



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicación web para gestión integral de talleres de vehículos.


Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Inmaculada María Canet Ferrá

Tutor: Vicent Lorente Garcés

2015/2016



Aplicación web para gestión integral de talleres de vehículos.

Resumen

El presente proyecto pretende desarrollar un sistema de gestión, mediante una aplicación web dinámica, encargado de toda la actividad relacionada con las reparaciones realizadas en un taller mecánico de vehículos, de manera que la aplicación desarrollada para llevar a cabo esta gestión tiene dos partes:

- Una parte para la empresa: donde el administrador, dueño o encargado del taller puede llevar un registro y seguimiento de las reparaciones y revisiones de los vehículos, así como dar de alta nuevos clientes y vehículos y gestionar los antiguos. De esta manera, gestiona todos sus clientes, los vehículos que pasan por su taller, así como las reparaciones asociados a éstos simplificando la gestión del negocio.
- Otra parte para el cliente: donde un cliente, previamente dado de alta por el administrador mencionado en la parte de la empresa, puede acceder a la información personalizada de sus vehículos y de las reparaciones hechas a éstos. De esta manera, el cliente tiene información en tiempo real de las reparaciones de los vehículos que se encuentran actualmente en el taller, así como un historial de todos los vehículos y sus reparaciones.

El acceso a estos dos módulos se realizará a través de una interfaz común desarrollada en una página Web pública.

Se trata de una aplicación web dinámica que permite el acceso a la aplicación desde cualquier navegador, lo que la hace especialmente adecuada para talleres que realizan reparaciones en ruta y útil y cómoda para todos los clientes del taller, ya que pueden acceder a la web con sólo disponer de una conexión a Internet, sin necesidad de instalar ningún software adicional. Además, las actualizaciones se realizan del lado del servidor, lo que no afecta al sistema operativo del lado del cliente.

La aplicación constará de una base de datos y una página web dinámica en HTML que realiza consultas SQL a una base de datos por medio de código PHP. El código PHP estará incrustado dentro del HTML interactuando con el mismo, lo que permite diseñar la página web con un editor común de HTML y añadir el código dinámico PHP usando las etiquetas indicadas para ello.

El lenguaje PHP está especialmente indicado para el desarrollo de aplicaciones web dinámicas con acceso a información que está almacenada en una base de datos, de esta manera, será la propia aplicación web la encargada de comunicarse con la base de datos, en la que se consultará, modificará o eliminará la información relacionada con la transacción realizada, tanto de la gestión de clientes como de la gestión de vehículos y reparaciones. Cabe resaltar que el propio código en PHP es invisible al cliente, dando lugar a una programación segura.

Cabe destacar la mejora y utilidad real que esta aplicación proporcionará tanto al taller de vehículos como a los clientes, al taller dada su eficiencia en la gestión y facilidad de uso y a los clientes por su comodidad, ya que podrán ver desde su casa sus vehículos y las reparaciones asociadas a éstos sin tener que desplazarse al taller y ni siquiera llamar por teléfono.

Palabras clave: Aplicación Web dinámica, PHP, HTML, servidor, cliente.



Abstract

This project pretends to develop a management system through a dynamic web application. This system is the responsible for the whole activity related to the mechanical repairs in a car repair shop, so that the developed application will have two different parts:

- One part is for the company, in which the administrator, the owner o the person who is in charge of the car repair shop can create, edit and manage the costumers, vehicles and their mechanical repairs. In this way, this application simplifies the business management.
- The other part is for the customer, where a customer, who has to be previously created by the administrator, can access to his personal information of his vehicles and repairs made to them. In this way, the customer obtains useful information in real time for the repairs of the vehicles currently in the car repair shop, as well as a history of all his vehicles and repairs.

The access to these two modules is done through a common interface, which is developed on a public Web page.

This dynamic web application provides access to the application from any browser, making it especially suitable for car repair shops that perform repairs in route and it is also useful and convenient for all customers of the car repair shop, as they can access to the web with a simple Internet connection, without installing any additional software. In addition, updates are made on the server side, which does not affect the operating system on the client side.

The application will consist of a database and a dynamic HTML web page that performs SQL queries to a database using PHP code. The PHP code is embedded in HTML and it is interacting with it, allowing the web designer to create the website with a common HTML editor and the dynamic PHP code is added using the labels given to it.

The PHP language is especially suitable for the development of dynamic web applications that access to stored information in a database, so the web application itself will be the responsible for the communication with the database, in which the information related to the current transaction will be consulted, modified or deleted, both for the customer management and for the management of vehicles and repairs. It is also remarkable that the PHP code itself is invisible to the customer, resulting in a secure programming.

Note the improvement and real utility that this application will provide both the car repair shop and costumers, to the car repair shop because of its efficient management and ease of use and to the costumers for their convenience because they can see their vehicles and repairs from their home without going to the car repair shop or doing any phone call.

Keywords: Dynamic Web Application, PHP, HTML, server, client.

Tabla de contenidos

1. Introducción	11
1.1- Motivación	11
1.2.- Descripción	11
1.2.1.- Entorno de aplicación	13
1.2.2.-Ventajas / Beneficios	13
1.2.3.- Inconvenientes	13
1.3.- Objetivos	13
1.4.- Descripción del documento.....	14
2. Planificación temprana.....	15
3. Estudio de viabilidad.....	17
3.1.- Económico	17
3.2.- Técnico.....	17
3.3.- Legal	18
3.4.- Operacional.....	19
3.5.- Conclusión del estudio.....	19
4. Especificación de requisitos software.....	21
4.1.- Introducción	21
4.1.1.- Naturaleza y propósito	21
4.1.2.- Entorno.....	22
4.1.3.- Ámbito	22
4.2.- Descripción general	22
4.2.1.- Perspectiva del producto	22
4.2.2.- Funciones del producto	23
4.2.3.- Características del usuario	23
4.2.4.- Restricciones generales.....	24
4.2.5.- Supuestos y dependencias.....	24
4.3.- Requisitos específicos.....	24
4.3.1.- Requisitos de interfaces externos.....	24



4.3.1.1.- Interfaces de usuario	24
4.3.1.2.- Interfaces Hardware	25
4.3.1.3.- Interfaces Software	25
4.3.1.4.- Interfaces de comunicaciones	25
4.3.2.- Requisitos funcionales	25
4.3.2.1- Autenticación del usuario	25
4.3.2.2- Módulo Cliente	26
4.3.2.3- Módulo Administrador.....	26
4.3.3.- Requerimientos de eficiencia y rendimiento.....	29
4.3.4.- Restricciones de diseño	30
4.3.5.- Atributos	30
4.4.- Definiciones, acrónimos y abreviaturas	30
4.5.- Referencias.....	31
5. Análisis	33
5.1.- Casos de uso.....	33
5.1.1.- Actor Anónimo	34
5.1.2.- Actor Cliente	35
5.1.3.- Actor Administrador	36
5.2.- Diagrama de clases	38
5.3.- Diagramas de secuencia.....	40
5.3.1.- Autenticación del usuario	40
5.3.2.- Módulo Cliente	41
5.3.3.- Módulo Administrador.....	43
6. Planificación	51
7. Diseño	55
7.1.- Arquitectura de tres capas.....	55
7.1.1.- Capa de presentación	56
7.1.2.- Capa de lógica de negocio	59
7.1.3.- Capa de persistencia.....	60

8. Prototipo.....	61
9. Implementación.....	63
9.1.- Tecnologías	63
9.2.- Funcionalidades y descripción de la implementación.....	63
10. Pruebas.....	67
10.1.- Prueba de los módulos	67
10.2.- Prueba de validación de enlaces.....	72
10.3.- Prueba de navegadores.....	72
10.4.- Prueba de seguridad de acceso.....	74
11. Plazos y coste real.....	75
12. Conclusiones.....	79
12.1. Trabajo realizado	79
12.2. Valoración personal.....	79
12.3. Futuras ampliaciones.....	80
Referencias	81



Índice de ilustraciones

Ilustración 2.1: Planificación temprana. Fases y tareas.....	15
Ilustración 2.2: Planificación temprana. Diagrama de Gantt.	16
Ilustración 2.3: Planificación temprana. Costes estimados.	16
Ilustración 5.1: Diagrama de Uso: Actor anónimo. Navegar por VICARMECANICS S.L.	34
Ilustración 5.2: Diagrama Casos de Uso: Actor cliente. Listar vehículos y listar reparaciones.	35
Ilustración 5.3: Diagrama Casos de Uso: Actor administrador. Gestión de clientes.	36
Ilustración 5.4: Diagrama Casos de Uso: Actor administrador. Gestión de vehículos.	37
Ilustración 5.5: Diagrama Casos de Uso: Actor administrador. Gestión de reparaciones.	37
Ilustración 5.6: Diagrama de clases.....	38
Ilustración 5.7: Diagrama de secuencia: Autenticación usuario.....	40
Ilustración 5.8: Diagrama de secuencia: Listado de vehículos del cliente.	41
Ilustración 5.9: Diagrama de secuencia: Listado de reparaciones del cliente.....	42
Ilustración 5.10: Diagrama de secuencia: Listado de clientes.....	43
Ilustración 5.11: Diagrama de secuencia: Crear nuevo cliente.	44
Ilustración 5.12: Diagrama de secuencia: Modificar cliente.....	44
Ilustración 5.13: Diagrama de secuencia: Eliminar cliente.....	45
Ilustración 5.14: Diagrama de secuencia: Listado de vehículos.....	46
Ilustración 5.15: Diagrama de secuencia: Crear nuevo vehículo.	46
Ilustración 5.16: Diagrama de secuencia: Modificar vehículo.....	47
Ilustración 5.17: Diagrama de secuencia: Eliminar vehículo.....	47
Ilustración 5.18: Diagrama de secuencia: Listado de reparaciones.	48
Ilustración 5.19: Diagrama de secuencia: Crear nueva reparación.....	48
Ilustración 5.20: Diagrama de secuencia: Modificar reparación.	49
Ilustración 5.21: Diagrama de secuencia: Eliminar reparación.....	49
Ilustración 6.1: Planificación después del análisis. Fases y tareas.	52
Ilustración 6.2: Planificación después del análisis. Diagrama de Gantt.....	52
Ilustración 6.3: Planificación después del análisis. Costes.	53

Ilustración 7.1: Arquitectura de tres capas.	55
Ilustración 7.2: Interfaz parte pública I.	56
Ilustración 7.3: Interfaz parte pública II.	57
Ilustración 7.4: Interfaz parte pública III.	57
Ilustración 7.5: Interfaz parte privada cliente.	58
Ilustración 7.6: Interfaz parte privada administrador.	58
Ilustración 7.7: Módulos de la aplicación.	59
Ilustración 7.8: Capa de persistencia.	60
Ilustración 9.1: Implementación parte privada del cliente.	64
Ilustración 9.2: Implementación parte privada del administrador.	66
Ilustración 10.1: Prueba de validación de enlaces.	72
Ilustración 10.2: Navegador Mozilla Firefox.	73
Ilustración 10.3: Navegador Google Chrome.	73
Ilustración 10.4: Navegador Safari.	74
Ilustración 10.5: Validación de usuario.	74
Ilustración 11.1: Plazo real. Fases y tareas.	76
Ilustración 11.2: Plazo real. Diagrama de Gantt.	76
Ilustración 11.3: Coste real del proyecto.	77

Aplicación web para gestión integral de talleres de vehículos.

1. Introducción

1.1- Motivación

Al estudiar las herramientas y técnicas utilizadas para la gestión de un taller de vehículos, nos encontramos con aplicaciones poco eficaces como hojas de cálculo, bloc de notas y otras técnicas poco recomendables, por lo que se decide llevar a cabo una informatización de dicha gestión permitiendo también la interacción con el usuario mediante una aplicación Web que gestionará toda esta información de manera eficaz, sencilla y cómoda.

Esto nos permitirá tener todos los elementos de nuestro negocio organizados y visibles de manera que podamos encontrar la información requerida en el momento deseado. Mediante las herramientas utilizadas mantendremos toda esta información organizada dentro de un SGBD (Sistema Gestor de Base de Datos).

Por otra parte, hoy en día, los clientes de un taller de vehículos no tienen ninguna información de sus vehículos ni de las reparaciones realizadas hasta que van a recoger los vehículos al taller, encontrándose con la probable sorpresa de una reparación mayor a la prevista y por supuesto, un aumento del coste de la misma. Mediante esta aplicación Web, el usuario podrá acceder en tiempo real al estado de sus vehículos y las reparaciones realizadas, de manera que puede conocer al momento qué reparación ha realizado el taller y si su vehículo está listo para ser recogido o, si por el contrario, se necesita su aprobación o consulta para una reparación mayor. De esta manera se elimina el sentimiento de incertidumbre que rodea a los clientes de un taller mecánico.

1.2.- Descripción

El presente proyecto pretende desarrollar un sistema de gestión, mediante una aplicación web dinámica, encargado de toda la actividad relacionada con las reparaciones realizadas en un taller mecánico de vehículos, de manera que la aplicación desarrollada para llevar a cabo esta gestión tiene dos partes:

- Parte pública: Constará de una página Web en la que el usuario podrá visualizar todos los contenidos publicados. Contendrá información sobre la empresa, información sobre los

Aplicación web para gestión integral de talleres de vehículos.

servicios que ofrece, una galería de imágenes del taller e información de contacto, así como un mapa de localización.

Además, dispondrá de dos enlaces: uno para que el cliente registrado pueda acceder a la parte privada de clientes, y otro, para que el administrador del taller pueda acceder a la parte privada de gestión del taller.

- Parte privada: esta parte, a la que se accederá con usuario y contraseña, consta de dos módulos:
 - módulo del cliente: donde un cliente puede acceder a la información personalizada de sus vehículos y de las reparaciones hechas a éstos. De esta manera, el cliente tiene información en tiempo real de las reparaciones de los vehículos que se encuentran actualmente en el taller, así como un historial de todos los vehículos y las reparaciones realizadas.
 - módulo del administrador del taller: donde el administrador del taller tendrá acceso a gestionar los datos referentes a los clientes, vehículos y reparaciones del taller mecánico, pudiendo en cada momento añadir, actualizar o eliminar datos referentes a cada uno de ellos, simplificando así la gestión del negocio.

Se trata de una aplicación web dinámica que permite el acceso a la aplicación desde cualquier navegador, lo que la hace especialmente adecuada para talleres que realizan reparaciones en ruta y útil y cómoda para todos los clientes del taller, ya que pueden acceder a la web con sólo disponer de una conexión a Internet, sin necesidad de instalar ningún software adicional.

Cabe destacar la mejora y utilidad real que esta aplicación proporcionará tanto al taller de vehículos como a los clientes, al taller dada su eficiencia en la gestión y facilidad de uso y a los clientes por su comodidad ya que podrán ver desde su casa sus vehículos y las reparaciones asociadas a éstos sin tener que desplazarse al taller y ni siquiera llamar por teléfono.

Por otro lado, será la propia aplicación Web la encargada de comunicarse con la base de datos, en la que se consultará, modificará o borrará la información referente a las transacciones realizadas en cada momento. Además, las actualizaciones se realizan del lado del servidor, y dado que se trata de una aplicación cliente-servidor, estas actualizaciones no afectarán al sistema operativo del lado del cliente.

1.2.1.- Entorno de aplicación

Se considera entorno de aplicación a los diversos ámbitos en los que puede ser utilizada la aplicación, en este caso, esta aplicación será utilizada en el ámbito empresarial, bien en la propia empresa para la cual ha sido desarrollada la aplicación Web o para cualquier taller de vehículos.

1.2.2.-Ventajas / Beneficios

- Mejora de la gestión del taller de vehículos.
- Simplicidad en la gestión del negocio.
- Eficiencia, organización y comodidad.
- Portabilidad, ya que solo se necesita una conexión a Internet.
- Actualizaciones del lado del servidor.
- Interactividad con el usuario.
- Información en tiempo real.
- No discriminación del Sistema Operativo del lado del cliente.

1.2.3.- Inconvenientes

- Estado del servidor, ya que si el servidor cae no podrá servir las páginas y demandas del cliente y/o administrador.
- Espacio de almacenamiento de información limitado.
- Seguridad de los datos sujeta al servidor donde se aloje.
- Estabilidad de la aplicación dependiente del número de visitas en un momento dado.

1.3.- Objetivos

El principal objetivo del proyecto que nos atañe es diseñar e implementar una aplicación Web dinámica siguiendo los requisitos del cliente, en este caso, del taller mecánico de vehículos Vicarmecanics S.L.

Siguiendo el ciclo de vida del desarrollo de una aplicación [1], se empezará por la descripción del proyecto, la enumeración de sus objetivos y la realización de una planificación temprana y un estudio de viabilidad, a lo que seguirá una especificación de requisitos y un análisis del modelo conceptual. Con todo esto, se puede ya llevar a cabo una planificación más

detallada para empezar con el diseño, la creación de un prototipo y la implementación de la aplicación Web. Una vez desarrollado, se realizarán las pruebas oportunas para comprobar su correcto funcionamiento, una valoración de los plazos y coste real, así como una conclusión final del proyecto.

1.4.- Descripción del documento

El presente documento describe todas y cada una de las fases del ciclo de vida del desarrollo de una aplicación Web.

En el capítulo 1 se describe el proyecto, en el capítulo 2 se realiza una planificación temprana del mismo y en el capítulo 3 se explica el estudio de viabilidad llevado a cabo. A continuación, en el capítulo 4 se muestra la especificación de requisitos, en el capítulo 5 el análisis del modelo conceptual y en el capítulo 6 se puede observar una planificación más exhaustiva del proyecto. El diseño de la aplicación se detalla en el capítulo 7, el prototipo realizado se describe en el capítulo 8 y en el capítulo 9 se muestra la implementación llevada a cabo. Finalmente, en el capítulo 10 se especifican las pruebas realizadas a la aplicación, en el capítulo 11 se valoran los plazos y el coste real del mismo, y por último, en el capítulo 12 se muestran las conclusiones.

2. Planificación temprana

El primer paso a realizar en el desarrollo de una aplicación es la planificación. En el proyecto que nos ocupa se pretende desarrollar una aplicación de alta calidad de una manera correcta y oportuna.

