



CarSpecs

Desarrollo de Aplicaciones Web (DAW)

Carlos Cantos García

Luz María Álvarez Moreno

INDICE

ABSTRACT.....	5
JUSTIFICACIÓN DEL PROYECTO.....	7
OBJETIVOS.....	8
R01 – Login/Registro:	8
R02 – Buscador	11
R03 – Perfil de lista de usuarios.....	15
R04 – Gestión de coches	22
R05 – Gestión de usuarios.....	24
R06 – Proceso Creación de Coches	27
DISEÑOS.....	29
Sección Usuario.....	29
Sección de Administrador	31
Estructura de la base de datos.....	33
TECNOLOGÍAS	35
METODOLOGÍA	38
IMPLEMENTACIONES.....	40
DESPLIEGUE/ INSTALACIÓN	43
Instalación de WAMP.....	43
Configuración de Apache	43
Configuración de MySQL	44
Instalación de PHP	44
Descompresión del archivo del proyecto	44
Configuración del proyecto	45
TRABAJOS FUTUROS.....	46
Cambios en la gestión:	46
Cambios para el usuario:	46
Flexibilidad en el futuro:.....	47
CONCLUSIONES	49
Accesibilidad y Simplicidad:.....	49
Interactividad:.....	49

Uso de Nuevas Herramientas y Métodos:50

REFERENCIAS 51

ABSTRACT

El proyecto consiste en un catálogo de automóviles en el que el usuario puede buscar el automóvil deseado, visualizar información resumida o más detallada, y en caso de tener una cuenta, poder guardar los elementos deseados en listas organizadas de forma arbitraria. Dentro del perfil del usuario, aparte de poder gestionar sus listas, también puede acceder a la gestión de sus datos, pudiendo modificar toda su información, siempre que no cause duplicaciones en la BBDD y también suscribirse realizando un pago a través de redsys en el que dependiendo del plan seleccionado tendrá un periodo determinado de rol VIP el cual le proporciona el beneficio de poder crear un número de listas limitado a 15 originalmente limitado a tres.

Por otra parte, hay una sección de administración en el que se accede mediante las cuentas creadas con el rol de administrador las cuales permiten el acceso a la gestión de usuarios y la gestión de coches.

- **Gestión de usuarios:**

Los administradores pueden filtrar y buscar usuarios, modificar su suscripción o desactivar a los usuarios. También se muestra un gráfico con los registros del número de suscriptores, no suscriptores y el total de cuentas en la base de datos.

- **Gestión de coches:**

Los administradores pueden buscar y filtrar los automóviles guardados, modificar su información y crearlos de forma manual o automática utilizando un script de Python.

El proyecto estará creado usando PHP con la librería de Codeigniter en su versión 3, JavaScript, Python para los scripts, y las librerías mencionadas

en la Bibliografía y usando el MVC (Modelo, Vista, Controlador), para mantener una estructura definida.

This project consists of a catalog of automobiles where the user can search for the desired automobile, visualize a resume of its information or a more detailed text about the element, and in case of having an existing account, the user could save those elements in lists organized at will. Inside the user account, Within the user's profile, aside from managing their lists, they can also access their data management section, allowing them to modify all their information, as long as it does not cause duplications in the database. Additionally, users can subscribe by making a payment through Redsys, where, depending on the selected plan, they will have a specific VIP role period. This VIP role provides the benefit of creating up to 15 lists, which is originally limited to three.

On the other hand, there is an administration section accessible through accounts with the administrator role. This section allows access to user management and car management.

- **User management:**

Administrators can filter and search for users, modify their subscription, or deactivate them. It also displays a graph showing the number of subscribers, non-subscribers, and total accounts in the database.

- **Car management:**

Administrators can search and filter stored cars, modify their information, and create cars manually or automatically using a Python script.

The project will be created using PHP with the CodeIgniter library version 3, JavaScript, Python for the scripts, and the libraries mentioned in the Bibliography and using MCV (Model, View, Controller) in order to keep a defined structure.

JUSTIFICACIÓN DEL PROYECTO

La idea nace de un estudio anterior acerca de las herramientas que se pueden usar para encontrar datos sobre automóviles, como la página Auto-Data. Estas herramientas suelen tener mucha información en tablas fijas, con interfaces complicadas y no ofrecen funciones interactivas para los usuarios. Durante el análisis preliminar de estas páginas, se detectaron algunos problemas importantes:

- **Interfaz abrumadora:** A pesar de tener toda la información, la disposición de la misma no facilita la visualización de la información ni la navegación por las diferentes páginas.
- **Falta de adaptación:** Los usuarios carecen de alternativas para participar más allá de revisar la información. Características como guardar vehículos favoritos, realizar comparaciones personalizadas o aplicar filtros actualmente no están disponibles.
- **Falta de valor agregado:** algunas páginas web solo permiten iniciar sesión para obtener un estatus sin beneficios reales, dejando de lado la posibilidad de ofrecer una experiencia enriquecedora para el usuario.

Estos problemas muestran una falta en el mercado: la ausencia de una interfaz simple con capacidades que dejen a los usuarios compartir, adaptar y manejar la información de manera más efectiva.

De acuerdo con esta evaluación inicial realizada, se optó por crear un inventario de vehículos que no solo resuelva estos obstáculos, sino que también integre artilugios contemporáneos y un diseño de navegación más ordenado. De este modo, la propuesta sobresale como una opción original en el ámbito de las herramientas de búsqueda de información para automóviles.

OBJETIVOS

El objetivo principal es la realización de una página web que permita de forma ágil y sencilla disponer de un buscador de coches, que permita a los usuarios


Para poder cumplir con el objetivo principal se han definido una serie de requisitos y tareas que se indican a continuación:

R01 – Login/Registro:

Debe disponer de un Inicio de Sesión, donde poder acceder al Registro también, y en caso de no tener la contraseña, que se mande un email al correo del usuario donde se redireccionará al restablecimiento de la contraseña.

- **R01F01 – Registro**

Disponer de un menú para el registro del usuario en la aplicación.

