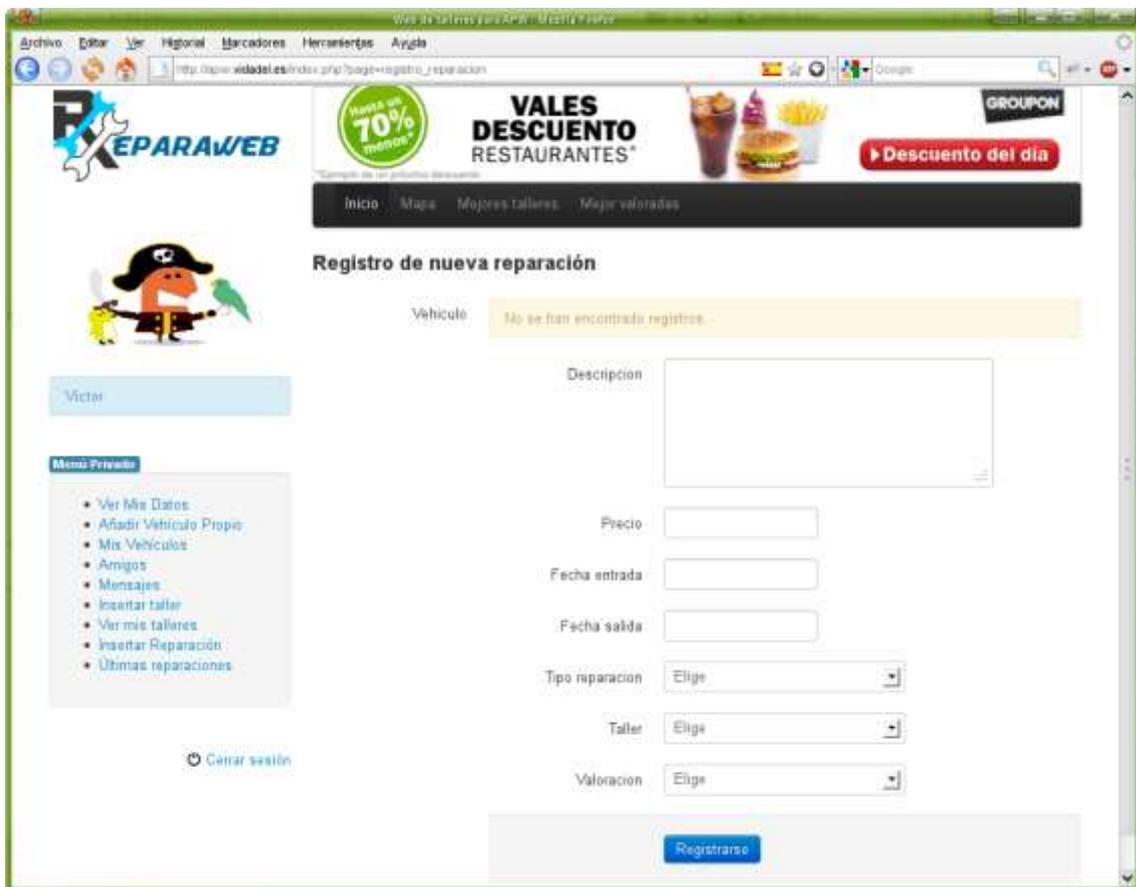


Fig (67) Modificar datos del taller

Si pulsáramos **Modificar datos**, nos llevaría a otra página, similar a la página de registro, pero con las casillas con los datos del taller para poder modificarlos.

Desde la última opción del menú **Insertar Reparación**, podrá añadir una reparación añadiendo los datos necesarios y asignando esta reparación a unos de los talleres que previamente teníamos registrados dentro de nuestro usuario.



The image shows a web browser window displaying the 'Registro de nueva reparación' (New Repair Registration) form on the REPARAWEB website. The browser's address bar shows the URL 'http://www.vialdes.es/index.php?page=registro_reparacion'. The website header includes the REPARAWEB logo, a '70% descuento' badge, and a 'VALES DESCUENTO RESTAURANTES' banner. The main content area features a navigation menu with 'Inicio', 'Mapa', 'Mejores talleres', and 'Mejor valoradas'. The form itself is titled 'Registro de nueva reparación' and includes a 'Vehículo' dropdown menu (currently showing 'No se han encontrado registros'), a 'Descripción' text area, a 'Precio' input field, 'Fecha entrada' and 'Fecha salida' input fields, and three dropdown menus for 'Tipo reparación', 'Taller', and 'Valoración' (all currently set to 'Elige'). A 'Registrarse' button is located at the bottom of the form. On the left side, there is a user profile section for 'Victor' and a 'Menú Privado' with links to 'Ver Mis Datos', 'Añadir Vehículo Propio', 'Mis Vehículos', 'Amigos', 'Mensajes', 'Insertar taller', 'Ver mis talleres', 'Insertar Reparación', and 'Últimas reparaciones'. A 'Cerrar sesión' link is also visible.