En el punto en el que nos encontramos se lleva a cabo una planificación inicial o temprana para poder valorar, de manera general, los tiempos necesarios para el desarrollo de la aplicación, así como el presupuesto inicial.

A continuación, en la ilustración numerada como 2.1. se incluye una especificación de los trabajos a realizar, que según esta planificación temprana tendrán una duración total de 46 días. En la ilustración 2.2. se muestra el Diagrama de Gantt, en el que se puede observar de manera gráfica la consecución de las tareas y cómo éstas se llevarán a cabo de manera secuencial.

Task Name	Duration	Start	Finish
PLANIFICACIÓN TEMPRANA	46 days	Wed 20/01/16	Tue 29/03/16
1 Especificación de requisitos	15 days	Wed 20/01/16	Tue 09/02/16
2 Modelado conceptual	9 days	Wed 10/02/16	Mon 22/02/16
3 Codificación	15 days	Tue 23/02/16	Mon 14/03/16
4 Pruebas	7 days	Tue 15/03/16	Tue 29/03/16

Ilustración 2.1: Planificación temprana. Fases y tareas.

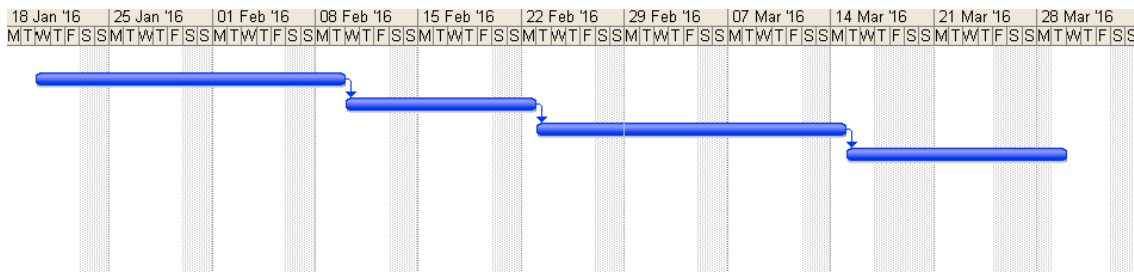


Ilustración 2.2: Planificación temprana. Diagrama de Gantt.

Según esta primera planificación, y tal y como podemos ver en la ilustración 2.3, el tiempo de trabajo necesario para desarrollar este proyecto será de 46 días de trabajo con un presupuesto inicial de 3.680 €.

Project Statistics for 'planificación temprana'			
	Start		Finish
Current	Wed 20/01/16		Tue 29/03/16
Baseline	NA		NA
Actual	NA		NA
Variance	0d		0d
	Duration	Work	Cost
Current	46d	368h	3.680,00 €
Baseline	0d?	0h	0,00 €
Actual	0d	0h	0,00 €
Remaining	46d	368h	3.680,00 €
Percent complete:			
Duration: 0%		Work: 0%	
			Close

Ilustración 2.3: Planificación temprana. Costes estimados.

3. Estudio de viabilidad

Todo proyecto es realizable con recursos ilimitados y un tiempo infinito, pero, como esto no es real, deberemos realizar un estudio de viabilidad antes de comenzar con el desarrollo del mismo, consistente en estudiar y evaluar la viabilidad o posibilidad de realizar el proyecto que nos atañe.

Para ello, se estudiarán los siguientes cuatro aspectos: económico, técnico, legal y operacional.

3.1.- Económico

En el estudio económico, se evaluará cuan de positiva es la relación costes/beneficios.

A continuación, se desglosan los diferentes gastos surgidos de la realización del proyecto:

- Editor HTML y PHP de libre distribución: 0€
- Costes de desarrollo: Según la planificación temprana, mostrada en el punto anterior, las horas de trabajo que se dedicarán a este proyecto son 368h, y los costes asociados son 3.680 €.
- Gastos de alojamiento: 0€, debido a que se utilizará la actual conexión a Internet que el cliente ya tiene contratada para dar acceso a la aplicación Web desarrollada.

En cuanto a los beneficios obtenidos del desarrollo y puesta en marcha de la aplicación Web, aunque se evalúan por parte del cliente, podemos encontrar: mayor simplicidad en la gestión del negocio, reducción y eliminación de errores y un aumento en la fiabilidad del sistema.

3.2.- Técnico

En el estudio técnico, se estudiará si existen y están a nuestro alcance las herramientas necesarias para el desarrollo del proyecto.

Las herramientas que se van a utilizar, dentro de todas las disponibles, para el desarrollo de la aplicación Web son:

HTML: HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje para la elaboración de páginas web. Es un estándar que define una estructura básica y sirve de referencia que conecta con la elaboración de páginas web en sus diferentes versiones. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW).

PHP: Hypertext Pre-processor (PHP) es una de las tecnologías Web más extendida en la actualidad, fácil de aprender y puede alojarse en cualquier servidor. El código fuente es abierto.

SQLite: Sistema de gestión de bases de datos. Es un ligero motor de bases de datos de código abierto, que se caracteriza por mantener el almacenamiento de información persistente de forma sencilla.

Finalmente, el propósito es brindarle al usuario una manera más sencilla de realizar la navegación.

3.3.- Legal

En el estudio legal, se evaluarán los requisitos legales para la creación de una aplicación web:

En primer lugar, respecto a la LOPD [2]: Se diferencia entre el responsable del fichero y el encargado del fichero. El taller de vehículos que va a utilizar la aplicación deberá cumplir las obligaciones que le correspondan como responsable del fichero respecto a los datos personales de los clientes (nombre, dirección, teléfono...)[3]. El desarrollador de la aplicación tiene acceso a los datos, y por tanto, se le considera, a efectos de protección de datos como encargado del fichero, y sus obligaciones son las del artículo 12 de la LOPD [4].

En segundo lugar, respecto a la propiedad intelectual e industrial, se debe incluir la cláusula del copyright, el año y el nombre de la empresa en un contrato [5] en el que, además del encargo, los plazos y el precio, deberá constar la cesión de los derechos de propiedad intelectual e industrial que, como desarrollador de la aplicación, se ceden al taller de vehículos.

3.4.- Operacional

En el estudio operacional, se valorará si podrá implantarse el proyecto de manera efectiva considerando el equipo de desarrollo. En este caso, se poseen los suficientes conocimientos técnicos para llevar a cabo el trabajo técnico requerido y necesario para desarrollar la aplicación deseada.

3.5.- Conclusión del estudio

Una vez analizados todos los puntos del estudio de viabilidad, se puede afirmar que se estima el proyecto como viable, ya que los gastos, inicialmente afrontados por el desarrollador, se van a subsanar con la venta del producto, las herramientas necesarias para el desarrollo del proyecto existen y están disponibles, legalmente se puede desarrollar el proyecto y se tiene suficiente capacidad operacional para llevarlo a cabo.

Aplicación web para gestión integral de talleres de vehículos.

4. Especificación de requisitos software

4.1.- Introducción

La Especificación de Requisitos Software, en adelante ERS, es una especificación de un producto software, programa o conjunto de programas concretos que realizan ciertas funciones en un entorno específico [6].

Los objetivos de una buena ERS son principalmente los siguientes:

- Ayudar a los clientes a describir claramente lo que se desea obtener mediante un cierto software.
- Ayudar a los desarrolladores a entender qué quiere exactamente el cliente.
- Servir de base para desarrollar los estándares de ERS particulares para cada organización.

Una buena ERS proporciona las siguientes ventajas:

- Contrato cliente-desarrollador.
- Reducción del esfuerzo de desarrollo.
- Base para la estimación de costes y planificación.
- Punto de referencia para procesos de verificación y validación.
- Base para posibles mejoras.

4.1.1.- Naturaleza y propósito

Cada producto software es diferente porque se construye para cumplir los requisitos únicos de un cliente. Éste es el caso de la aplicación Web que nos atañe, ya que es un producto a medida que parte desde cero.

El propósito de la ERS que nos ocupa es definir cuáles son los requerimientos que debe tener una aplicación Web que gestiona toda la actividad relacionada con los clientes, vehículos y reparaciones que pasan por un taller mecánico de vehículos, además de la gestión de una página principal de la propia empresa y un área de clientes privada.

Para la elaboración de esta ERS se han seguido las recomendaciones del estándar ANSI std IEEE 830-1998 [7] [8], uno de los estándares más referenciados para la especificación de requisitos en proyectos Web.

4.1.2.- Entorno

La ERS aquí presentada forma parte de la documentación asociada al software de la aplicación web que se realiza. Se definen correctamente en esta especificación todos los requisitos software, no describiendo ningún detalle de diseño o implementación, verificación, gestión del proyecto, plan de calidad, etc.

4.1.3.- Ámbito

El producto deberá ofrecer las operaciones a realizar en las bases de datos de la herramienta tras una transacción. Estas transacciones pueden desarrollarse dentro de los dos módulos principales de la parte privada de la aplicación Web: el módulo del cliente y el módulo del administrador del taller. La aplicación también consta de una página web pública, pero dado que no conlleva transacciones en la base de datos, no se describe en este apartado.

Esta parte privada de la aplicación Web podrá ser utilizada por cualquier persona autorizada que esté registrada en la misma, tanto por el administrador del taller como por los diferentes clientes autorizados.

4.2.- Descripción general

4.2.1.- Perspectiva del producto

Este producto pretende servir de utilidad a cualquier cliente y administrador del taller de vehículos.

Se trata de una aplicación completamente independiente destinada a ejecutarse sobre cualquier navegador. Esta aplicación no es un componente de un sistema más amplio.

4.2.2.- Funciones del producto

La función principal de la aplicación a desarrollar se centra en la automatización de la información requerida en la gestión de clientes, vehículos y reparaciones, además de ofrecer información a los clientes sobre los vehículos que han pasado por el taller o que actualmente están siendo reparados, con lo que obtienen información en tiempo real sobre sus vehículos y las reparaciones realizadas.

Esta parte privada consiste en dos módulos: el módulo del cliente, que mostrará la información sobre los vehículos y reparaciones del cliente, y el módulo del administrador del taller de vehículos, responsable del mantenimiento de la información relacionada con los clientes, vehículos y reparaciones.

Funcionalidad del primer módulo (cliente):

- Consultas sobre los vehículos.
- Consultas sobre las reparaciones.

Funcionalidad del segundo módulo (administrador del taller):

- Altas, bajas, modificaciones y consultas sobre los clientes.
- Altas, bajas, modificaciones y consultas sobre los vehículos.
- Altas, bajas, modificaciones y consultas sobre las reparaciones.

El acceso a estos dos módulos se realizará a través de una interfaz común desarrollada en una página Web pública, en la que además, el taller de vehículos podrá incluir información sobre la empresa, una descripción de los servicios que ofrece, una galería de imágenes de su taller y los datos de contacto, así como un mapa de localización.

4.2.3.- Características del usuario

La interfaz de la página principal, así como la interfaz de los dos módulos descritos (cliente y administrador), son muy sencillas e intuitivas, con lo que no requieren ningún conocimiento previo por parte del usuario para realizar un uso correcto.

4.2.4.- Restricciones generales

La aplicación Web se realizará con HTML y PHP. No se necesitará un hardware especialmente potente para el uso de esta aplicación, aunque sí se aconseja una conexión a Internet adecuada.

Para poder realizar las transacciones en la aplicación Web los usuarios, tanto clientes como administradores, deben identificarse con una contraseña personal, que la aplicación Web validará para poder acceder a la misma.

Esta aplicación necesita un nivel medio de seguridad, ya que los datos que se manejan no están especialmente protegidos por la LOPD. Se realizarán copias de seguridad periódicamente de la Base de Datos.

4.2.5.- Supuestos y dependencias

El portal web requiere que los usuarios utilicen un navegador web para acceder a ella. No será necesario el uso de un navegador específico, pero estos navegadores deberán ser compatibles con el protocolo HTTP.

El servidor donde se hospede nuestro sistema deberá admitir:

- Ejecución de código PHP.
- Procesos transaccionales con bases de datos.

4.3.- Requisitos específicos

4.3.1.- Requisitos de interfaces externos

4.3.1.1.- Interfaces de usuario

La interfaz de usuario de Vicarmecanics S.L. es sencilla, intuitiva y amigable.

A simple vista el usuario puede ver todas las funciones que pueden realizar dentro de la página a través de los menús.

4.3.1.2.- Interfaces Hardware

Los interfaces hardware utilizados son todos aquellos dispositivos electrónicos con conexión a Internet, ya sea inalámbrica o cableada. Entre otros podemos hablar de ordenadores, bien portátiles o de sobremesa, tabletas, teléfonos móviles o incluso televisiones con acceso a Internet.

4.3.1.3.- Interfaces Software

Para la correcta ejecución de nuestra aplicación Web solamente es necesario un navegador, de los muchos disponibles hoy en día, trabajando sobre cualquier Sistema Operativo.

4.3.1.4.- Interfaces de comunicaciones

Como requisito indispensable podríamos destacar un dispositivo de acceso a Internet, ya sea una tarjeta de red interna, externa o inalámbrica.

4.3.2.- Requisitos funcionales

4.3.2.1- Autenticación del usuario

➤ Comprobar usuario y contraseña:

Objetivo: verificación del usuario, ya sea cliente o administrador, y la contraseña.

Entradas: introducción del nombre de usuario y la contraseña de acceso.

Proceso: comprobación de los datos de acceso.

Salida: si la validación es correcta, se accede al área privada, bien del cliente o bien del administrador, según usuario.

4.3.2.2- Módulo Cliente

➤ Listar vehículos:

Objetivo: listar todos los vehículos de un cliente determinado.

Entradas: no existen entradas.

Proceso: se listarán todos los vehículos asociados al cliente.

Salida: se mostrará por pantalla una tabla con todos los campos de los vehículos de un cliente determinado.

➤ Listar reparaciones:

Objetivo: listar todas las reparaciones de todos los vehículos de un cliente determinado.

Entradas: no existen entradas.

Proceso: se listarán todas las reparaciones de todos los vehículos asociados al cliente.

Salida: se mostrará por pantalla una tabla con los campos de las reparaciones de los vehículos de un cliente determinado.

4.3.2.3- Módulo Administrador

- Funciones respecto a los clientes:

➤ Listar clientes:

Objetivo: listar todos los clientes de la Base de Datos.

Entradas: no existen entradas.

Proceso: se listarán todos los clientes de la tabla Clientes de la Base de Datos.

Salida: se mostrará por pantalla una tabla con todos los atributos de los clientes existentes.

➤ Añadir un nuevo cliente:

Objetivo: alta de un cliente en la base de datos.

Entradas: introducción del CIF del cliente, el nombre, la dirección, el email, la persona de contacto y dos teléfonos.

Proceso: creación de una nueva tupla con los campos obtenidos del formulario en la tabla correspondiente de la Base de Datos.

Salida: Base de Datos actualizada con los nuevos datos.

➤ **Modificar un cliente:**

Objetivo: acceso a la Base de Datos y modificación de los atributos que se deseen de un cliente determinado.

Entradas: una vez filtrado el cliente a modificar, introducción de los nuevos valores del CIF del cliente, el nombre, la dirección, el email, la persona de contacto o los dos teléfonos.

Proceso: cambio de un atributo o de varios atributos de un cliente determinado por otros nuevos.

Salida: Base de Datos actualizada con los nuevos datos.

➤ **Eliminar un cliente:**

Objetivo: eliminación de un cliente concreto de la Base de Datos de clientes.

Entradas: se seleccionará el nombre del cliente a eliminar.

Proceso: eliminación del cliente de la Base de Datos.

Salida: aviso en pantalla de que la transacción se ha realizado con éxito.

- **Funciones respecto a los vehículos:**

➤ **Listar vehículos:**

Objetivo: listar todos los vehículos de la Base de Datos.

Entradas: no existen entradas.

Proceso: se listarán todos los atributos de los vehículos de la tabla Vehículos de la Base de Datos, así como su propietario.

Salida: se mostrará por pantalla una tabla con todos los vehículos existentes y sus atributos.

➤ **Añadir un nuevo vehículo:**

Objetivo: alta de un vehículo en la base de datos.

Entradas: una vez filtrado el propietario del vehículo, se introducirá la matrícula, el número de

bastidor, el tipo del vehículo, la marca y el modelo.

Proceso: creación de una nueva tupla con los campos obtenidos del formulario en la tabla correspondiente de la Base de Datos.

Salida: Base de Datos actualizada con los nuevos datos.

➤ **Modificar un vehículo:**

Objetivo: acceso a la Base de Datos y modificación de los atributos que se deseen de un vehículo determinado.

Entradas: una vez filtrado el propietario del vehículo y el vehículo a modificar, se introducirán los nuevos valores de la matrícula, el número de bastidor, el tipo del vehículo, la marca o el modelo.

Proceso: cambio de un atributo o de varios atributos de un vehículo determinado por otros nuevos.

Salida: Base de Datos actualizada con los datos modificados.

➤ **Eliminar un vehículo:**

Objetivo: eliminación de la Base de Datos de vehículos de un vehículo concreto.

Entradas: una vez filtrado el propietario del vehículo, se seleccionará la matrícula del vehículo a eliminar.

Proceso: eliminación del vehículo de la Base de Datos.

Salida: aviso en pantalla de que la transacción se ha realizado con éxito.

- **Funciones respecto a las reparaciones:**

➤ **Listar reparaciones:**

Objetivo: listar todas las reparaciones de la Base de Datos.

Entradas: no existen entradas.

Proceso: se listarán todos los atributos de las reparaciones de la tabla Reparaciones de la Base de Datos, así como la matrícula asociada.

Salida: se mostrará por pantalla una tabla con todas las reparaciones existentes y sus atributos.

➤ **Añadir una nueva reparación:**

Objetivo: alta de una reparación en la base de datos.

Entradas: una vez filtrada la matrícula del vehículo asociado, se introducirá la fecha de la reparación, los kilómetros a los que se realiza la reparación y una descripción de la misma.

Proceso: creación de una nueva tupla con los campos obtenidos del formulario en la tabla correspondiente de la Base de Datos.

Salida: Base de Datos actualizada con los nuevos datos.