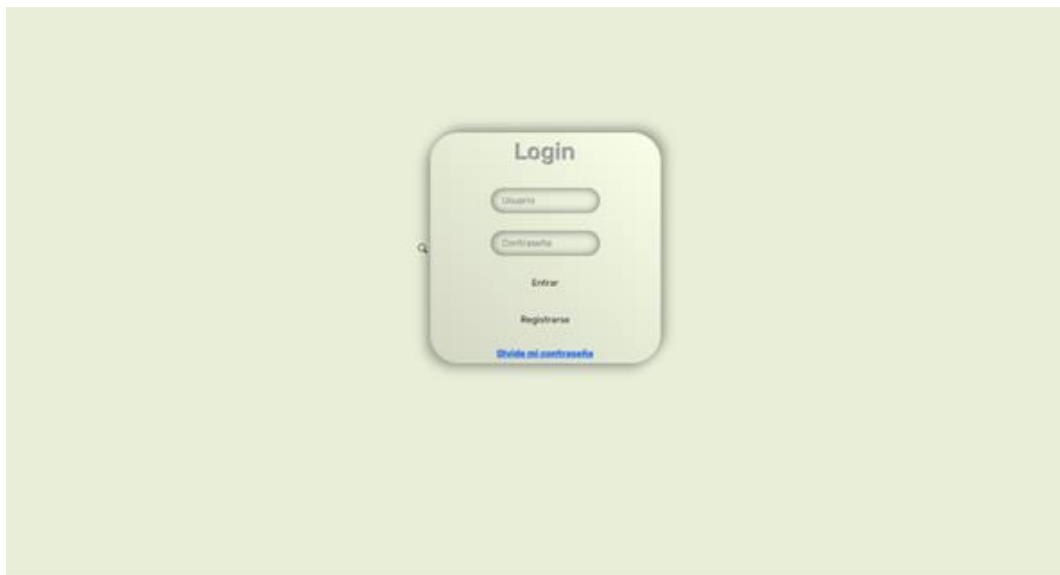
A screenshot of a 'Sign Up' form. The form is a light green rounded rectangle centered on a darker green background. It contains the title 'Sign Up' at the top, followed by three input fields labeled 'Usuario', 'Correo', and 'Contraseña'. At the bottom of the form is a 'Submit' button.

- **R01F01T01** - Creación del formulario de registro junto al controlador que envíe la información al modelo, donde se

gestionará la información introducida enviando la información a la respectiva tabla usuarios de la base de datos.

- **R01F01T01P01** - Comprobar que la información que se manda al modelo está completa y validada.
- **R01F01T02** - Creación de las tablas usuarios, roles, statuses, lista_usuario, lista_cars, siendo estas dos últimas para poder asignar las listas a cada usuario y a cada lista los coches que deben contener de forma de que toda información que sea común a todos los usuarios, en este caso los roles y los status estén indexados para mejor optimización.
- **R01F01T02P01** - Mostrar las tablas y comprobar que la información se guarda correctamente.
- **R01F02- Inicio de Sesión**

Disponer de una pantalla de inicio de sesión del usuario.



- **R01F02T01** - Creación del formulario de inicio de sesión, junto a las queries en el modelo correspondiente para comprobar las credenciales introducidas si las credenciales son correctas, se

creará una sesión de codeigniter donde se guardará la id del usuario, su status y su rol. En caso de tener status inactivo, no inicia sesión y si el rol es admin, redireccionará al menú de administración.

- **R01F02T01P01**- Comprobar que la información que se manda al modelo está completa y validada y que redirecciona correctamente dependiendo del rol del usuario.

- **R01F03 - Recuperación de contraseña.**

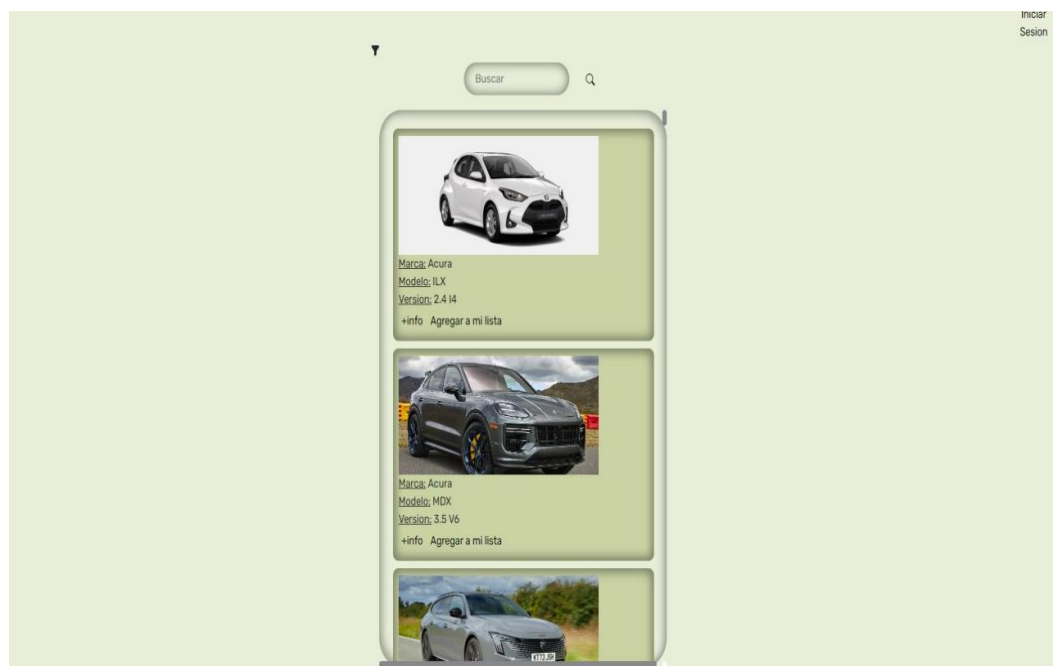
Disponer la posibilidad de recuperación de la contraseña para el inicio de sesión.

- **R01F03T01** - Uso de la librería PhpMailer , usando el servidor smtp de google y una cuenta configurada, para mandar al correo introducido en una vista que contiene solo un input con para poner el correo del usuario. El correo contendrá un enlace con la vista desde la que se puede restablecer la contraseña, y concatenado usando un json_encode la identificación del usuario codificado en base64 para evitar problemas de caracteres no permitidos en la URI. Desde dicha página se recoge mediante GET la identificación, se hace un json_decode para leer la identificación y al crear la nueva contraseña, se ejecuta la query que la modifica en la BBDD.
- **R01F03T01P01** - Prueba de uso siguiendo el flujo normal del usuario comprobando que el mail se manda con el formato deseado y con la información requerida.

R02 – Buscador

La web dispondrá de un buscador de coches, que será la vista predeterminada, mostrando unos elementos que contienen la información resumida del coche, y mediante botones dentro de cada elemento, permite ver un modal con la información detallada. Se debe implementar un buscador y unos filtros. Desde el buscador se podrá acceder al inicio de sesión, y en caso de estar ya iniciado, dar acceso al perfil.