Fig (68) Formulario para insertar una reparación

Aclarar que el Tipo de reparación, es un listado que el administrador del sitio ha dado de alta, no pudiendo por parte del usuario elegir un tipo de reparación distinto al de ese listado.

6 Conclusiones

6.1 Fases del proyecto

- El proyecto se ha realizado en las siguientes fases:
- Búsqueda de Webs similares.
- Determinar la funcionalidad básica de nuestra web.
- Especificación de requisitos.
- Diseño de diagramas
- Implantación del diseño
- Codificación
- Pruebas
- Corrección de errores
- Pruebas finales
- Corrección final de errores.

6.2 Dificultades

La dificultad para llevar a cabo el proyecto ha sido grande, ya que aunque no era mi primera aplicación Web, si era la primera vez que usaba alguna de las tecnologías o herramientas empleadas, he tenido que realizar un gran esfuerzo debido a la cantidad de programación y su complejidad, lo que ha requerido gran interés y ambición por mi parte para adquirir los conocimientos necesarios y ponerlos en práctica.

6.3 Experiencia

He aprendido bastante, en gran parte, por las dificultades afrontadas. El desarrollo de esta aplicación me ha hecho conocer en profundidad un tipo de herramientas y lenguajes que antes no había manejado tan ampliamente y que son de un uso extendido, lo que me puede ayudar en un futuro a desarrollar proyectos y aplicaciones en distintos ámbitos.

De tener que destacar algo que me haya cautivado durante el desarrollo del proyecto sin duda han sido el manejo de la api de google, y ver como algo tan simple como un mapa puede ofrecer tantas posibilidades con la adecuada implementación y librerías usadas.

6.4 Trabajo futuro

A pesar de la gran cantidad de tiempo dedicado y las funcionalidades implementadas, he pensado una serie de funcionalidades más que podría tener la aplicación, que ya serían parte de otro proyecto a desarrollar.

Ideas para posibles ampliaciones futuras de la funcionalidad:

- ❖ Se podría mejorar la base de datos, por ejemplo, agregando una nueva tabla de ofertas sobre algún tipo de reparación de un determinado taller.
- ❖ Implementar algún método de localización ip (para ofrecer al usuario ofertas personalizadas de su zona de residencia).
- ❖ Se podría agregar un mini foro para cada modelo de coche o tipo de reparación, para que los usuarios comentaran acerca de ellos o buscaran información concreta.
- ❖ Agregar la opción a los usuarios de comentar una reparación de otro usuario, y no solo una valoración de la misma.
- ❖ El administrador podría moderar/eliminar mensajes de los comentarios/foro.
- ❖ (Aunque ya existe integración con redes sociales) Agregar más redes sociales y la integración para poder postear en ella directamente comentarios o valoraciones de nuestra web.
- ❖ Implementar un buscador de talleres/reparaciones donde muestre las fichas de los talleres/reparaciones solicitados.
- ❖ ...

7 Bibliografía

- [1] *Web 2.0*. <http://www.pcnet.com.es/internet/web20.html>
- Web 2.0*. http://es.wikipedia.org/wiki/Web_2.0
- [2] *PFC Aproximación Etica y legal a las redes sociales*.
<http://riunet.upv.es/bitstream/handle/10251/9123/PFC%20-%20Aproximaci%C3%B3n%20C3%89tica%20y%20Legal%20a%20las%20Redes%20Sociales.pdf>
- [3] *Driverside*. <http://www.driverside.com/>
- [4] *foodspotting*. <http://www.foodspotting.com>
- [5] *Tallerator*. <http://www.tallerator.es/>
- [6] *Google maps, api v3*. <https://developers.google.com/maps/articles/phpsqlajax?hl=es>
- [7] *infobubble (customizar bocadillos del mapa)*. <http://google-maps-utility-library-v3.googlecode.com/svn/trunk/infobubble/examples/example.html>
- [8] *codigo .js twittear*. <https://dev.twitter.com/docs/tweet-button>
- [9] *manual php*. <http://php.net/manual/es/index.php>
- [10] *manual jquery*. <http://librojquery.com/>
- [11] *bootstrap de twitter para los css*. <http://twitter.github.com/bootstrap/>
- [12] *Cliente servidor*. <http://es.wikipedia.org/wiki/Cliente-servidor>
- [13] *Página del proyecto moskitt*. <http://www.moskitt.org/>
- [14] *Workbench de MySQL*. <http://www.mysql.com/products/workbench/>