➤ **Modificar una reparación:**

Objetivo: acceso a la Base de Datos y modificación de los atributos que se deseen de una reparación determinada.

Entradas: una vez filtrado el propietario del vehículo, el vehículo al que se refiere la reparación y la reparación , introducción de los nuevos valores de la fecha de la reparación, los kilómetros a los que se realiza la reparación o de la descripción de la misma.

Proceso: cambio de un atributo o de varios atributos de una reparación determinada por otros nuevos.

Salida: Base de Datos actualizada con los nuevos datos.

➤ **Eliminar una reparación:**

Objetivo: eliminación de la Base de Datos de una reparación concreta.

Entradas: una vez filtrado el propietario del vehículo y el vehículo asociado a la reparación, se seleccionará la reparación a eliminar.

Proceso: eliminación de la reparación de la Base de Datos.

Salida: aviso en pantalla de que la transacción se ha realizado con éxito.

4.3.3.- Requerimientos de eficiencia y rendimiento

La aplicación Web no exige demasiados requerimientos en esta materia. En todo caso, conviene cuidar la realización de transacciones de la forma más automatizada y organizada posible.

Teniendo en cuenta, además, que la base de datos de un taller de vehículos no suele crecer en más de 1000 tuplas al año, se puede afirmar que el sistema gestor de base de datos SQLite es un buen sistema para llevar a cabo la gestión de los datos.

4.3.4.- Restricciones de diseño

No se requieren especiales restricciones, ya que el producto cumple las reglas del Sistema Operativo sobre el que se ejecutará y se garantiza así que la aplicación Web se comporta correctamente.

4.3.5.- Atributos

Con respecto a la seguridad, se han tomado las precauciones oportunas a la hora de transmitir datos a través de internet, como ejemplo tomamos el envío de la contraseña del administrador de la aplicación Web.

El administrador, para acceder, debe enviar el usuario y la contraseña a través de Internet. Es en este paso cuando la información podría quedar desprotegida. Para protegerla, se encriptará la contraseña.

En cuanto al mantenimiento de la aplicación Web, vendrá dado por los posibles problemas que puedan ir surgiendo a lo largo de la vida de la aplicación. Cabe destacar que en un futuro, se piensa en implementar mejoras.

4.4.- Definiciones, acrónimos y abreviaturas

- Autenticación o autenticación: proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse.
- Base de Datos: Una base de datos es una colección organizada de información. Ésta contiene una colección de registros que puede buscar, ordenar y analizar rápidamente.
- SGBD: Sistema de Gestión de Bases de Datos.
- SQLite: Sistema para la creación y gestión SQL de la base de datos.
- Hardware: conjunto de elementos materiales que conforman una computadora.

- HTML (HyperText Markup Language): el lenguaje de autor usado para crear documentos en la World Wide Web.
- IEEE (Institute of Electrical and Electronics Engineers): Asociación de profesionales norteamericanos que aporta criterios de estandarización de dispositivos eléctricos y electrónicos.
- Interfaz: Parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario.
- Navegador: Programa “Navegador” usado para ver e interactuar con varios tipos de recursos de Internet disponibles en la World Wide Web.
- PHP (Hypertext Pre-processor): Lenguaje de programación originalmente diseñado para el desarrollo web de contenido dinámico.
- Tupla: Una tupla de la base de datos es una descripción de un artículo en particular que se almacena en la base de datos. En una base de datos relacional, cada fila de cada tabla es una tupla o registro.
- Sistema Operativo (SO): Un sistema operativo es un software de sistema, es decir, un conjunto de programas de computadora destinado a permitir una administración eficaz de sus recursos. Comienza a trabajar cuando se enciende la computadora y gestiona el hardware de la máquina desde los niveles más básicos, permitiendo también la interacción con el usuario. Como Windows, Linux, Mac OS X,...
- Software: conjunto de los componentes necesarios para hacer posible la realización de una tarea específica. Soporte lógico de la computadora.
- CIF: Código de Identificación Fiscal.

4.5.- Referencias

- IEEE std 830-IEEE Guía para la Especificación de los Requerimientos Software. [7][8]



Aplicación web para gestión integral de talleres de vehículos.

5. Análisis

En la fase de análisis del proyecto se estudian todas las funcionalidades de la aplicación a desarrollar.

El Lenguaje Unificado de Modelado, en adelante UML, por sus siglas en inglés Unified Modeling Language, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema [9].

Es importante resaltar que UML es un "lenguaje de modelado" para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

UML cuenta con varios tipos de diagramas, los cuales visualizan el sistema desde varias perspectivas. Los diagramas más interesantes son los de casos de uso, clases y secuencia, por lo que éstos son los que se desarrollan a continuación.

5.1.- Casos de uso

El diagrama de casos de usos representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como la interacción entre los actores y el sistema. En este caso, se pueden diferenciar tres tipos de actores: el usuario anónimo, que sólo accederá a la parte pública de la aplicación, y el cliente y el administrador del taller, que accederán a sus respectivas partes privadas. Las interacciones entre actores y casos de uso describen el comportamiento del sistema desde el punto de vista de los usuarios.

A continuación se ilustran los casos de uso organizados según la parte de la aplicación que estudiemos, es decir, primero se ilustrará al usuario anónimo en la parte pública de la aplicación y después se ilustrará al actor cliente y al actor administrador en la parte privada de la aplicación Web.



5.1.1.- Actor Anónimo

En la ilustración 5.1. se observar el caso de uso del actor Anónimo. Este tipo de actor sólo podrá acceder a la parte pública de la aplicación, es decir, a la página principal del taller de vehículos.

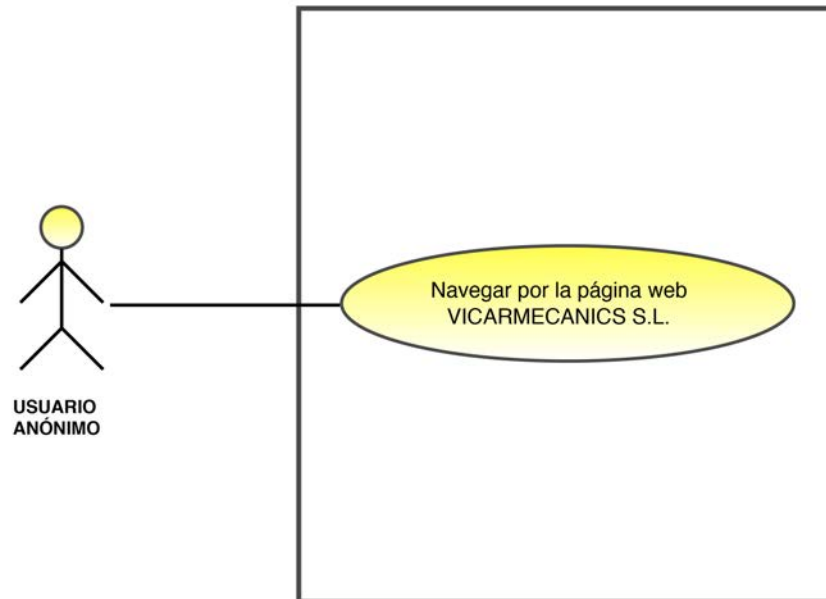


Ilustración 5.1: Diagrama de Uso: Actor anónimo. Navegar por VICARMECANICS S.L.

5.1.2.- Actor Cliente

En la ilustración 5.2. se puede observar el caso de uso asociado al actor Cliente del taller, que podrá listar todos los vehículos que han pasado o están actualmente en el taller de vehículos, así como todas las reparaciones asociadas. Para acceder a la parte privada, el cliente se ha identificado previamente.

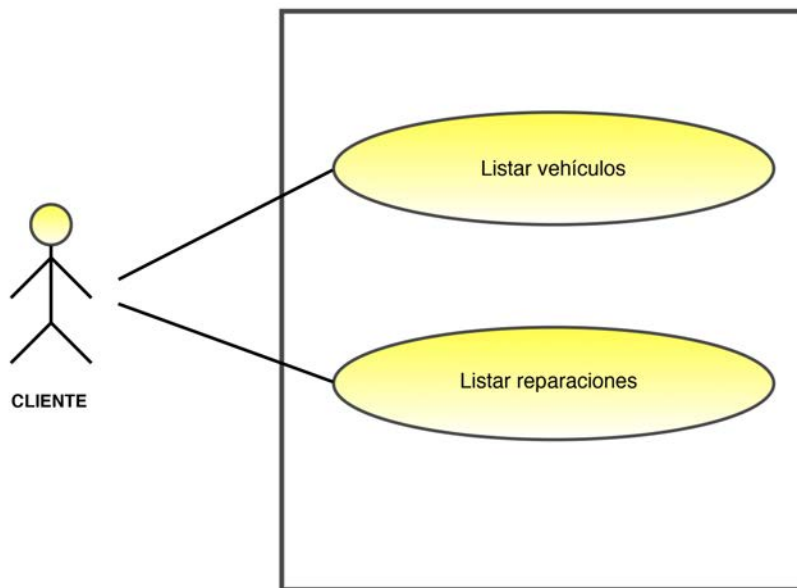


Ilustración 5.2: Diagrama Casos de Uso: Actor cliente. Listar vehículos y listar reparaciones.

5.1.3.- Actor Administrador

En las ilustraciones mostradas a continuación se muestran los casos de uso asociados al actor Administrador del taller.

La ilustración 5.3. corresponde a la gestión de clientes, en la que el administrador podrá listar todos los clientes del taller, crear nuevos clientes, modificar sus datos y también eliminar clientes del taller.

En la ilustración 5.4., correspondiente a la gestión de vehículos, puede observarse cómo el administrador del taller puede listar todos los vehículos que han pasado o están actualmente en el taller, puede crear nuevos vehículos, modificar sus datos o eliminar vehículos.

La ilustración 5.5. corresponde a la gestión de reparaciones, en la que el administrador podrá listar todas las reparaciones llevadas a cabo en el taller, crear nuevas, modificar sus datos y también eliminar reparaciones.

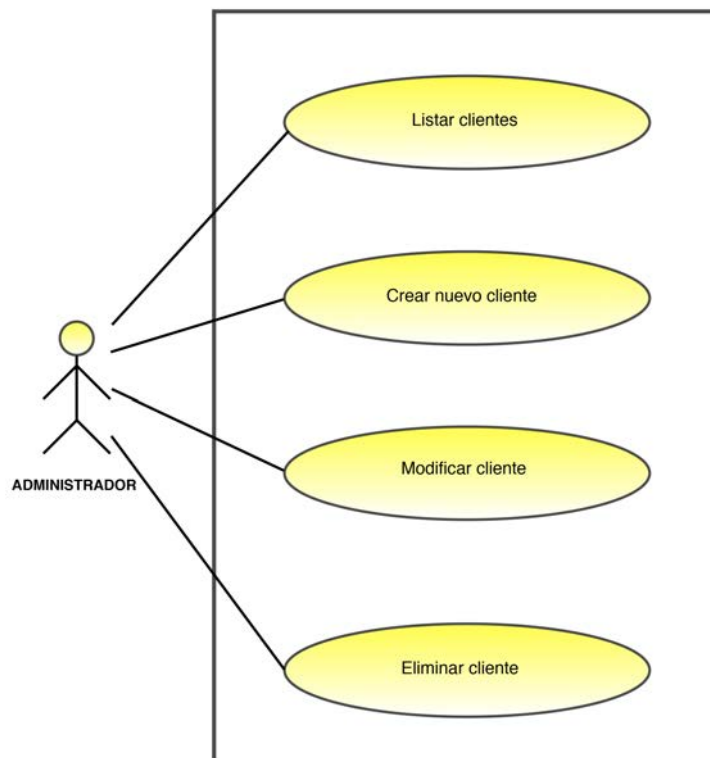


Ilustración 5.3: Diagrama Casos de Uso: Actor administrador. Gestión de clientes.

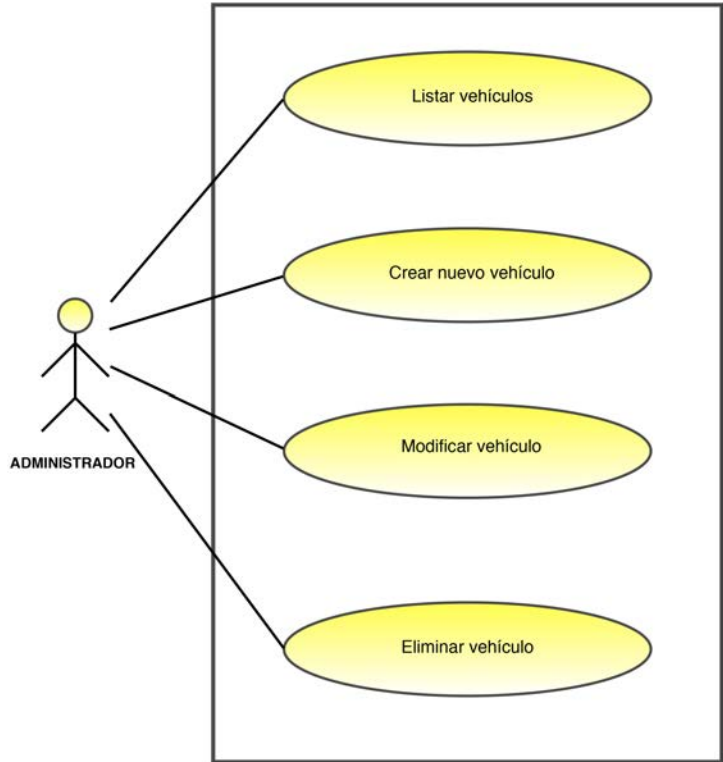


Ilustración 5.4: Diagrama Casos de Uso: Actor administrador. Gestión de vehículos.

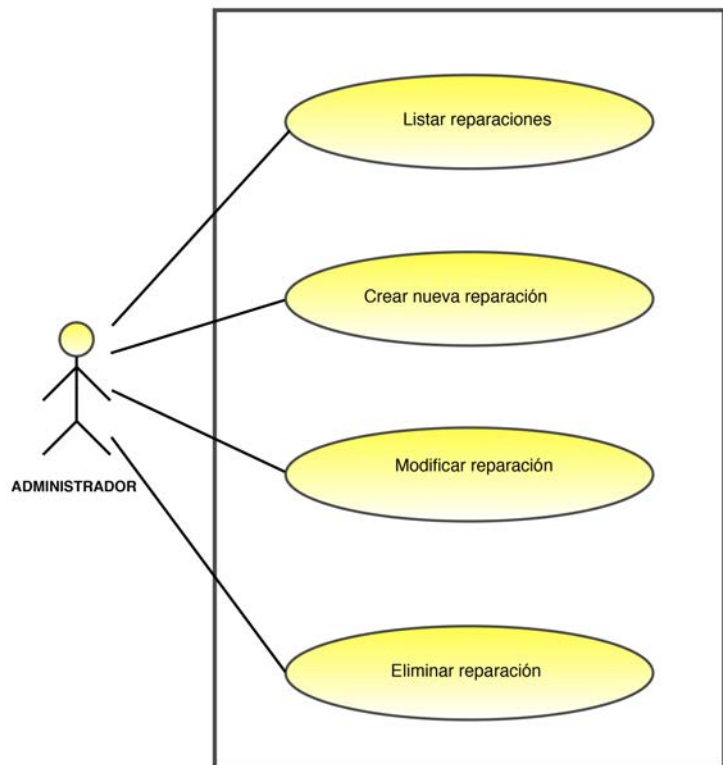


Ilustración 5.5: Diagrama Casos de Uso: Actor administrador. Gestión de reparaciones.

5.2.- Diagrama de clases

El diagrama de clases se utiliza para describir la estructura de un sistema o aplicación. Utilizando la notación UML para diagramas de clases se expone la estructura para la aplicación Web VICARMECANICS S.L., tal y como puede observarse en la ilustración 5.6.

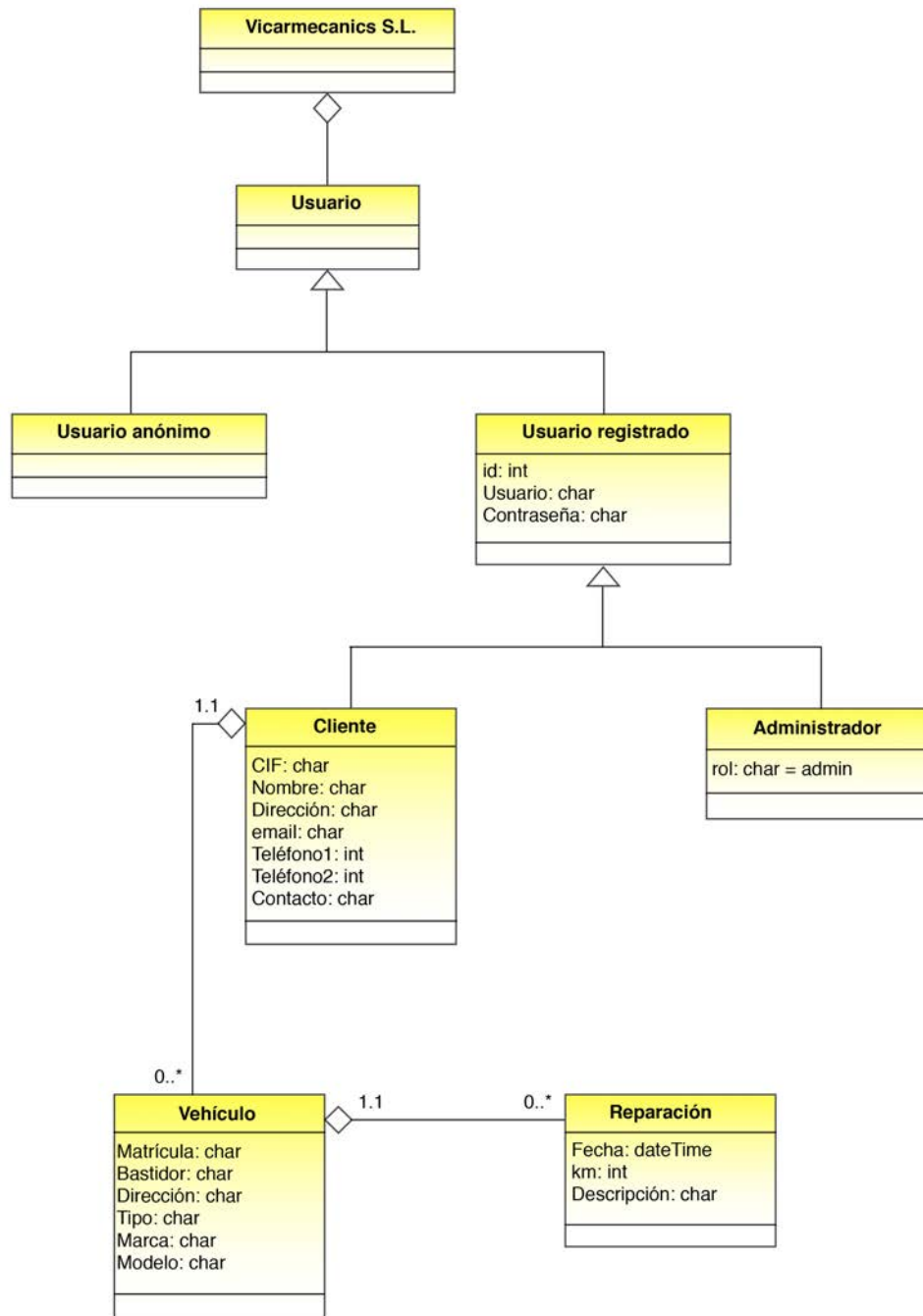


Ilustración 5.6: Diagrama de clases.