- **R02F01 - Mostrar la información de los coches de forma organizada con un buscador y filtros.**



- **R02F01T01** - Crear una tabla con los campos necesarios que contengan la información necesaria como son la tabla cars, brands, models, etiqueta y combustibles.
- **R02F01T01P01**-Mostrar las tablas creadas.

- **R02F01T02** - Crear la vista, que recibirá la información y creará los elementos iterando sobre cada conjunto recibido, con los botones de cada elemento, y los botones que permitan el acceso a otras secciones.
 - **R02F01T02P01**-Mostrar vistaBuscador.php.
- **R02F01T03** - Crear el modelo, que lanzará las consultas a la BBDD para extraer los conjuntos de información.
 - **R02F01T03P01**- Comprobar que las queries extraigan la información pedida mediante un `var_dump()` para mostrar todos los registros y los tipos de datos extraídos.
- **R02F01T04**- Crear el controlador que comunicara la vista y el modelo
 - **R02F01T04P01** - Comprobar que la información llega a vistaBuscador.php y se muestre correctamente con el formato deseado.
- **R02F02 - Crear el buscador y los filtros**

Con la creación del buscador se crean además los filtros que van a permitir realizar la búsqueda en función de los valores que el usuario introduzca. Se podrá seleccionar por los siguientes atributos:

- Combustible
- Potencia
- Etiqueta
- Ordenar por

Filtros ✕

- Combustible
- Potencia:
 - Potencia mínima (€
 - Potencia máxima (€
- Etiqueta
- Ordenar por:
 -

Aplicar Filtros

- **R02F02T01** - Creación de los inputs correspondientes.
 - **R02F02T01P01** - Comprobación de que los input's envían la información correctamente en el formato deseado.
- **R02F02T02** - Dependiendo del input, que contiene el contenido del buscador y de los filtros seleccionados, el modelo lanza una query diferente mediante if()else() para insertar diferentes secciones de query conforme a la búsqueda solicitada para extraer la información requerida permitiendo así el uso simultáneo de filtros y búsquedas de texto.
 - **R02F02T02P01** - Comprobar que la información saliente es la solicitada.

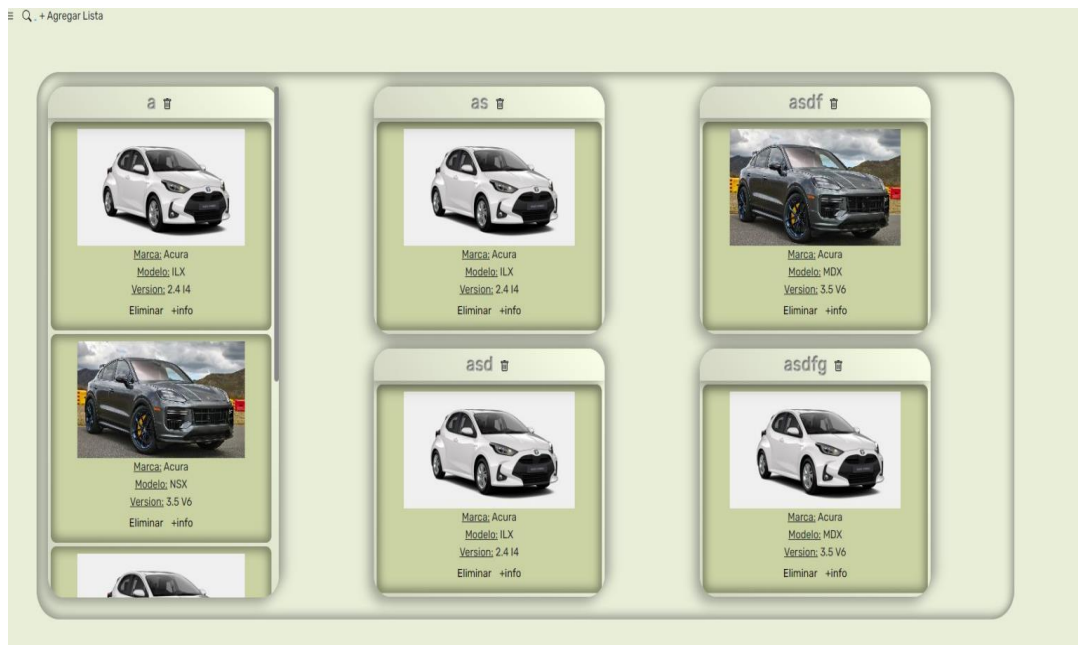
- **R02F03 – Añadir a la lista y enlace inicio de sesión y perfil**

Creación de botones de navegación a otras secciones como los modales de información detallada, añadir a la lista, enlace al Inicio de sesión y al perfil.

- **R02F03T01** - A cada elemento, se le asigna un input hidden con el id del coche que muestra, de forma que al darle a mostrar más, se manda esa información al modelo y la información del coche seleccionado se muestra en el modal.
 - **R02F03T01P01** - Comprobación de que se muestra la información correcta en el modal.
- **R02F03T02** - Creación de los botones que solo contienen enlaces. En caso de estar iniciado, se mostrará el botón de enlace al perfil, en su defecto, se mostrará el de Iniciar sesión.
 - **R02F03T02P01** - Comprobación de que los enlaces sean correctos y se muestre en el contexto que deban.
- **R02F03T03** - Creación del botón que permite **añadir el elemento a la lista que el usuario desee** mostrando las listas del usuario iniciado y ejecutando la query pertinente en el modelo con la información del coche a añadir y la lista destino.
 - **R02F03T03P01** - Comprobar en la BBDD que las instancias se han creado correctamente.

R03 – Perfil de lista de usuarios

Un usuario podrá disponer de varias listas donde vaya guardando los coches seleccionados, por lo que debe haber un perfil, donde se muestren dichas listas de usuario y se puedan crear, eliminar o modificar esas listas, teniendo un límite de 3 listas a excepción de ser **VIP**, en cuyo caso se aumentará ese límite a 15, se pueda acceder a sección donde modificar la información del usuario, cerrar sesión y acceso a la pasarela de pago para ser usuario VIP.