Usuario: todo aquel que interactúa con la aplicación, con más o menos privilegios.

Usuario anónimo: aquel que solo puede acceder a la parte pública de la aplicación Web.

Usuario registrado: esta clase representa a los usuarios que pueden acceder a la aplicación tras su correspondiente validación mediante 'usuario' y 'contraseña' y que podrán interactuar con el sistema según los privilegios asociados a su rol. Las subclases Cliente y Administrador derivan de esta clase.

Cliente: esta clase contiene la información de cada cliente que pasa por el taller de vehículos: CIF, dado que los clientes son empresas, nombre, dirección, email, teléfono1, teléfono2 y persona de contacto.

Vehículo: contiene la información de cada vehículo: matrícula, número de bastidor, qué tipo de vehículo es, así como la marca y el modelo. Cada vehículo está asignado a un único cliente.

Reparación: contiene la información de cada reparación: la fecha en la que se inició, los kilómetros a los que se hizo y una descripción de la misma. Cada reparación está asignada a un único vehículo.

5.3.- Diagramas de secuencia

En este tipo de diagrama se muestra la interacción de los objetos que componen el sistema según UML.

5.3.1.- Autenticación del usuario

La ilustración 5.7. muestra el diagrama de secuencias del acceso a la aplicación para usuarios registrados. Después de que el usuario introduzca su usuario y su contraseña, el formulario de validación autentificará y validará los datos. En caso correcto se crea una sesión y en caso incorrecto se produce un error.

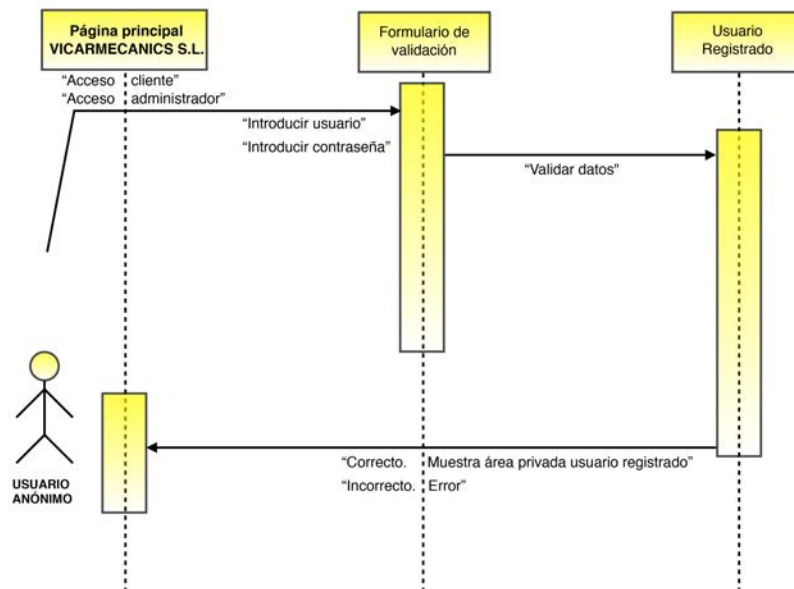


Ilustración 5.7: Diagrama de secuencia: Autenticación usuario.

5.3.2.- Módulo Cliente

En las ilustraciones mostradas a continuación se muestran los diagramas de secuencia correspondientes al módulo del Cliente del taller.

La ilustración 5.8. muestra el diagrama de secuencia de la creación de los listados de vehículos y la ilustración 5.9. muestra el diagrama de secuencia de la creación de reparaciones del cliente, ambas correspondientes a la parte privada del cliente.

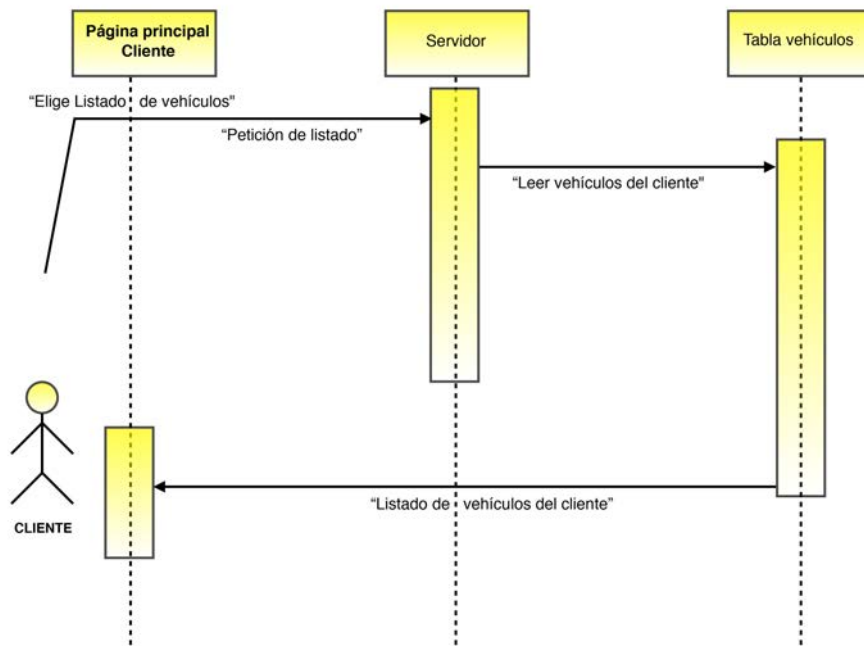


Ilustración 5.8: Diagrama de secuencia: Listado de vehículos del cliente.

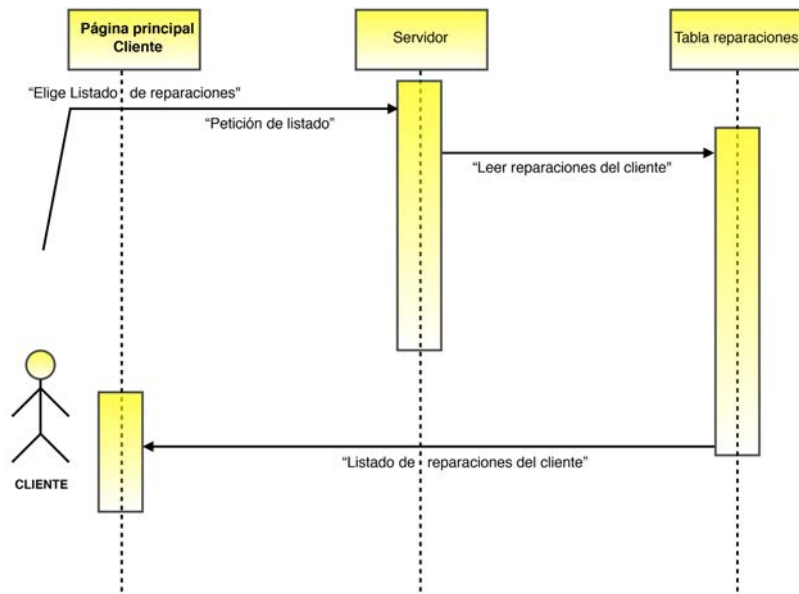


Ilustración 5.9: Diagrama de secuencia: Listado de reparaciones del cliente.

5.3.3.- Módulo Administrador

Las ilustraciones que se muestran a continuación muestran los diagramas de secuencia de la parte privada del Administrador del taller de vehículos.

En la ilustración 5.10. se muestra el diagrama de secuencia correspondiente a la creación de los listados de los clientes del taller. La ilustración 5.11. muestra el diagrama de secuencia de la creación de nuevo cliente, en la ilustración 5.12. se muestra el diagrama correspondiente a la modificación de los datos de un cliente y el diagrama de secuencia de la eliminación de un cliente se muestra en la ilustración 5.13. Todas estas ilustraciones corresponden a la gestión de clientes dentro de la parte privada del Administrador del taller.

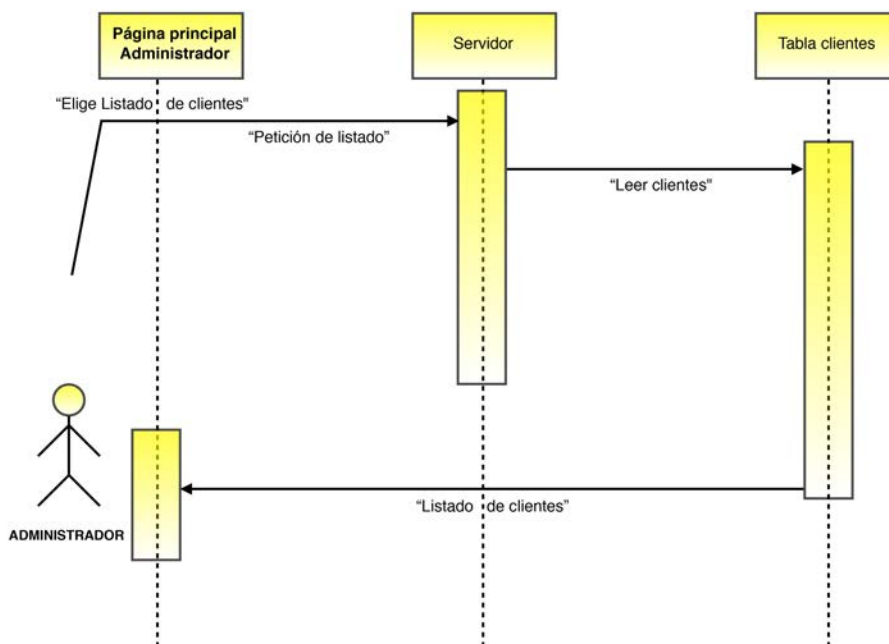


Ilustración 5.10: Diagrama de secuencia: Listado de clientes.

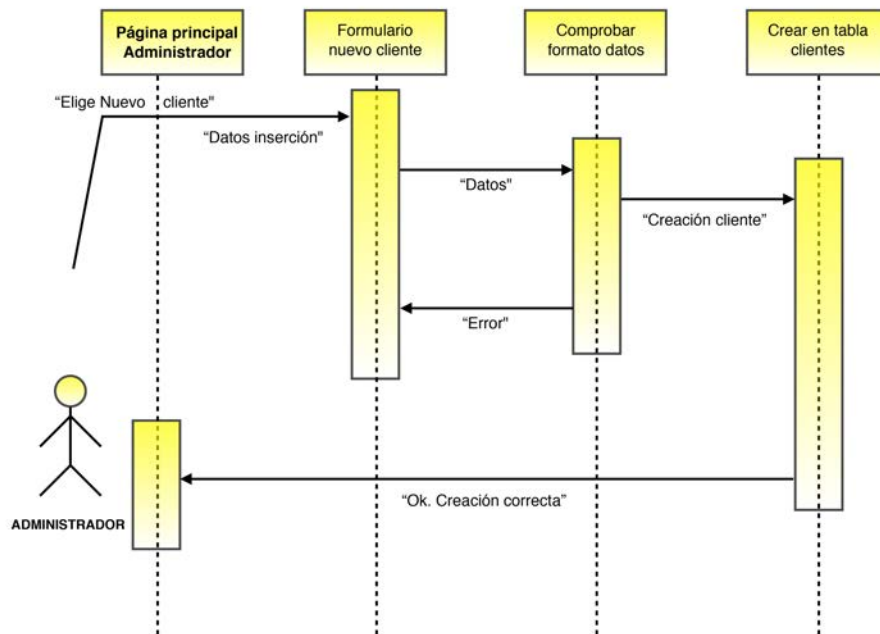


Ilustración 5.11: Diagrama de secuencia: Crear nuevo cliente.

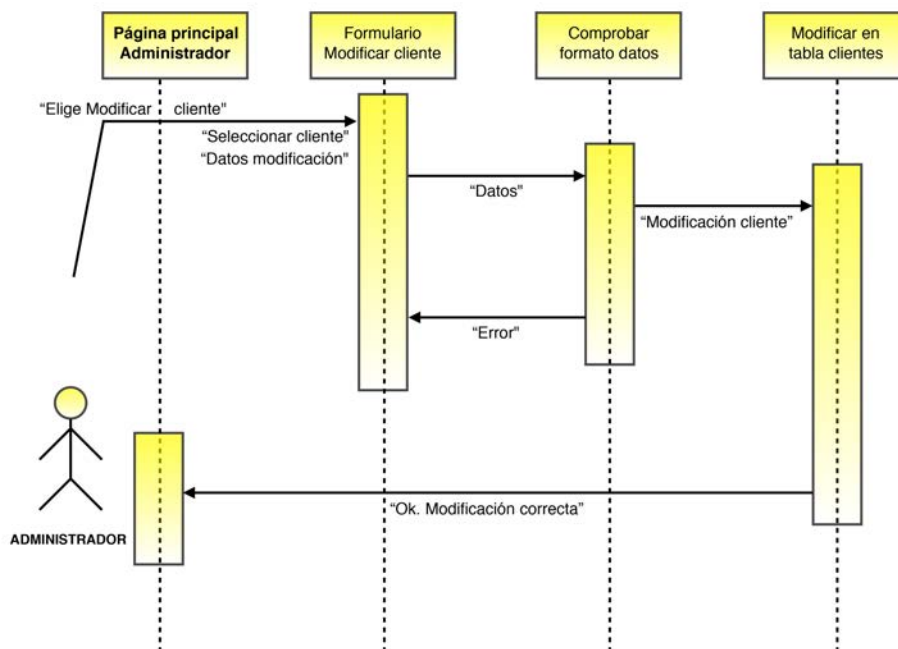


Ilustración 5.12: Diagrama de secuencia: Modificar cliente.

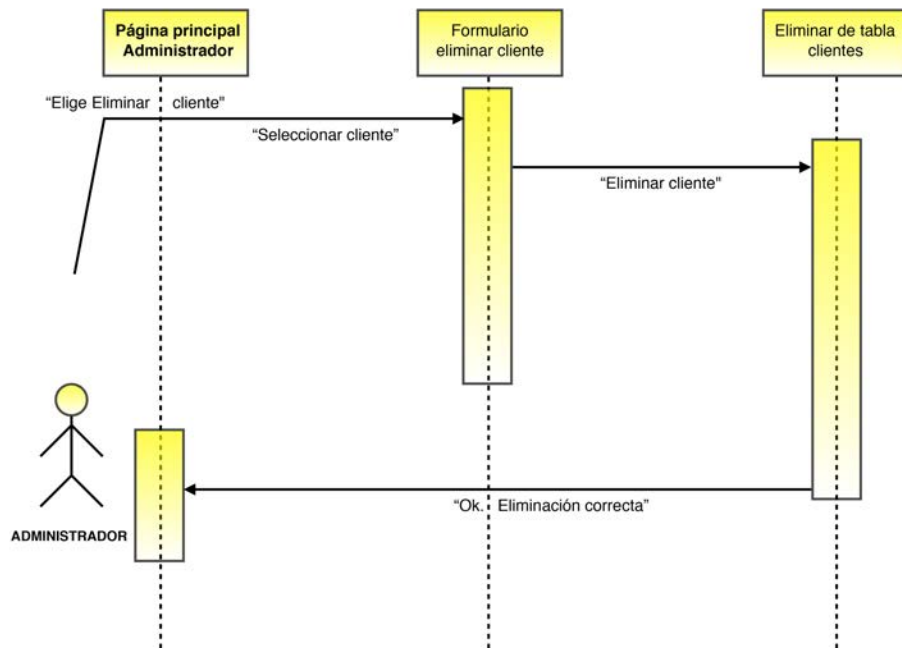


Ilustración 5.13: Diagrama de secuencia: Eliminar cliente.

Las ilustraciones 5.14, 5.15, 5.16 y 5.17 se corresponden con los diagramas de secuencia de la gestión de vehículos. La ilustración 5.14. muestra el diagrama correspondiente a la creación de los listados de los vehículos del taller. La ilustración 5.15. muestra el diagrama de la creación de un nuevo vehículo, en la ilustración 5.16. se puede ver el diagrama correspondiente a la modificación de los datos de un vehículo y el diagrama de secuencia de la eliminación de un vehículo se muestra en la ilustración 5.17.