- **R03F01 - Interfaz con las listas y su contenido propias del usuario que esté iniciado.**
 - **R03F01T01** - Crear la vista, el controlador y usar el modelo con las queries relacionadas con los coches y otro con los usuarios, hago la estructura de la interfaz y a través del controller le paso las listas y los coches correspondientes a la vista recogidas del modelo

también se incluye en la interfaz los botones de enlace a editar el perfil y a la pasarela de pago.

- **R03F01T01P01** - Mostrar vistaCoches.php y comprobar que toda la información esté bien estructurada
- **R03F02 - Eliminar coches de las listas**
 - **R03F02T01** - Usando el mismo código que al mostrar el modal, recojo la id del coche seleccionado y la lista en la que está y lo eliminó de dicha lista. Al eliminar usó un método usando secciones de vistas y JavaScript para, de forma dinámica con ajax, insertar en donde sea necesario la actualización de información requerida, este método al ser reutilizado en varias ocasiones lo llamaré **método de refresco** para abreviar.
 - **R03F02T02P01** - Trato de usar el botón que elimina el coche de la lista
- **R03F03 - Agregar y eliminar listas.**
 - **R03F02T01** - Para agregar listas haré un botón donde te deje elegir el nombre de la lista, la asignará a tu usuario en la BBDD y accionara el **método de refresco** para mostrar la nueva lista.
 - **R03F02T01P01** - Comprobación del buen funcionamiento del botón, de la actualización visual y de la generación de un nuevo registro en las tablas pertinentes de la base de datos
 - **R03F02T02** – Para eliminarlas, cada lista tendrá asignada su id, para que a la hora de darle al botón que acciona el proceso, se le pase a dicho proceso el dato que identifica a dicha lista.

- **R03F04 - Cerrar sesión**

- **R03F04T01** - Para cerrar sesión habrá un botón que cerrará la sesión de codeigniter donde está guardada la identificación del usuario actual y redireccionará al buscador.
- **R03F04T01P01** - Se comprobará el buen funcionamiento del botón.

- **R03F05 - Editar Perfil**



Datos de Usuario

Perfil

- Nombre Usuario :fgh Cambiar
- Email :fgh@fgh.com Cambiar
- Rol :NO VIP

Cambiar Contraseña Suscribirse

- **R03F05T01** - Crear una vista accesible desde el perfil donde se mostrará la información no sensible y se permitirá modificar toda la información mediante el correspondiente input. Una vez se mande la información del input, se comprobarán duplicaciones, en caso de causar alguna cancela la operación y avisa al usuario, si el campo cambiado es la contraseña no se comprobará nada, solo se le aplicará la validación creada para ello. En esta sección también se

mostrará la duración de la suscripción actual en caso de ser VIP, si no se es VIP, habrá un enlace a la pasarela de pago.

- **R03F05T01P01** - Se comprueba que todos los input funcionen correctamente y que en caso de duplicaciones se cancele la solicitud y salga un mensaje de error.

- **R03F06 - Pasarela de pago**

Para la suscripción de un plan es necesario disponer de la pasarela de pago. Se podrá seleccionar el plan de suscripción entre uno de los definidos:

The screenshot shows a web interface with a light green background. At the top, the title "Elige tu plan de suscripción" is centered. Below the title, there are three rectangular buttons arranged horizontally. Each button contains a price and a duration. The first button shows "5€" and "7 días". The second button shows "17.99€" and "30 días". The third button shows "180€" and "1 año".

Plan	Price	Duration
Plan 1	5€	7 días
Plan 2	17.99€	30 días
Plan 3	180€	1 año

Una vez seleccionada el plan al que se quiere suscribir se presentará una pantalla con la información resumen de dicho plan.

Subscripción Elegida:

17.99€ [30 dias]

Que incluye:

Agregar listas sin
ningun limite

Realizar Pago

Al pinchar en Realizar Pago, se abrirá la pasarela de pago:

The screenshot displays the Redsys payment gateway interface. On the left, a table titled 'Datos de la operación' (Operation Data) provides transaction details. On the right, the 'Pagar con Tarjeta' (Pay with Card) section shows 'Métodos de pago guardados' (Saved payment methods) with a Visa card ending in 4602. Below this, there is an option to 'Utilizar otra tarjeta' (Use another card) with a radio button and a text field. At the bottom of this section are 'Cancelar' (Cancel) and 'Pagar' (Pay) buttons. Further down, the 'Otras formas de pago' (Other payment methods) section shows a Google Pay button. The footer includes logos for various payment methods (Visa, Mastercard, American Express, JCB, etc.) and the Redsys logo.

Datos de la operación	
IMPORTE	17,99 €
Comercio:	RESC ESPAÑA (SPAIN)
Terminal:	999008881-1
Pedido:	350814
Fecha:	08/01/2025 23:28

Pagar con Tarjeta

Métodos de pago guardados

Visa ****4602 12/34

Utilizar otra tarjeta

☐ Utilizar/Añadir otra tarjeta

Cancelar Pagar

Otras formas de pago

Google Pay

Visa Mastercard AMERICAN EXPRESS JCB iD Check SafeKey

Redsys

© 2025 Redsys Servicios de Procesamiento, S.L. - Todos los derechos reservados. [Política de cookies](#)

Una vez finalizado el pago, se mostrará el resultado de dicha transacción.

Datos de la operación	
IMPORTE	17,99 €
Comercio:	RESC ESPAÑA (SPAUN)
Terminal:	999008881-1
Número pedido:	350814
Fecha:	09/01/2025 00:05

✓ OPERACIÓN AUTORIZADA CON CÓDIGO: 214022

Número Tarjeta: *****4602

Url Comercio: <http://www.redsys.es/canales.jsp>

Continuar

Powered by Redsys

(c) 2025 Redsys Servicios de Procesamiento, SL - Todos los derechos reservados.