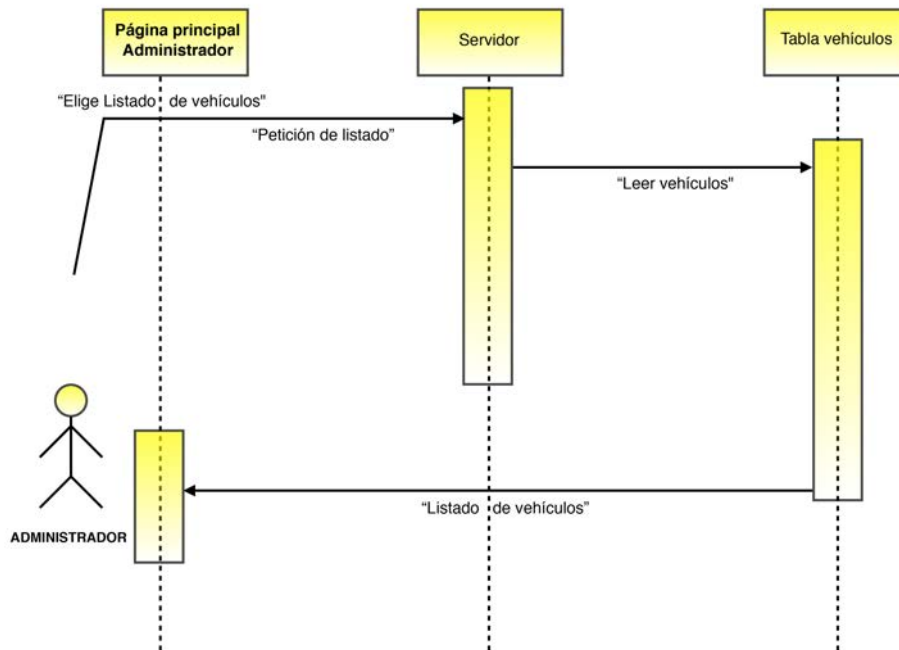


Ilustración 5.14: Diagrama de secuencia: Listado de vehículos.

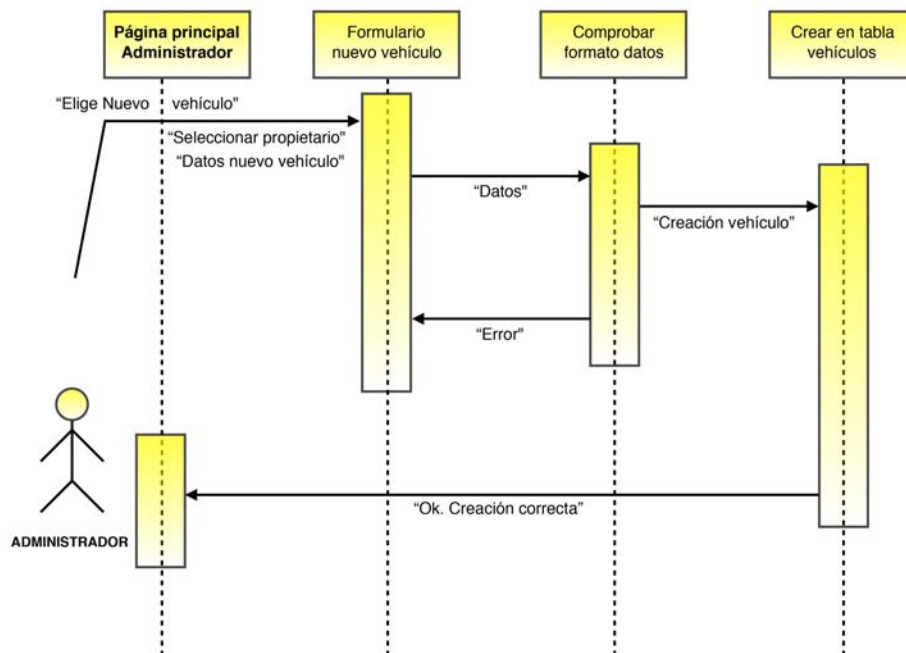


Ilustración 5.15: Diagrama de secuencia: Crear nuevo vehículo.

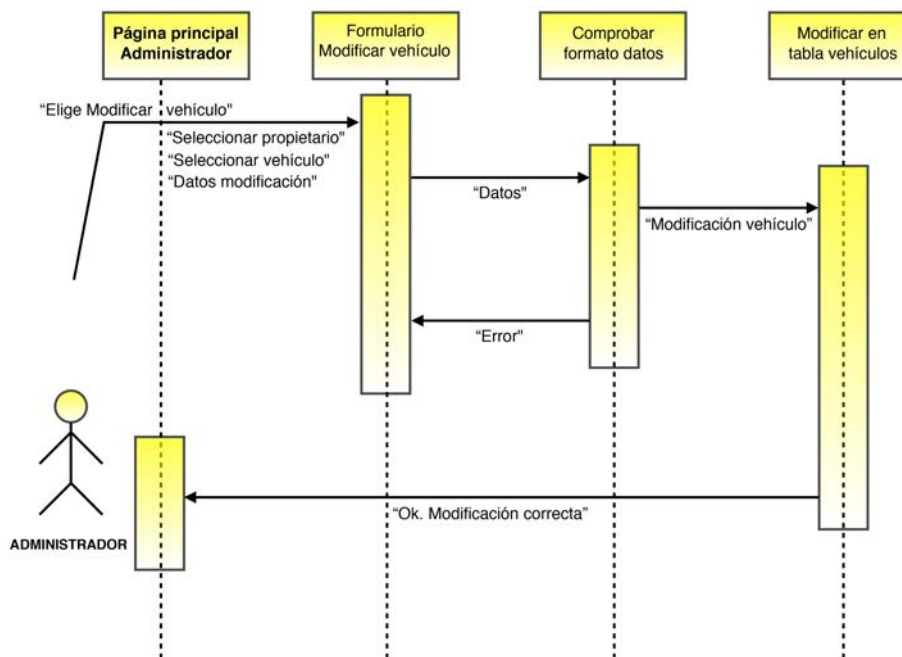


Ilustración 5.16: Diagrama de secuencia: Modificar vehículo.

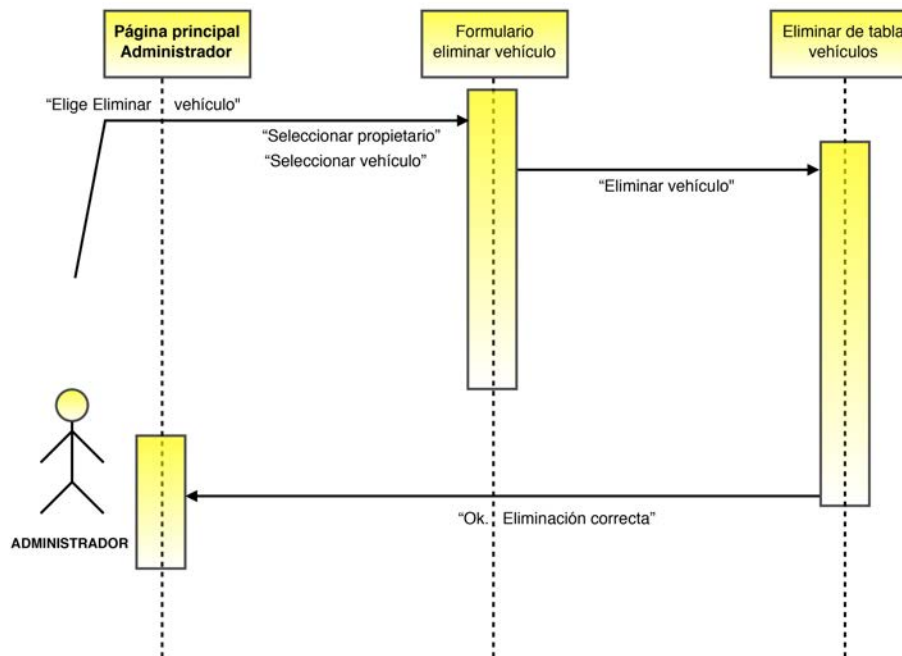


Ilustración 5.17: Diagrama de secuencia: Eliminar vehículo.

A continuación, se muestran las ilustraciones correspondientes con la gestión de reparaciones. El diagrama de secuencia de la creación de los listados de las reparaciones del taller se muestra en la ilustración 5.18. En la ilustración 5.19. puede observarse el diagrama correspondiente a la creación de una nueva reparación y en la ilustración 5.20. se muestra el diagrama de secuencia de la modificación de una reparación. Por último, la ilustración 5.21 muestra el diagrama de secuencia de la eliminación de una reparación.

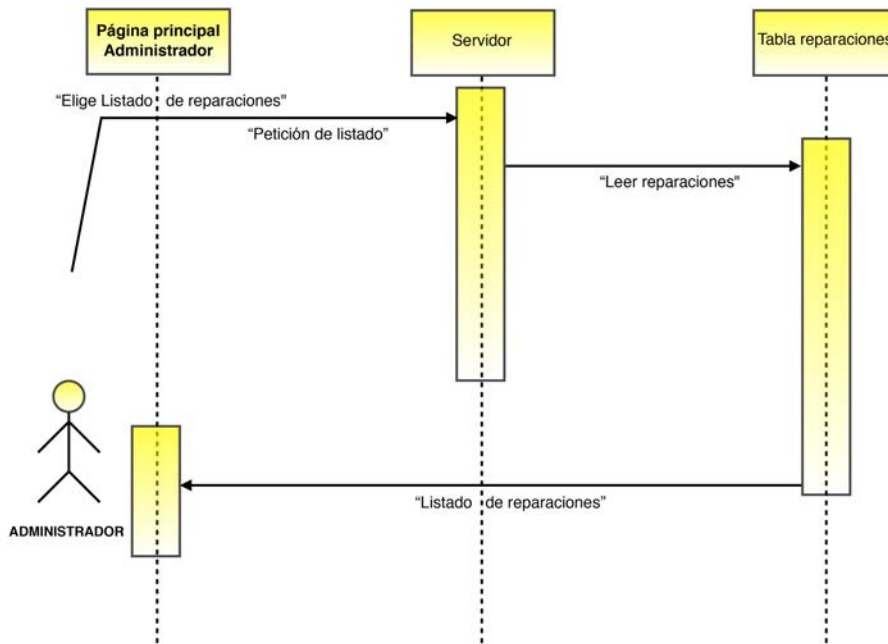


Ilustración 5.18: Diagrama de secuencia: Listado de reparaciones.

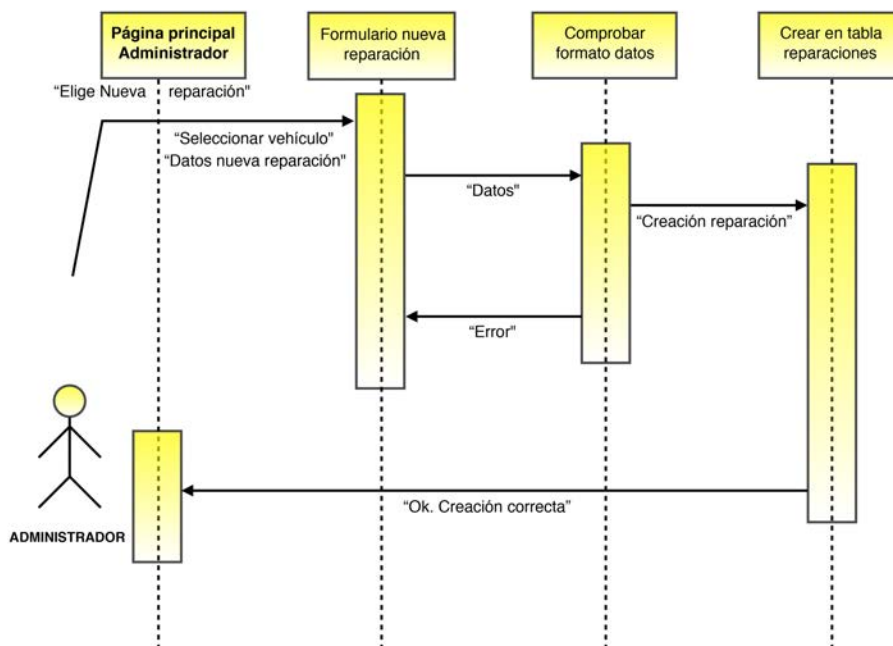


Ilustración 5.19: Diagrama de secuencia: Crear nueva reparación.

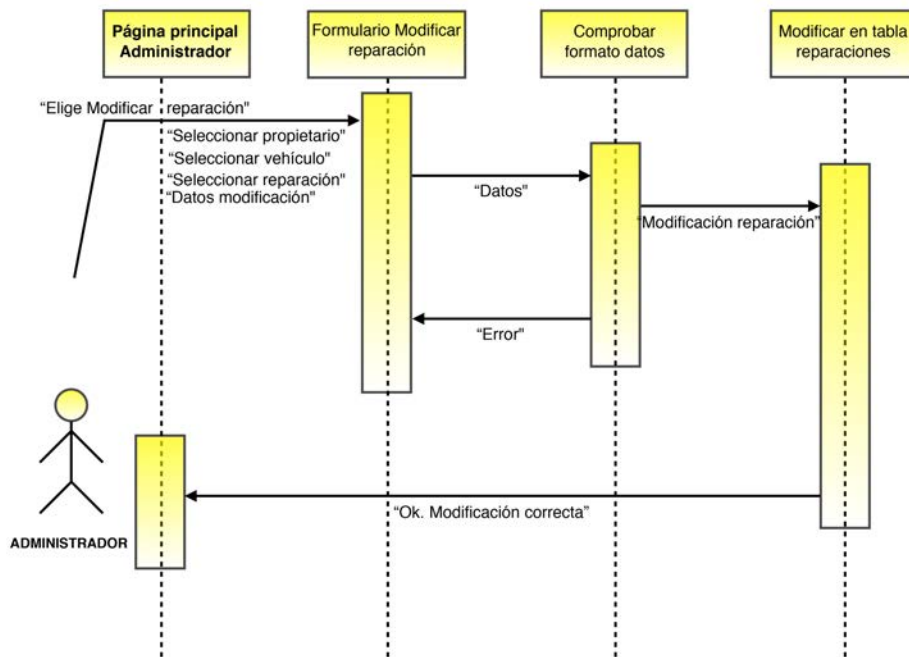


Ilustración 5.20: Diagrama de secuencia: Modificar reparación.

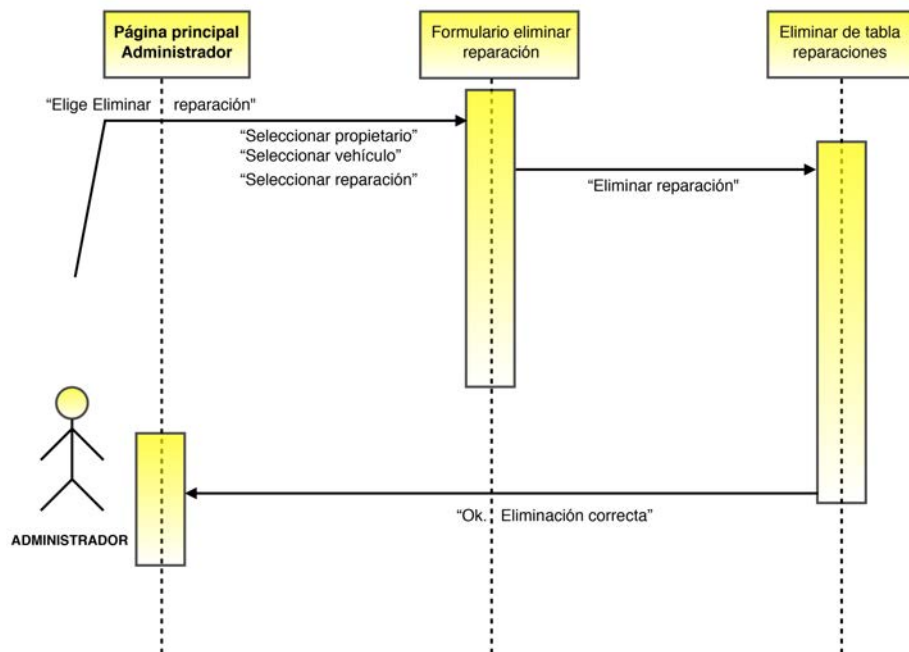


Ilustración 5.21: Diagrama de secuencia: Eliminar reparación.

Aplicación web para gestión integral de talleres de vehículos.

6. Planificación

A continuación, se muestran los resultados de una planificación llevada a cabo después de las reuniones con el cliente y el análisis de los requisitos demandados, ya que para llevar una planificación lo más real posible se necesita una comunicación frecuente entre el cliente y el equipo de desarrollo, de manera que el equipo pueda estimar el esfuerzo, recursos y tiempo requerido para la implementación de la aplicación, y el cliente, por su parte, pueda opinar y decidir sobre el ámbito y tiempo de las entregas.

Además, los posibles imprevistos y cambios que puedan aparecer durante el desarrollo del proyecto se deberán tener en cuenta, ya que la habilidad de responder rápida y eficazmente a estos cambios determina también el éxito o fracaso del mismo. Por tanto, la planificación no deberá ser estricta, sino flexible y abierta.

Los siguientes puntos desglosan todas las etapas o fases que se van a realizar en el desarrollo de la aplicación Web que nos compete. Existen muchos ciclos de vida posibles, pero en este caso, todas las tareas van a seguir un orden secuencial, dado que sólo hay una persona encargada de llevar a cabo el proyecto.

En la ilustración 6.1. encontramos todas las fases que se llevarán a cabo en el desarrollo de la aplicación Web englobadas en cuatro fases principales: especificación de requisitos, modelado conceptual, codificación y pruebas. En la ilustración 6.2. se muestra el diagrama de Gantt de todas estas fases en el tiempo, de manera que se puede observar de qué manera se llevarán a cabo. Como podemos ver en ambas ilustraciones, algunas entregas han cambiado de fecha respecto a la planificación temprana, debido a que la fase de especificación de requisitos nos ha costado un día más y también debido a que la fase de codificación o implementación durará más tiempo a causa de que los requerimientos deseados son mayores y más complejos de lo esperado inicialmente. Sin embargo, se estima poder realizar todas las pruebas en un espacio de tiempo más breve, con lo que la fecha de entrega final del proyecto no cambia y en consecuencia, el presupuesto tampoco, como podemos encontrar en la ilustración 6.3.