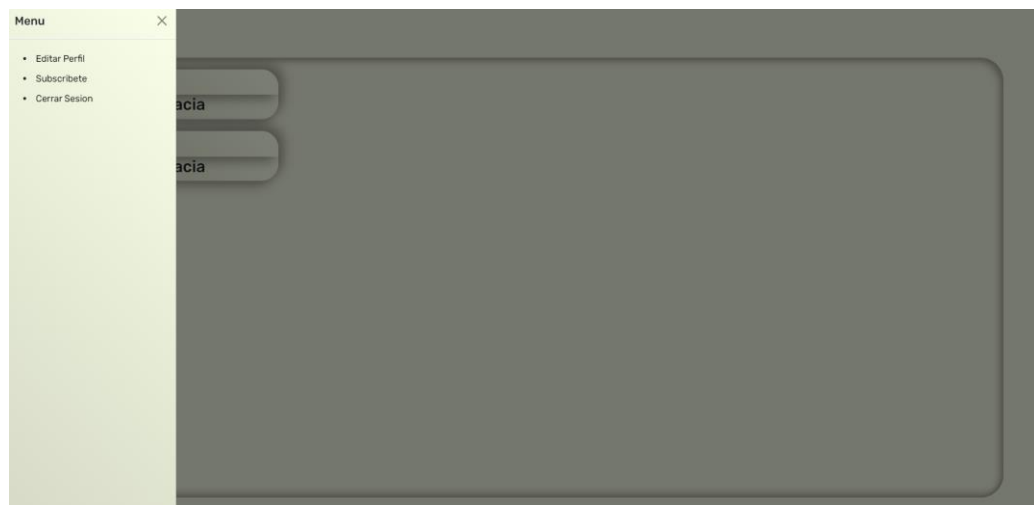
Pago realizado con éxito

[Volver al Perfil](#)

- R03F05T01** - Crear una vista accesible desde el perfil en el que se muestran las suscripciones disponibles en ese momento, con sus precios y periodos , al seleccionar alguna, se redirecciona a una página de confirmación donde se guardan los datos de la suscripción elegida, para que al confirmar el pago y hacer la redirección de redsys, se envíen los datos que pide la API los cuales son el precio, id de la orden, la moneda, la id del mercado, el tipo de transacción y las urls de las vistas de salida, tanto como para el todo ok como para la de error con esta información se crean los parámetros firma y params qué es lo que recibe la API de

redsys,. Una vez en la página de redsys se procederá con el pago, en caso de error, redirecciona a una vista donde sale un aviso de que ha habido un error, si el pago ha sido efectuado correctamente, se redireccionará a una vista de confirmación de que todo ha salido correctamente y habrá un botón que enviará al usuario a la vista de modificar el perfil.

- **R03F05T01P01** - Comprobar que el funcionamiento del proceso es correcto con el porte seleccionado y el periodo estipulado en la suscripción usando la tarjeta de prueba proporcionada por redsys 4918019160034602 12/34 123 y comprobar que cualquier error está recogido correctamente y que redireccione a las páginas que están establecidas para cada caso.



R04 – Gestión de coches

La Gestión de Coches debe mostrar la información detallada de cada coche en una lista con paginación, permitir la búsqueda y filtrado de los coches, modificar su información, y crear nuevos coches, marcas y modelos de forma manual y automática.

Gestion de Coches

Volver

ID Buscar Combustible: Etiqueta:

Prueba API

[Crear coche](#) [Crear Modelo](#) [Crear Marca](#)

ID	Marca	Modelo	Version	Combustible	Cosumo medio	Potencia	Torque	Peso	Etiqueta	Likes	Puntos	Seguridad	Compras	Ruta	Foto
1	Acura	ILX	2.4 I4	Gasolina	8	201	180	3095	No	0	30	24617	img/5.jpg		Editar Informacion
2	Acura	MDX	3.5 V6	Gasolina	10	290	267	4032	No	0	57	41774	img/3.jpg		Editar Informacion
4	Acura	RDX	2.0 I4	Gasolina	10	272	280	3783	No	0	37	35890	img/4.jpg		Editar Informacion
5	Acura	MDX	3.5 V6	Gasolina	10	310	272	3977	No	0	49	50947	img/5.jpg		Editar Informacion
5	Acura	TLX	2.4 I4	Gasolina	9	206	182	3505	No	0	38	31343	img/5.jpg		Editar Informacion
7	Alfa Romeo	MDX	1.7 I4	Gasolina	8	237	258	2487	No	0	51	64646	img/5.jpg		Editar Informacion
3	Alfa Romeo	Giulia	2.0 I4	Gasolina	9	280	306	3522	No	0	37	37463	img/3.jpg		Editar Informacion
7	Alfa Romeo	Stelvio	2.0 I4	Gasolina	9	280	306	3901	No	0	42	39378	img/4.jpg		Editar Informacion
10	Bugatti	MDX	4.0 V8	Gasolina	12	503	513	4134	No	0	0	0	img/4.jpg		Editar Informacion
11	Bugatti	MDX	5.2 V12	Gasolina	14	715	663	0	No	0	0	0	img/4.jpg		Editar Informacion
12	Bugatti	MDX	4.0 V8	Gasolina	12	503	505	0	No	0	0	0	img/3.jpg		Editar Informacion
13	Audi	A3	2.0 I4	Gasolina	8	184	221	3197	No	0	21	31304	img/2.jpg		Editar Informacion
14	Audi	A4	2.0 I4	Gasolina	8	188	236	3417	No	0	22	35156	img/4.jpg		Editar Informacion
15	Audi	A4 allroad	2.0 I4	Gasolina	9	248	273	3847	No	0	23	41924	img/3.jpg		Editar Informacion
16	Audi	A5	2.0 I4	Gasolina	9	248	273	3725	No	0	21	40327	img/2.jpg		Editar Informacion
22	Audi	Q3	2.0 I4	Gasolina	11	228	258	3905	No	0	20	32620	img/2.jpg		Editar Informacion
23	Audi	Q5	2.0 I4	Gasolina	10	248	273	4045	No	0	24	40704	img/4.jpg		Editar Informacion

[123>Last<](#)

- **R04F01 - Mostrar la información de los coches**

- **R04F01T01** - Mostrar información sacado de la BBDD mediante una query en el modelo de coches en la vista creada para este fin estando la información organizada y bien estructurada.
- **R04F01T01P01** - Comprobación de que la estructura esté correcta y la información se muestre correctamente donde debe estar organizada.

- **R04F02 - Buscador y filtros**
 - **R04F02T01** - Reutilizar el buscador y filtros de la página inicial pero mostrando en esta ocasión toda la información detallada directamente.
 - **R04F02T01P01** - Comprobación de que el buscador devuelve la información con los parámetros pedidos en los inputs.
- **R04F03 - Crear marcas , modelos y coches.**
 - **R04F03T01** - Crear los botones correspondientes, en el caso de marcas y modelos, se crean registros en sus respectivas tablas, y al crear coches, se despliega un modal que en el que introducir los datos a crear, los datos que estén ya definidos se muestran mediante un <select> cuyas opciones se generan dinámicamente parseando las tablas en las que esta almacenadas esas opciones. Al elegir una marca, el <select> del modelo se actualiza para mostrar solo las opciones que sean compatibles con esas marcas. Al crear o modificar cualquier cosa se actualiza mediante el **método de refresco**.
 - **R04F03T01P01** - Comprobar que al usar todos los input se creen los registros correspondientes
- **R04F04 - Paginación**
 - **R04F04T01** - Uso el método de paginación que me da codeigniter 3 el cual requiere del número de elementos a mostrar y de un número de referencia, información que se pasa al controller para que se por una parte la información extraída tenga un LIMIT y para que se cree la paginación que permita avanzar o retroceder en los registros.