Task Name	Duration	Start	Finish
▣ Especificación de requisitos	16 days	Wed 20/01/16	Wed 10/02/16
Estudio previo y entrevistas con el cliente	6 days	Wed 20/01/16	Wed 27/01/16
Identificación de requisitos de usuario	4 days	Thu 28/01/16	Tue 02/02/16
Identificación de requisitos de rendimiento y HW/SW	2 days	Wed 03/02/16	Thu 04/02/16
Identificación de los recursos de la interfaz	2 days	Fri 05/02/16	Mon 08/02/16
Preparar la documentación del sistema	2 days	Tue 09/02/16	Wed 10/02/16
▣ Modelado conceptual	9 days	Thu 11/02/16	Tue 23/02/16
Construcción del modelo	5 days	Thu 11/02/16	Wed 17/02/16
Diseño de la Base de Datos	1 day	Thu 18/02/16	Thu 18/02/16
Creación de un prototipo	1 day	Fri 19/02/16	Fri 19/02/16
Aprobación prototipo por parte de cliente	2 days	Mon 22/02/16	Tue 23/02/16
▣ Codificación	16 days	Wed 24/02/16	Wed 16/03/16
Implementación de la aplicación	16 days	Wed 24/02/16	Wed 16/03/16
▣ Pruebas	5 days	Mon 21/03/16	Tue 29/03/16
Pruebas unitarias de los módulos	3 days	Mon 21/03/16	Wed 23/03/16
Pruebas de integración de los componentes	1 day	Thu 24/03/16	Thu 24/03/16
Pruebas de navegadores	1 day	Tue 29/03/16	Tue 29/03/16
Pruebas de seguridad de acceso	1 day	Tue 29/03/16	Tue 29/03/16
Pruebas validación de enlaces	1 day	Tue 29/03/16	Tue 29/03/16

Ilustración 6.1: Planificación después del análisis. Fases y tareas.

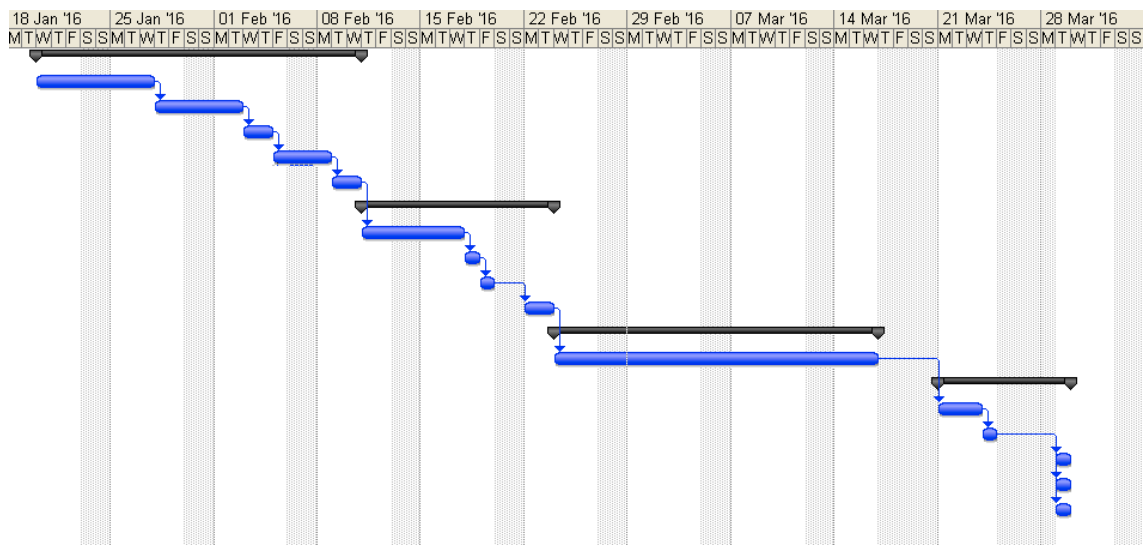


Ilustración 6.2: Planificación después del análisis. Diagrama de Gantt.

Project Statistics for 'planificación'			
	Start		Finish
Current	Wed 20/01/16		Tue 29/03/16
Baseline	NA		NA
Actual	NA		NA
Variance	0d		0d
	Duration	Work	Cost
Current	46d	368h	3.680,00 €
Baseline	0d?	0h	0,00 €
Actual	0d	0h	0,00 €
Remaining	46d	368h	3.680,00 €

Percent complete:

Duration: 0% Work: 0%

Ilustración 6.3: Planificación después del análisis. Costes.

7. Diseño

7.1.- Arquitectura de tres capas

El diseño sigue una arquitectura de tres capas con un reparto claro de funciones: todo sistema que gestiona datos tendrá una base de datos para guardar esos datos y una interfaz de usuario con la que los usuarios interactúan. Además, es necesario que una parte del sistema se encargue de procesar los datos y gestionar lo que se hace con ellos. La arquitectura en tres capas lo que hace es dividir el sistema en tres partes bien diferenciadas, de forma que cada capa solamente se comunique con la capa inmediatamente inferior. Esas tres capas se denominan como capa de presentación, capa de lógica de negocio y una última de persistencia de datos. Con esta arquitectura se pretende conseguir una alta cohesión y un bajo acoplamiento para facilitar el desarrollo y mantenimiento de la aplicación.

En la ilustración 7.1. podemos ver una representación gráfica de la arquitectura de tres capas, presentación, lógica de negocio y persistencia, así como la relación entre las mismas.

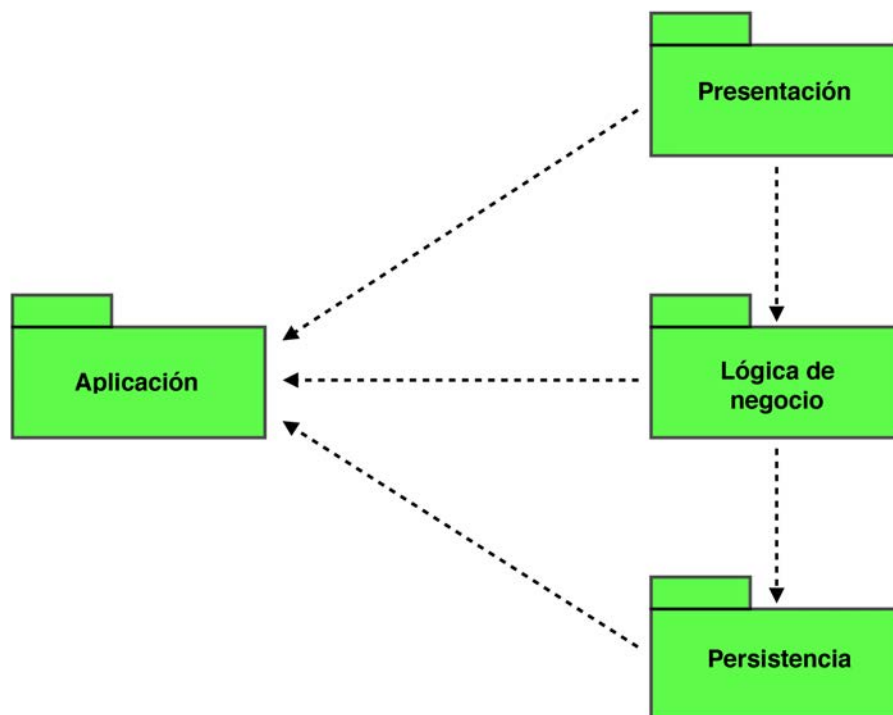


Ilustración 7.1: Arquitectura de tres capas.

7.1.1.- Capa de presentación

En esta capa se crea la interfaz del usuario. Su única función es pasarle las acciones que realice el usuario a la capa de lógica de negocio. La interfaz de usuario será amigable, es decir entendible y con un diseño sencillo para facilitar al usuario la usabilidad de la aplicación.

Las tres ilustraciones siguientes muestran la interfaz gráfica de la parte pública de la aplicación Web. La ilustración 7.2. muestra los apartados de ‘Talleres Vicarmecanics S.L.’, ‘Nuestra empresa’ y también ‘Servicios’. En la ilustración 7.3. puede observarse el apartado de ‘Galería’ y por último, en la ilustración 7.4. se muestran los apartados de ‘Acceso clientes’ y ‘Contacto’. Como puede observarse el menú superior siempre está visible, donde a parte de encontrar los apartados citados anteriormente, se encuentran también los enlaces de ‘Acceso clientes’ y ‘Acceso administrador’.

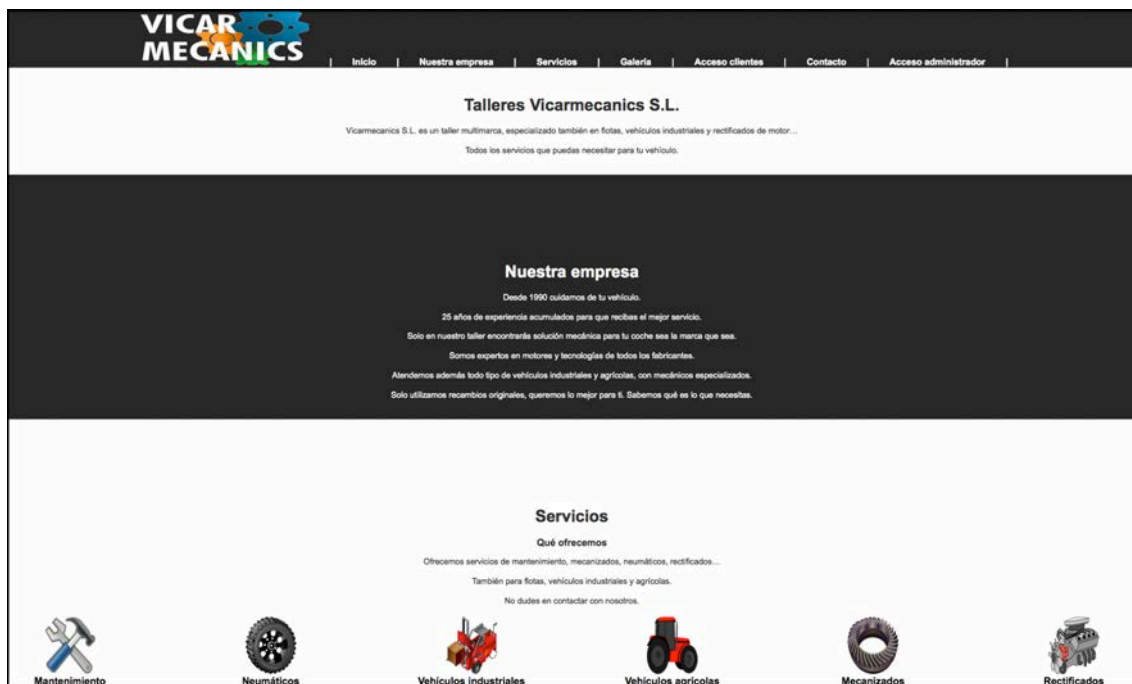


Ilustración 7.2: Interfaz parte pública I.

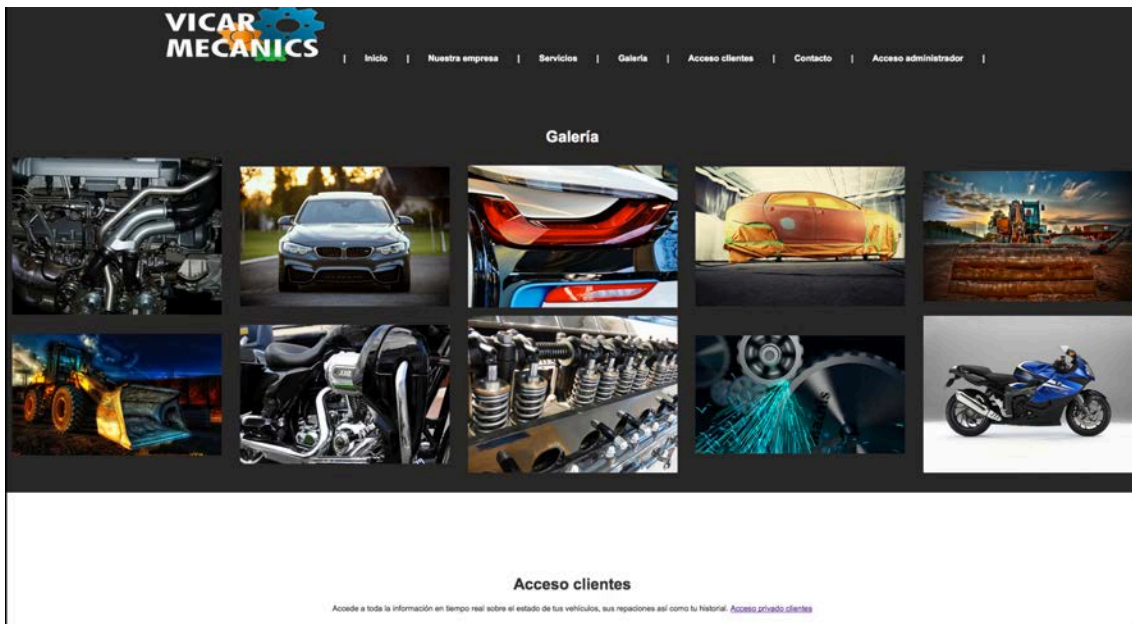


Ilustración 7.3: Interfaz parte pública II.

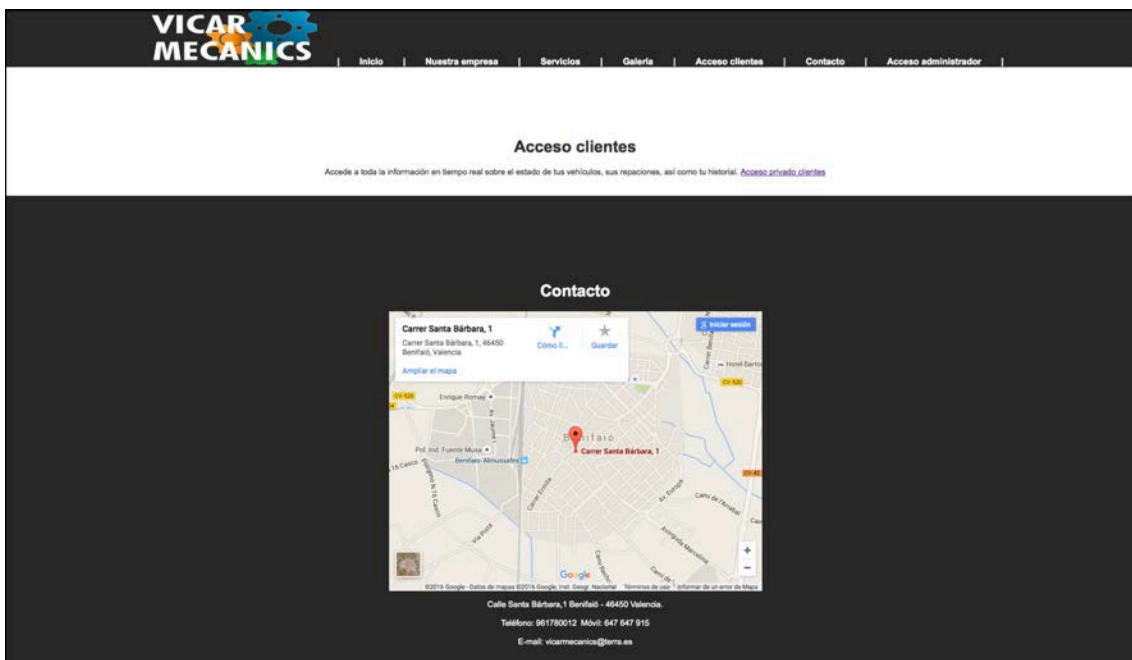


Ilustración 7.4: Interfaz parte pública III.

Aplicación web para gestión integral de talleres de vehículos.

A continuación, se incluyen las ilustraciones que muestran las interfaces de la parte privada del cliente y del administrador del taller. En la ilustración 7.5. se muestra la parte privada del cliente y en la ilustración 7.6. puede observarse la parte privada del administrador del taller.



Ilustración 7.5: Interfaz parte privada cliente.



Ilustración 7.6: Interfaz parte privada administrador.

Como se puede comprobar en las ilustraciones se guarda la homogeneidad en todas las páginas de la aplicación, tanto públicas como privadas.

7.1.2.- Capa de lógica de negocio

En esta capa se gestiona la lógica de la aplicación y es la que contiene los componentes software que implementan el comportamiento especificado en la fase anterior. Esta capa debe comunicarse con otras capas, con la de presentación para dar respuesta a las solicitudes de información y con la de persistencia para hacer consultas de información.

Toma información de la base de datos según las necesidades del usuario y del sistema. También recogerá información suministrada por el usuario mediante la capa de presentación y la almacenará en la base de datos.

Los componentes software necesarios para implementar el comportamiento de la solución vienen definidos por el conjunto de archivos que forman la aplicación y que pueden observarse en la ilustración 7.7.

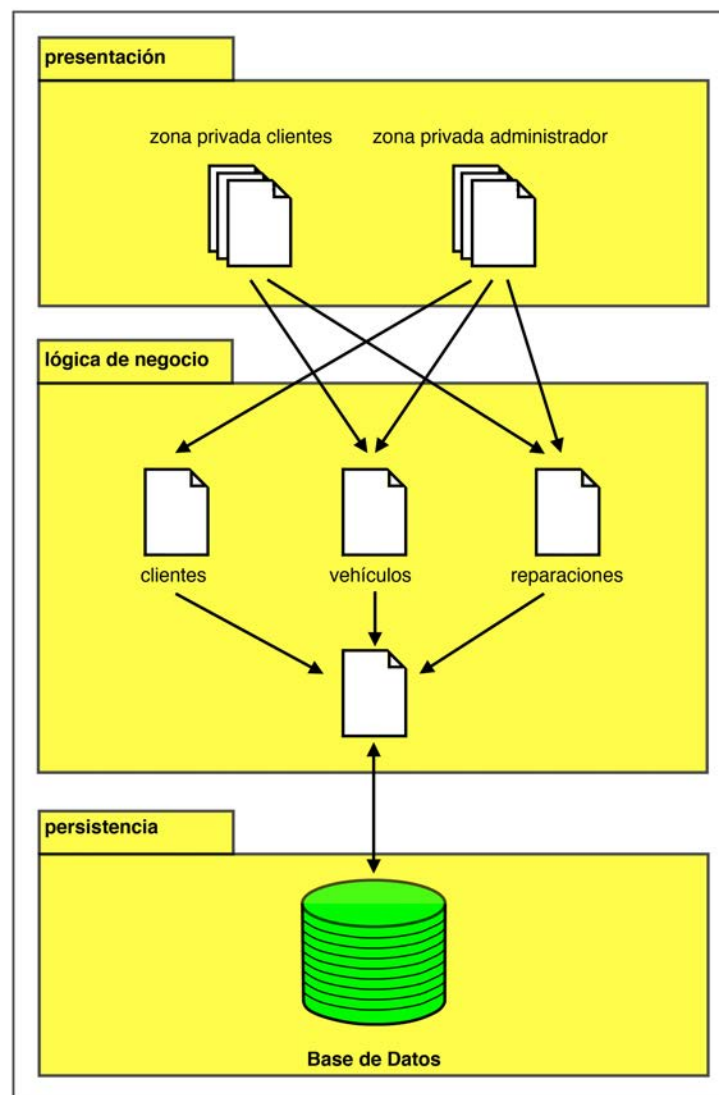


Ilustración 7.7: Módulos de la aplicación.

7.1.3.- Capa de persistencia

Esta capa se encarga de guardar los datos. Será donde se gestione todo lo relativo a la base de datos y a la creación, edición y borrado de datos. La base de datos está compuesta por las tablas del diagrama Entidad-Relación que pueden observarse en la ilustración 7.8. mostrada a continuación.



Ilustración 7.8: Capa de persistencia.

8. Prototipo

Un prototipo tiene dos usos principales: por un lado, es un método eficaz para visualizar y definir de manera clara los requerimientos de los usuarios y de una aplicación en general y, por otro lado, es el mejor medio para verificar el diseño del sistema.

Un prototipo es, una aplicación que trabaja, creada de forma rápida y económica pero que muestra distintos aspectos de la aplicación de una manera muy general.

Como se puede observar en la planificación, dada la sencillez de nuestra aplicación Web, en nuestro proyecto se realizará un solo prototipo.

El prototipo desarrollado se caracteriza por ofrecer a nuestro cliente una visión general de lo que será la aplicación Web, con una base de datos sencilla que permite un seguimiento muy básico de la gestión del taller. De esta manera, el cliente podrá visualizar la estructura de la interfaz principal y la organización de los distintos módulos, así como distribuir la información como desee o añadir o modificar información que considere relevante para su negocio.

De esta manera, este prototipo nos permite realizar un feedback con el cliente, ayudándonos a realizar la aplicación Web tal y como el cliente desea.

Aplicación web para gestión integral de talleres de vehículos.

9. Implementación

9.1.- Tecnologías

Para la implementación de la aplicación Web que nos atañe se crearán contenidos dinámicos para la total configuración, de manera que todo el acceso a la base de datos es transparente tanto al cliente como al administrador del taller de vehículos.

Para cumplir con lo especificado, la aplicación constará de una base de datos y una página web dinámica en HTML que realiza consultas SQL a una base de datos por medio de código PHP. El código PHP estará incrustado dentro del HTML interactuando con el mismo, lo que permite diseñar la página web con un editor común de HTML y añadir el código dinámico PHP usando las etiquetas indicadas para ello [10].

El lenguaje PHP está especialmente indicado para el desarrollo de aplicaciones web dinámicas con acceso a información que está almacenada en una base de datos, de esta manera, será la propia aplicación web la encargada de comunicarse con la base de datos, en la que se consultará, modificará o eliminará la información relacionada con la transacción realizada, tanto de la gestión de clientes como de la gestión de vehículos y reparaciones. Cabe resaltar que el propio código en PHP es invisible al usuario, dando lugar a una programación segura [11].

9.2.- Funcionalidades y descripción de la implementación

La aplicación recibe el nombre de Vicarmecanics S.L.

Las funcionalidades implementadas son las siguientes:

- Parte pública: “inicio.html”
 - Página Web principal del taller de vehículos con los apartados de ‘Nuestra empresa’, ‘Servicios’, ‘Galería’, ‘Acceso clientes’, ‘Contacto’ y ‘Acceso administrador’.

- Parte privada:
 - Módulo de cliente del taller, “index_client.html” con acceso mediante usuario y contraseña, formado por “Titol_cli.html”, “Benvinguda_cli.html” y “Index_cli.html” con las siguientes funcionalidades:
 - Gestión de vehículos:
Listado de vehículos del cliente -> “LlistaVehicles_cli.php”
 - Gestión de reparaciones:
Listado de reparaciones del cliente -> “LlistaReparacions_cli.php”

La parte privada del cliente queda estructurada como se puede ver en la ilustración 9.1, en la que podemos observar cómo se organizan y visualizan los archivos .html y .php anteriormente comentados.

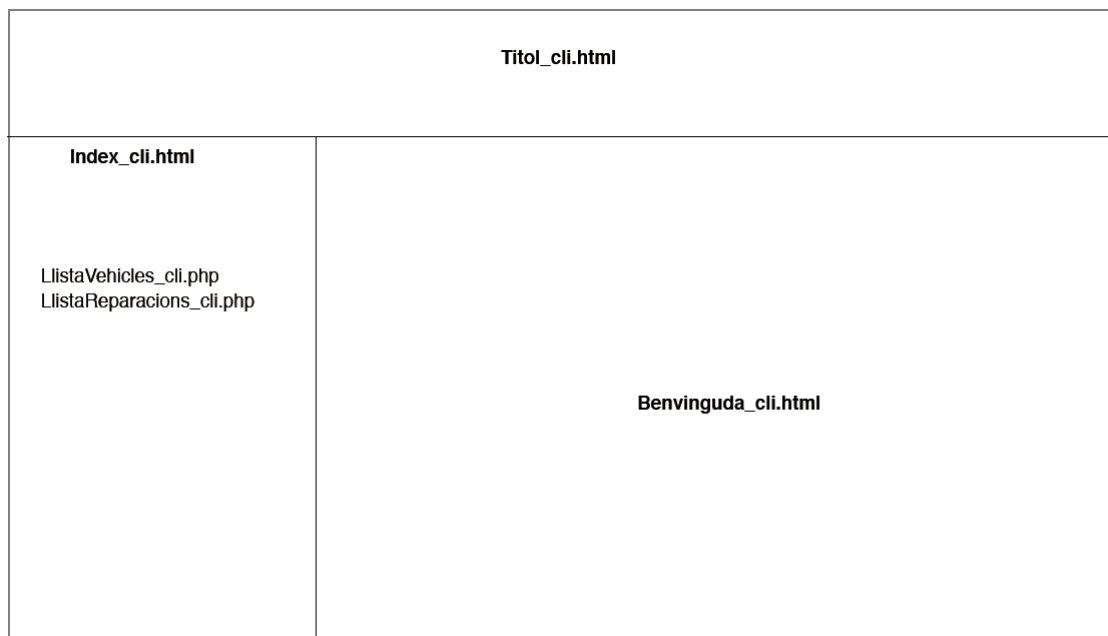


Ilustración 9.1: Implementación parte privada del cliente.

- Módulo de administrador del taller, “vicarmecanics2.html” con acceso mediante usuario y contraseña, formado por “Titol.html”, “Benvinguda.html” y “index_admin.html” con las siguientes funcionalidades:

- Gestión de clientes:

Listado de clientes ->LlistaClients.php

Nuevo cliente ->NouClient.php

Modificar cliente ->ModificaClient.php

Eliminar cliente ->BorraClient.php

- Gestión de vehículos:

Listado de vehículos ->LlistaVehicles.php

Nuevo vehículo ->NouVehicle.php

Modificar vehículo ->ModificaVehicle.php

Eliminar vehículo ->BorraVehicle.php

- Gestión de reparaciones:

Listado de reparaciones ->LlistaReparacions.php

Nueva reparación ->NouReparacio.php

Modificar reparación ->ModificaReparacio.php

Eliminar reparación ->BorraReparacio.php

La parte privada del administrador se estructura como puede observarse en la ilustración 9.2. En esta ilustración se muestra la organización de todos los archivos .html y .php que conforman esta parte privada del administrador.

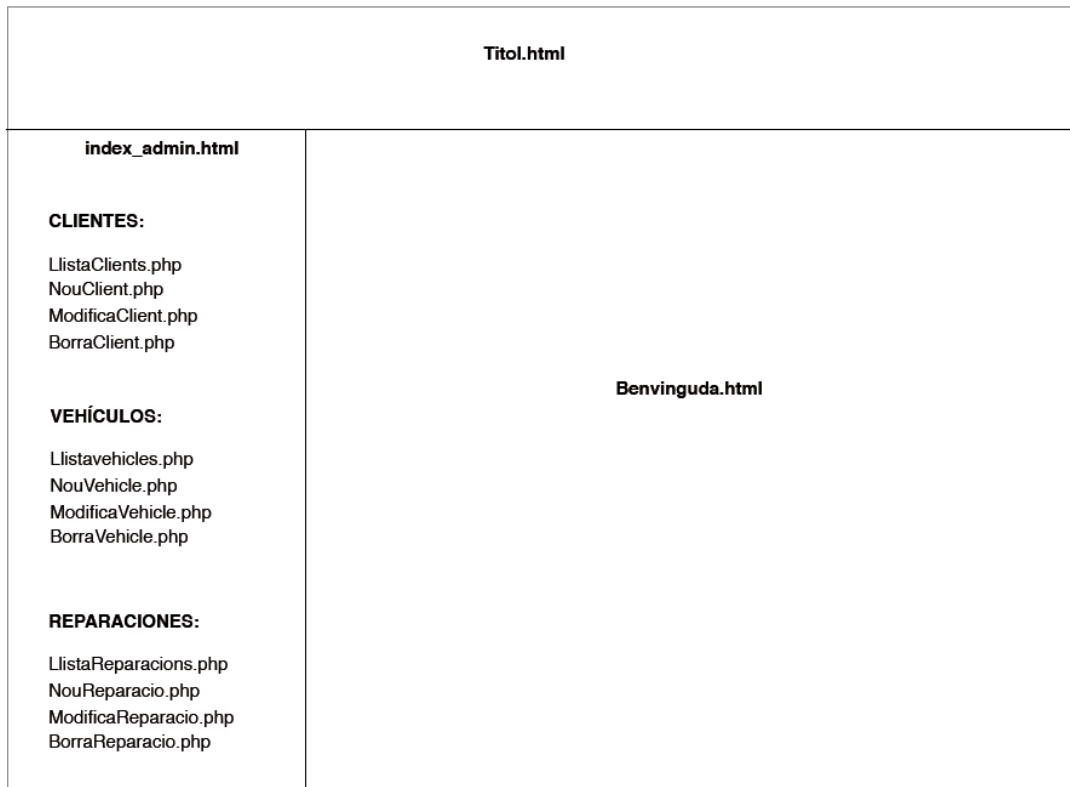


Ilustración 9.2: Implementación parte privada del administrador.

10. Pruebas

Esta etapa consiste en comprobar que la aplicación Web desarrollada lleve a cabo eficientemente las tareas definidas en la especificación del proyecto.

10.1.- Prueba de los módulos

La primera técnica de prueba elegida para llevar a cabo esta comprobación, será la de probar cada uno de los módulos software por separado, para una vez comprobados, probar toda la aplicación de manera integral.

Aunque se considera como buena práctica que las distintas pruebas sean llevadas a cabo por alguien distinto al desarrollador, por la propia naturaleza del proyecto en este caso será la misma persona quien pruebe la aplicación Web.

A continuación, se enumeran las distintas pruebas realizadas a la aplicación Web desarrollada.

- **Módulo de cliente del taller:**

- **Usuario / Contraseña.**

- Longitud del campo de Usuario.

- Longitud del campo de Contraseña.

- Mensajes de error.

- Validación de nombre de Usuario y Contraseña correctos.

- **Listado de vehículos.**

- Visualización del listado.

- **Listado de reparaciones.**

- Visualización del listado.

- **Módulo de Administrador del taller:**

- **Usuario / Contraseña.**

- Longitud del campo de Usuario.

- Longitud del campo de Contraseña.

- Mensajes de error.

- Validación de nombre de Usuario y Contraseña correctos.

- **Gestión de clientes:**

- **Listado de clientes.**

- Visualización del listado.

- **Nuevo cliente.**

- Longitud de los campos.

- Campos requeridos.

- El CIF empieza por una letra y tiene 8 dígitos.

- El teléfono tiene 9 cifras numéricas y debe empezar por 9 ó 6.

- Mensajes de error.

- Sin los campos obligatorios correctos el botón de 'Crear' no se activa.

- Creación del nuevo cliente en la Base de Datos.

- Limpiar los datos del formulario.

- **Modificar cliente.**

- Selección del cliente a modificar.

- Los campos se rellenan automáticamente y se pueden modificar.

- Longitud de los campos.

- Campos requeridos.

- El CIF empieza por una letra y tiene 8 dígitos.

El teléfono tiene 9 cifras numéricas y debe empezar por 9 ó 6.

Mensajes de error.

Sin la selección de un cliente el botón de 'Modificar Cliente' no se activa.

Modificación de los datos del cliente en la Base de Datos.

Limpiar los datos del formulario.

➤ **Eliminar cliente.**

Selección del cliente a eliminar.

Sin la selección del cliente el botón de Eliminar Cliente no se activa.

Eliminación del cliente en la Base de Datos.

- **Gestión de vehículos:**

➤ **Listado de vehículos.**

Visualización del listado.

➤ **Nuevo vehículo.**

Selección del propietario del vehículo.

Longitud de los campos.

Campos requeridos.

La matrícula acepta el formato de las matrículas antiguas y actuales.

El bastidor está formado por 17 caracteres, tanto texto como números.

Mensajes de error.

Sin la selección de un propietario el botón de 'Crear' no se activa.

Creación del nuevo vehículo en la Base de Datos.

Limpiar los datos del formulario.

➤ **Modificar vehículo.**

Selección del propietario del vehículo.

Selección del vehículo a modificar.

Los campos se rellenan automáticamente y se pueden modificar.

Longitud de los campos.

Campos requeridos.

La matrícula acepta el formato de las matrículas antiguas y actuales.

El bastidor está formado por 17 caracteres, tanto texto como números.

Mensajes de error.

Sin la selección de un propietario y de un vehículo el botón de ‘Modificar Vehículo’ no se activa.

Modificación de los datos del vehículo en la Base de Datos.

Limpiar los datos del formulario.

➤ **Eliminar vehículo.**

Selección del propietario del vehículo.

Selección del vehículo.

Sin la selección de un propietario y del vehículo el botón de ‘Eliminar Vehículo’ no se activa.

Eliminación del vehículo de la Base de Datos.

- **Gestión de reparaciones:**

➤ **Listado de reparaciones.**

Visualización del listado.

➤ **Nueva reparación.**

Selección del vehículo asociado a la reparación.

Longitud de los campos.

Campos requeridos.

Fecha actual automática.

El campo km es numérico y tiene una longitud máxima de 6 números.

Mensajes de error.

Sin la selección de un vehículo el botón de 'Crear' no se activa.

Creación de la nueva reparación en la Base de Datos.

Limpiar los datos del formulario.

➤ **Modificar reparación.**

Selección del propietario del vehículo.

Selección del vehículo asociado a la reparación.

Selección de la reparación a modificar.

Los campos se rellenan automáticamente y se pueden modificar.

Longitud de los campos.

Campos requeridos.

El campo km es numérico y tiene una longitud máxima de 6 números.

Mensajes de error.

Sin la selección de una reparación el botón de 'Modificar Reparación' no se activa.

Modificación de los datos de la reparación en la Base de Datos.

➤ **Eliminar reparación.**

Selección del propietario.

Selección del vehículo.

Selección de la reparación.

Sin la selección de una reparación el botón de 'Eliminar Reparación' no se activa.

Eliminación de la reparación de la Base de Datos.

10.2.- Prueba de validación de enlaces

Esta prueba consiste en comprobar que todos los enlaces de la aplicación Web están bien formados y que todos enlazan correctamente.

Para ello se ha usado la siguiente herramienta disponible en la página web <http://validator.w3.org/checklink> [12]. En la ilustración 10.1 se muestra una captura de pantalla en la que se ve que el resultado de esta prueba ha sido válido.



Ilustración 10.1: Prueba de validación de enlaces.

10.3.- Prueba de navegadores

Se ha comprobado el correcto funcionamiento de la aplicación Web en distintos navegadores. En la ilustración 10.2. se muestra la aplicación en el navegador Mozilla Firefox. En la ilustración 10.3. se muestra la aplicación en el navegador Google Chrome, y, por último, se muestra también la aplicación en navegador Safari , como puede verse en la ilustración 10.4. Como se observa en las tres ilustraciones los tres navegadores soportan correctamente la aplicación Web desarrollada.

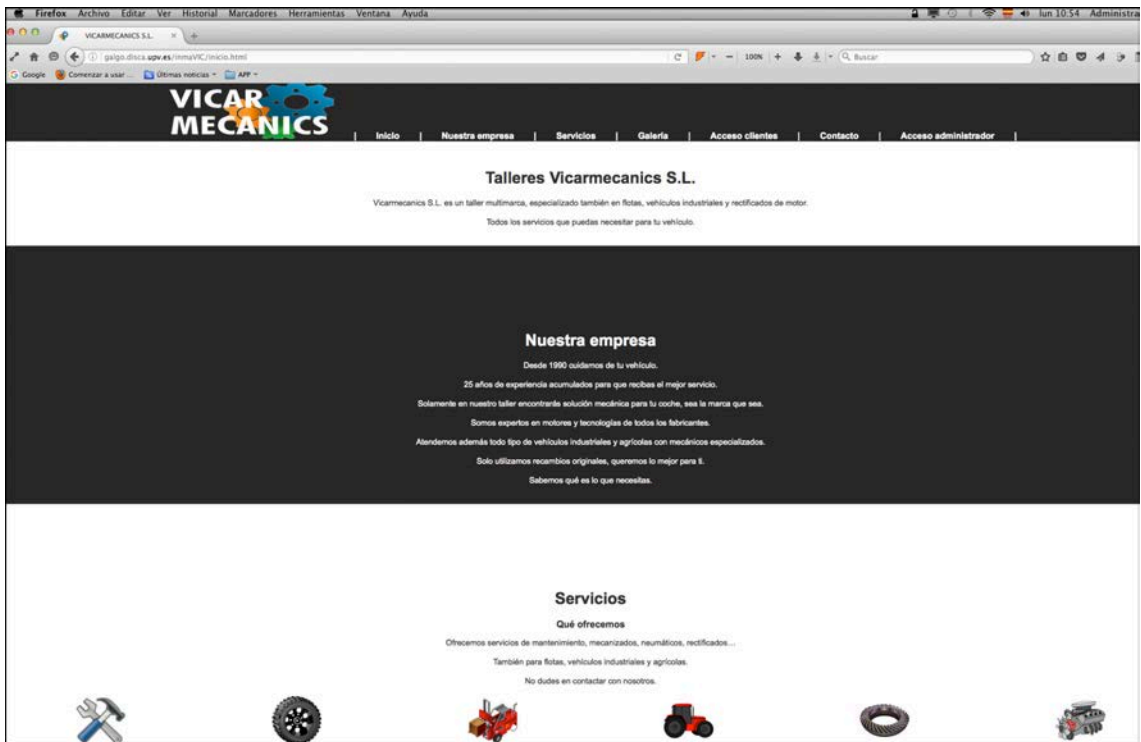


Ilustración 10.2: Navegador Mozilla Firefox.