- **R04F04T01P01** - Comprobación de que la paginación funcione correctamente y no se salten registros

R05 – Gestión de usuarios

La Gestion de Usuarios debe mostrar la información de los usuarios a excepción de la contraseña, y permitir la búsqueda y filtrado de los usuarios, modificar su suscripción y desactivar usuarios y mostrar la gráfica con el avance mensual de los usuarios respecto al número de suscriptores.

Gestion de Usuarios

Volver

ID

Nombre de Usuario

Email

Roles : Todos

Status : Todos

Buscar

Prueba Excel

ID	Nombre	Email	Rol	Fecha_VIP	Status
1	asd	asd@asd.com	VIP	2025-02-14	Activo Desactivar Quitar VIP
2	qwe	qwe@qwe.com	VIP	2024-12-06	Activo Desactivar Quitar VIP
23	fgh	fgh@fgh.com	NO VIP		Activo Desactivar Hacer VIP
25	Carlos	Admin@admin.com	ADMIN		Activo
26	Quinn Lloyd	ategestas.a@protonmail.org	NO VIP		Activo Desactivar Hacer VIP
27	Holly Woodwa	quis@outlook.couk	NO VIP		Activo Desactivar Hacer VIP
28	Calvin Campo	pede.suspendisse@outlook.edu	NO VIP		Inactivo Activar Hacer VIP
29	Orii Hurley	sollicitudin.commodo@aol.ca	VIP	2025-11-14	Activo Desactivar Quitar VIP
50	Idona Vega	velit,justo.nec@outlook.com	NO VIP		Inactivo Activar Hacer VIP
51	Ava Walton	elementum.lorem@aol.couk	VIP	2024-11-16	Activo Desactivar Quitar VIP
52	Michelle Sto	neque.sed@aol.org	NO VIP		Inactivo Activar Hacer VIP
53	Kylee Allen	mauris@icloud.com	NO VIP		Activo Desactivar Hacer VIP
54	Allistair Ev	nec.leo@google.couk	NO VIP		Activo Desactivar Hacer VIP
55	Malik Stark	fusce.mollis.duis@outlook.net	VIP	2024-11-16	Inactivo Activar Quitar VIP
56	Shaine Ramse	magna.nec@yahoo.net	NO VIP		Activo Desactivar Hacer VIP
57	Brenna Snide	quam@outlook.ca	NO VIP		Inactivo Activar Hacer VIP
58	Howard Camer	vestibulum.mauris.magna@google.ca	NO VIP		Inactivo Activar Hacer VIP
59	Amber Mcinto	litora@icloud.edu	NO VIP		Inactivo Activar Hacer VIP

[123>Last <](#)

- **R05F01 - Mostrar información, buscador y filtros.**
 - **R05F01T01** - Extraer la información de la BBDD a través del modelo con las queries de usuario, exceptuando la contraseña y mostrarla en la vista creada para este fin estando la información organizada y bien estructurada.
 - **R05F01T01P01** - Comprobación de que la estructura esté correcta y la información se muestre correctamente donde debe estar organizada.

- **R05F01T02** - Reutilizar el método de búsqueda y filtros que con los coches, pero cambiar los filtros para adecuarlos a los datos que puedan tener los usuarios.
 - **R05F01T02P01** - Comprobación de que el buscador devuelve la información con los parámetros pedidos en los inputs.
-
- **R05F02 - Modificar información del usuario.**
 - **R05F02T01** - Para modificar la información del usuario se crean dos grupos de botones , el primer grupo cambiará el rol, solo en caso de no ser admin el cual será inmutable, si se es VIP, se mostrará el botón de eliminar VIP y viceversa, al hacer VIP requiere de un input en el que poner hasta cuando se le da VIP a ese usuario que al pasarlo al controller hay que darle el formato preciso para que se guarde exitosamente en la base de datos, el otro grupo modifica el status, si el usuario es activo, se muestra el de desactivar , y cuando es inactivo muestra el de activar. Todo esto de forma dinámica tal que si lo cambias se muestra el cambio al momento y por supuesto también cambiaría el botón. Cada modificación se ejecuta a través de las queries en el modelo.
 - **R05F02T01P01** - Comprobar que la interfaz se actualice correctamente y que los registros en la base de datos estén correctos.

- **R05F03 - Grafica.**

Se dispondrá de un resumen gráfico de los datos de las suscripciones.



- **R05F03T01** - Proporcionando los datos necesarios mediante el controlador pasándolo a formato json debido a que la librería de Highcharts solo admite ese formato y utilizando código proporcionado por dicha librería y modificándolo para adecuarlo a mis necesidades, para que por cada registro en la BBDD se cree una serie de columnas mostrando la información referente a cada mes.
- **R05F03T01P01**-Mostrar la gráfica en la vista `GestionUsuarios.php` y comprobar que se actualice de forma dinámica y sin intervención de forma periódica.