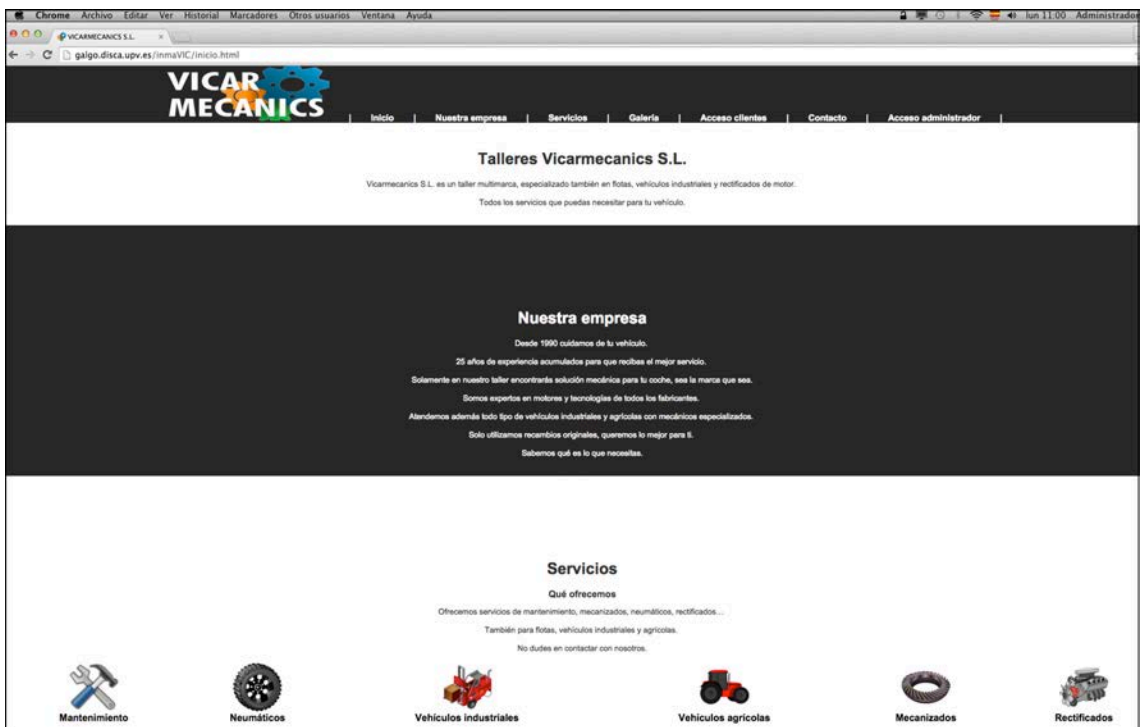


Ilustración 10.3: Navegador Google Chrome.

Aplicación web para gestión integral de talleres de vehículos.

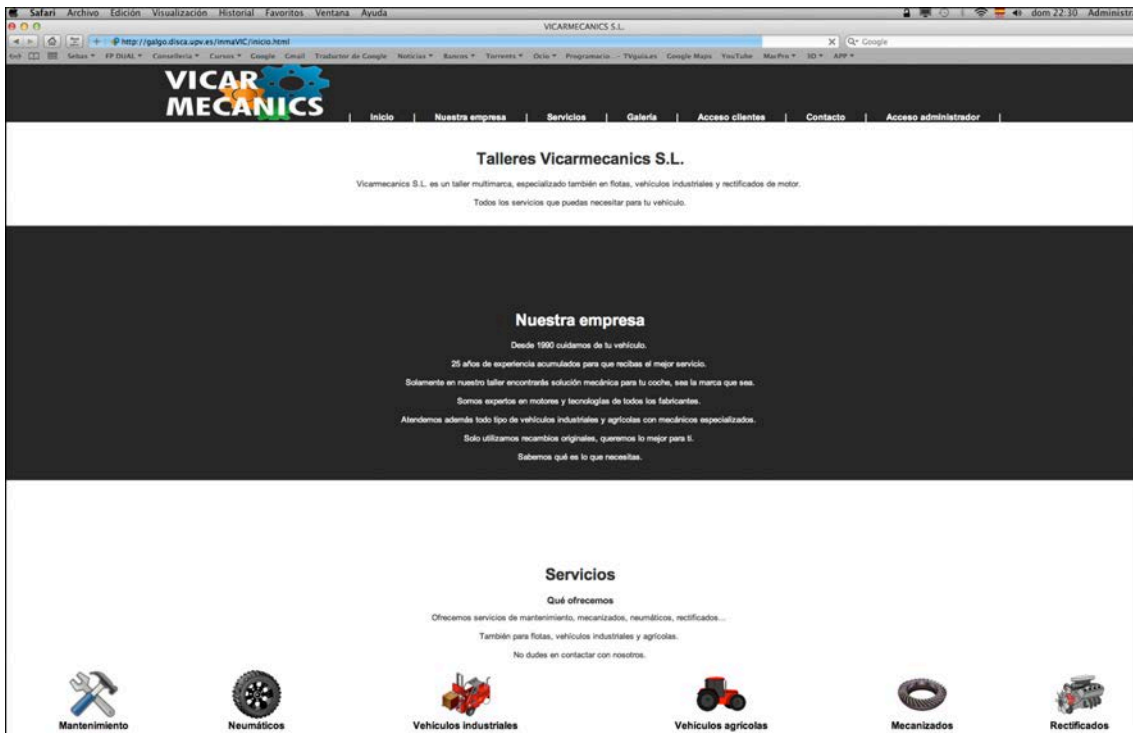


Ilustración 10.4: Navegador Safari.

10.4.- Prueba de seguridad de acceso

Para acceder a la aplicación como usuario registrado, tanto para cliente como para administrador, es necesario insertar el 'usuario' y la 'contraseña' en una pantalla de validación como la que se muestra en la ilustración 10.5. En caso de ser correcto el registro, nos reenviará a la página inicial de la parte privada según corresponda, ya sea la parte privada de los clientes o la parte privada del administrador del taller de vehículos.

The image shows a dialog box titled 'Identificación requerida' with a blue question mark icon. The text inside reads: 'http://galgo.disca.upv.es está solicitando un nombre de usuario y una contraseña. El sitio dice: "Identificación de Vicarmecanics"'. Below this text are two input fields: 'Nombre de usuario:' and 'Contraseña:'. At the bottom right of the dialog are two buttons: 'Cancelar' and 'Aceptar'.

Ilustración 10.5: Validación de usuario.

11. Plazos y coste real

A continuación, se describen los plazos reales que se han necesitado para llevar a cabo el desarrollo de la aplicación Web VICARMECANICS S.L., así como el coste real que este desarrollo ha supuesto.

Como podemos observar en la ilustración 11.1, que muestra el desarrollo de fases y tareas del proyecto, y en la ilustración 11.2, del diagrama de Gantt, podemos afirmar que el tiempo destinado a la realización del proyecto ha sido mayor que el estimado. Este cambio se ha debido al incremento de un día en la creación del prototipo y a los dos días más empleados en la fase de codificación o implementación. Sin embargo, como inicialmente el plazo de entrega había sido fijado antes de la finalización objetivo, el proyecto sigue estando dentro del plazo adecuado y este incremento no ha afectado a la entrega en tiempo del proyecto. Lo que sí ha sido afectado por este incremento en el tiempo ha sido el presupuesto final del proyecto. Sin embargo, este coste es asumible y acordado con el cliente, por lo que podemos afirmar que el proyecto se ha realizado en tiempo y coste correctos. El coste real de proyecto asciende a 3.920€, tal y como muestra la ilustración 11.3. y el tiempo de trabajo necesario ha sido de 49 días de trabajo.

Aplicación web para gestión integral de talleres de vehículos.

Task Name	Duration	Start	Finish
▣ Especificación de requisitos	16 days	Wed 20/01/16	Wed 10/02/16
Estudio previo y entrevistas con el cliente	6 days	Wed 20/01/16	Wed 27/01/16
Identificación de requisitos de usuario	4 days	Thu 28/01/16	Tue 02/02/16
Identificación de requisitos de rendimiento y HW/SW	2 days	Wed 03/02/16	Thu 04/02/16
Identificación de los recursos de la interfaz	2 days	Fri 05/02/16	Mon 08/02/16
Preparar la documentación del sistema	2 days	Tue 09/02/16	Wed 10/02/16
▣ Modelado conceptual	10 days	Thu 11/02/16	Wed 24/02/16
Construcción del modelo	5 days	Thu 11/02/16	Wed 17/02/16
Diseño de la Base de Datos	1 day	Thu 18/02/16	Thu 18/02/16
Creación de un prototipo	2 days	Fri 19/02/16	Mon 22/02/16
Aprobación prototipo por parte de cliente	2 days	Tue 23/02/16	Wed 24/02/16
▣ Codificación	18 days	Thu 25/02/16	Wed 23/03/16
Implementación de la aplicación	18 days	Thu 25/02/16	Wed 23/03/16
▣ Pruebas	5 days	Thu 24/03/16	Fri 01/04/16
Pruebas unitarias de los módulos	3 days	Thu 24/03/16	Wed 30/03/16
Pruebas de integración de los componentes	1 day	Thu 31/03/16	Thu 31/03/16
Pruebas de navegadores	1 day	Fri 01/04/16	Fri 01/04/16
Pruebas de seguridad de acceso	1 day	Fri 01/04/16	Fri 01/04/16
Pruebas validación de enlaces	1 day	Fri 01/04/16	Fri 01/04/16

Ilustración 11.1: Plazo real. Fases y tareas.

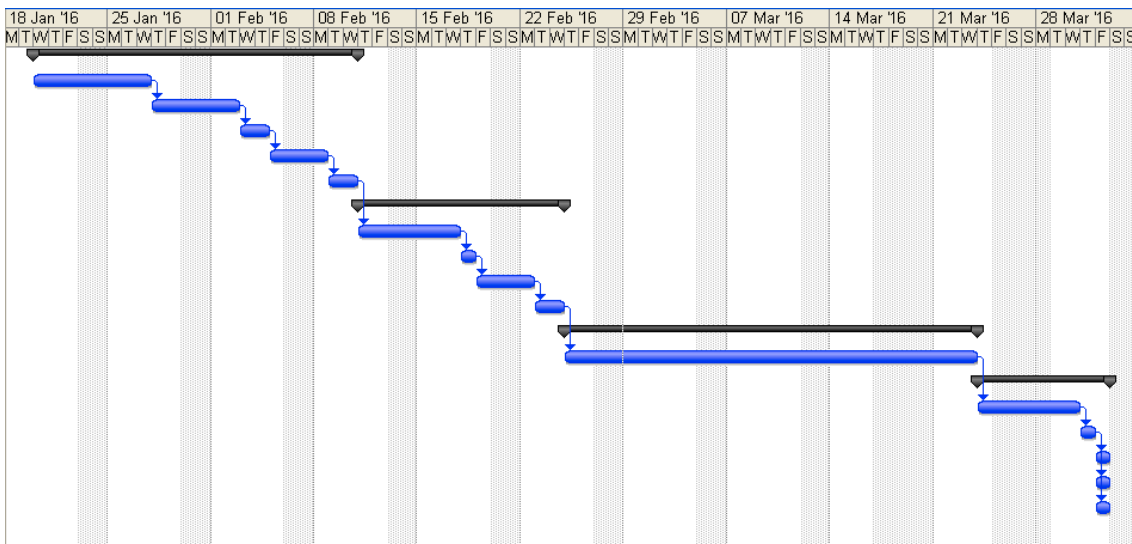


Ilustración 11.2: Plazo real. Diagrama de Gantt.

Project Statistics for 'plazos y coste real'				
	Start		Finish	
Current	Wed 20/01/16		Fri 01/04/16	
Baseline	NA		NA	
Actual	NA		NA	
Variance	0d		0d	
	Duration	Work	Cost	
Current	49d	392h	3.920,00 €	
Baseline	0d?	0h	0,00 €	
Actual	0d	0h	0,00 €	
Remaining	49d	392h	3.920,00 €	

Percent complete:

Duration: 0% Work: 0%

Ilustración 11.3: Coste real del proyecto.

Aplicación web para gestión integral de talleres de vehículos.

12. Conclusiones

12.1. Trabajo realizado

En este Trabajo de Final de Grado se ha diseñado e implementado una aplicación Web llamada VICARMECANICS S.L.

VICARMECANICS S.L. ofrece al usuario una interfaz intuitiva de fácil manejo y que tiene aspecto cuidado, atractivo, sencillo y homogéneo en todas sus partes y módulos. La aplicación es capaz de realizar la gestión de todos y cada uno de los parámetros referentes a la gestión de un taller de vehículos, así como de dar respuesta a los clientes.

Para llevar a cabo la implementación de esta aplicación Web dinámica se ha utilizado HTML, PHP y SQLite.

Antes de la realización de la aplicación se llevó a cabo una planificación temprana y un estudio de viabilidad para considerar si era posible desarrollarla o no. Una vez concluido que el proyecto era viable y se podía llevar a término, se realizaron todas las fases de desarrollo de una aplicación.

Aunque el coste real de la implementación difiere del previamente planificado debido a un incremento en las horas de trabajo en la fase de desarrollo del prototipo e implementación de la aplicación, este aumento fue asumido por el cliente del proyecto, por lo que el proyecto se pudo concluir de manera adecuada dando lugar a una aplicación Web que cumple satisfactoriamente con las expectativas deseadas y requerimientos numerados en la especificación de requisitos y, por tanto, ofrece un alto nivel de cumplimiento de objetivo.

12.2. Valoración personal

Aunque, sin duda, ha sido un trabajo costoso y que ha supuesto mucho tiempo y esfuerzo, gracias al desarrollo de este proyecto he podido refrescar y poner en práctica todo el ciclo de vida de desarrollo de una aplicación Web. Empezando por la planificación y el estudio del proyecto, la especificación de requisitos, la elección de tecnologías a utilizar, la fase de implementación y hasta la fase de pruebas para que el usuario no tenga problemas de uso.

Además de poner a prueba mi capacidad de trabajo, ya que además de realizar este proyecto trabajo a tiempo completo, también he tenido la oportunidad de aplicar todos los conocimientos que he ido adquiriendo a lo largo de la Ingeniería Técnica, primeramente, y actualmente con el Grado. Asimismo, he conseguido obtener una mayor competencia en las tecnologías utilizadas.

Para concluir, me gustaría decir que estoy satisfecha del proyecto realizado y de todo lo aprendido en el desarrollo del mismo.

12.3. Futuras ampliaciones

El empleo de los estándares en el diseño de la aplicación facilita también a los diseñadores y programadores el desarrollo de futuras ampliaciones en cuanto a funcionalidades se refiere. Así pues, se puede asegurar que VICARMECANICS S.L. es una aplicación Web abierta a mejoras y ampliaciones.

A continuación, se expone un listado de posibles futuras ampliaciones:

- Revisión de la interfaz gráfica.
- Creación de un sistema de banners publicitarios.
- Mejora de los formularios de la aplicación.
- Implementación de una funcionalidad de envío de mensajes al móvil cuando el vehículo ya esté listo para su recogida.
- Generador de archivos .pdf para la impresión de los listados.

Referencias

- [1] [Laboratorio Nacional de Calidad del Software, 2009] “INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA”. Madrid. Ministerio de Industria, Turismo y Comercio.
- [2] Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter General.
- [3] Obligaciones del responsable del fichero que vienen establecidas en la LOPD. : <https://www.agpd.es/portalwebAGPD/canalresponsable/obligaciones/index-ides-idphp.php>
- [4] Obligaciones del encargado del fichero, recogidas en el artículo 12 de la LOPD: https://www.agpd.es/portalwebAGPD/canaldocumentacion/publicaciones/common/pdfs/guia_responsable_ficheros.pdf
- [5] Contrato para el tratamiento de datos personales por cuenta de terceros: <https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjHoY6anr7LAhUHVhQKHSRPANkQFggdMAA&url=http%3A%2F%2Fwww.jprenafeta.com%2Fwp-content%2Fuploads%2F2007%2F12%2FModelo%2520contrato%2520tratamiento%2520datos%2520terceros.rtf&usg=AFQjCNGtqLUIhUKoO-querXLWPHn5d6t95A&sig2=pN8LY4uhS3vll6Irl1DdTA>
- [6] Documentación de la asignatura IDR (ingeniería de requisitos) de la UPV.
- [7] IEEE Std 830 - IEEE Guide to Software Requirements Specifications. IEEE Standards Board. 345 Eas 47 th Street. New York, NY 10017, USA. 1984.
- [8] Guía del IEEE para la Especificación de Requerimientos Software. Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia.
- [9] UML: <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>
- [10] [Glass, 2004] Glass, Michael (2004) “Desarrollo Web con PHP y MySQL.” Madrid. Anaya Multimedia.
- [11] Página oficial PHP <http://www.php.net/>
- [12] Herramienta para validación de enlaces <http://validator.w3.org/checklink>