R06 – Proceso Creación de Coches

Script para automatizar el proceso de creación de coches lo máximo posible

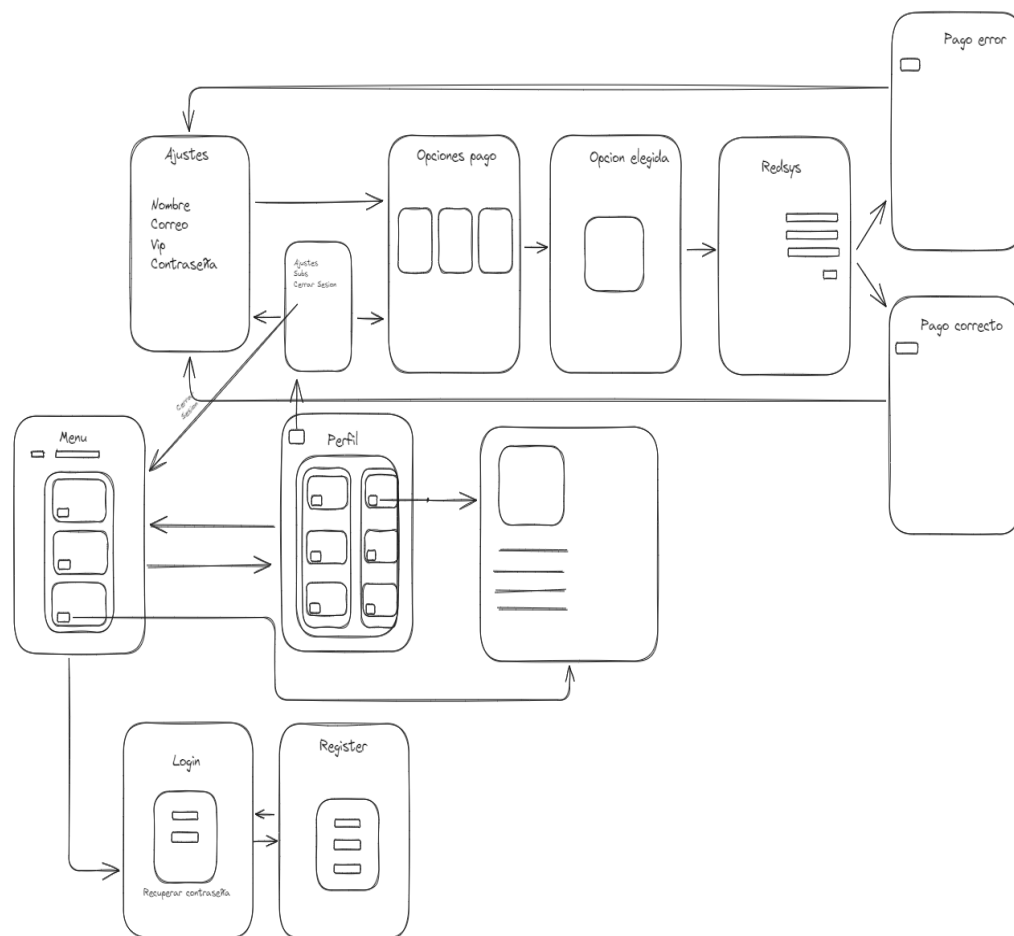
- **R06F01 - Creación de diferentes métodos en caso de que uno no sea viable en ese momento queda mencionar que es un requisito que la página este en español, por ejemplo si la página es <https://www.auto-data.net/en/seat-leon-i-1m-fr-1.9-tdi-150hp-13613> , seria invalido y habria que cambiar el .../en/... por un .../es/... .**
- **R06F01T01 -** Uso de la interfaz propia de la página mediante selenium para acceder a la sección de gestión de coches a través de las etiquetas html e introducir los datos en el formulario destinado a crear nuevos coches
- **R06F01T01P01 -** Comprobación de que la información se recoge correctamente, se le da el formato correcto y se adecua al formulario destino, la automatización navega a través de la página sin problemas e introduce los datos en los inputs o cambia los selects pertinentes, y, finalmente el coche es agregado en la base de datos.
- **R06F01T02 -** Uso de la librería request para mandar navegar por la página destino hasta la gestión de coches y mandar una petición POST con toda la información con el formato correcto
- **R06F01T02P01 -** Comprobación de que la información se recoge correctamente, se le da el formato correcto y se adecua al formulario destino, la automatización navega a través de la página sin problemas, hace las peticiones en el formato correcto sacando de la base de datos los índices correspondientes a la información extraída, y , finalmente el coche es agregado en la base de datos.

- **R06F01T02** -Uso de la librería mysql connector introducir directamente la información en la base de datos
- **R06F01T02P01** - Comprobación de que la información se recoge correctamente en el formato correcto sacando de la base de datos los índices correspondientes a la información extraída, y, finalmente el coche es agregado en la base de datos.

DISEÑOS

A continuación, se muestran los diseños realizados durante la ejecución del proyecto.

Sección Usuario

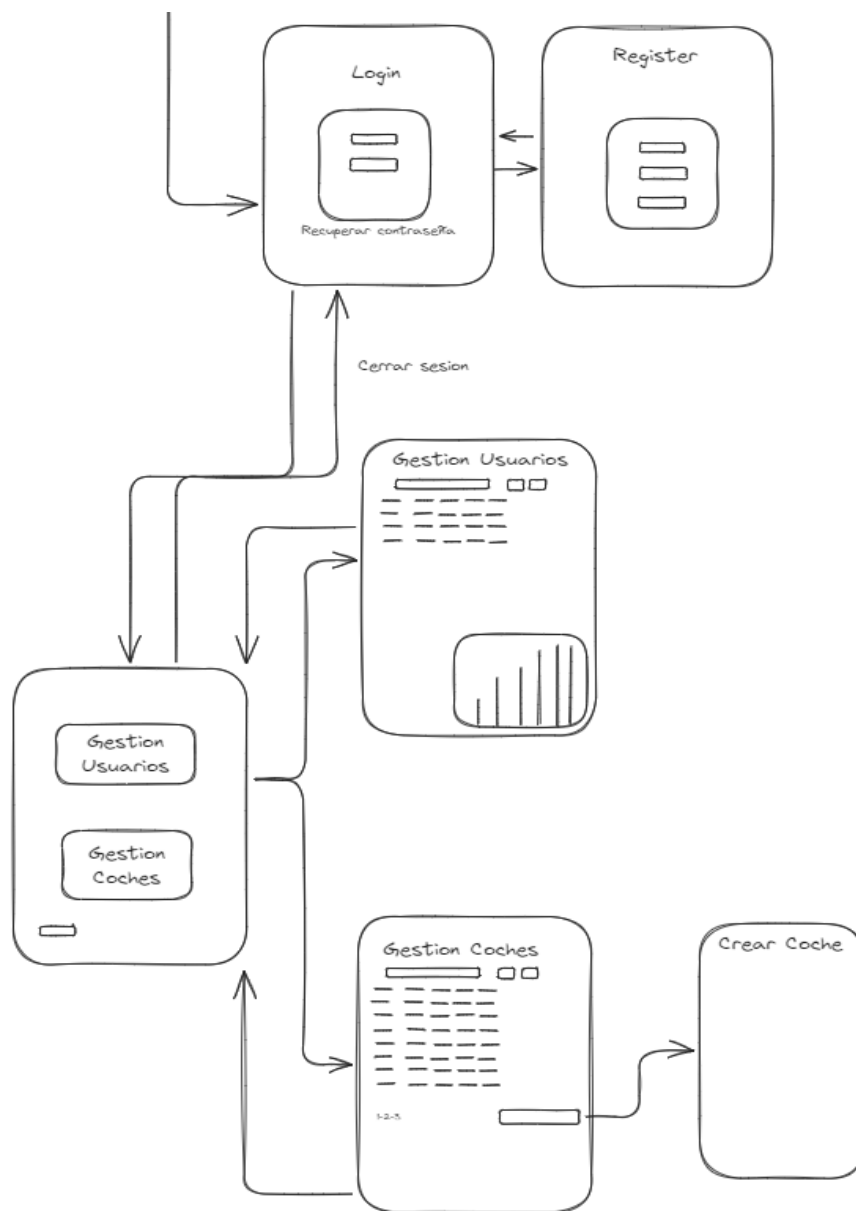


Este es el diseño final del diagrama de flujo, ofreciendo numerosos beneficios significativos:

- Hace que sea más fácil para los usuarios encontrar información, el camino desde "Inicio" a secciones como "perfil" o "ajustes" son claras, lo que reduce las molestias y evita que la gente abandone la página.
- La planificación previa: Permite identificar interacciones innecesarias o secciones necesarias antes de la implementación. Esto ahorra tiempo y recursos durante el desarrollo web.
- Un buen diseño ayuda a mantener la apariencia y el funcionamiento del sitio web iguales en cada página.
- Si en el futuro quieres añadir más cosas, este mapa te muestra si todo puede encajar sin estropear el funcionamiento actual, asegurando su crecimiento.

En conclusión, mejora la experiencia del usuario a la vez que optimiza la gestión y el mantenimiento del proyecto, sincronizando las funciones con los requisitos del usuario y los objetivos del proyecto.

Sección de Administrador

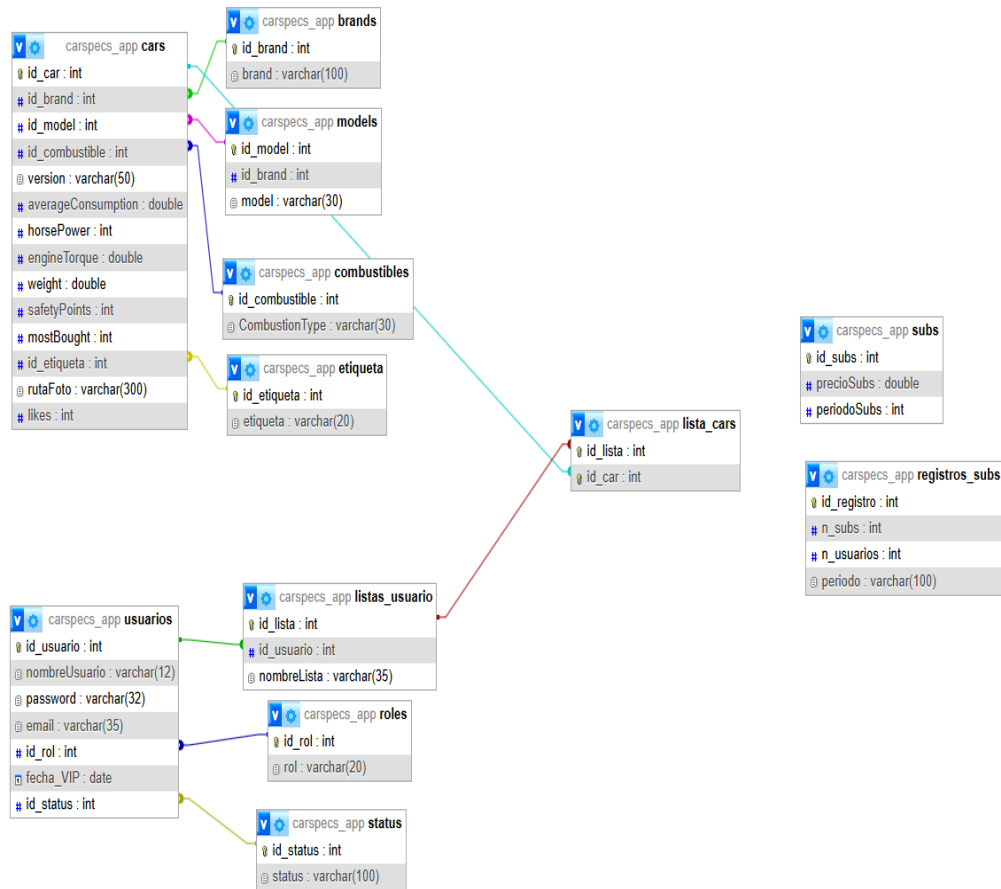


Este diseño de flujo para la sección de administración tiene varias ventajas clave:

- Ayuda al administrador a acceder a tareas importantes, como gestionar personal y vehículos, mediante un proceso claro y sencillo. Esto mejora la velocidad y la precisión en las operaciones administrativas.

- Separa las funciones, como "Atención al Usuario" y "Mantenimiento de Vehículos", para simplificar el proceso, ayudando a cada parte a manejar solo una tarea. Líneas claras entre tareas, para que un administrador pueda hacer múltiples tareas sin tener que volver siempre a la pantalla principal. Esto optimiza el tiempo y mejora la usabilidad. Los bloques modulares nos permiten agregar nuevas secciones fácilmente más adelante, como formas adicionales de gestionar cosas, sin tener que cambiar todo el sistema antiguo.
- El sistema debe tener un nombre de usuario y una contraseña de administrador claros para controlar quién puede entrar y proteger las funciones importantes de usuarios no deseados y así mantener la seguridad del sistema.
- Este diseño optimiza las tareas de gestión a la vez que fomenta intercambios flexibles y seguros, permitiendo así un mantenimiento y una progresión constantes de la plataforma

Estructura de la base de datos



Las tablas están organizadas por categorías lógicas donde se separa claramente los coches y las tablas relacionadas con sus atributos indexados como pueden ser brands, model, combustible, etiqueta, por otra parte, los usuarios junto a los roles y estatus. Para relacionar ambas categorías se usan dos tablas, una en la que se asigna cada lista al usuario que pertenece y por otra donde se asigna cada coche con cada lista en la que esta agregada, de esta forma cuando un usuario quiere acceder a sus listas, primero accede a su lista con id x y esa lista contiene una serie de coches asignados. Por ultimo están las tablas que no tienen relación con el entorno de uso normal , sino que cada una tiene su propia categoría más

orientada a ser funcional, por ejemplo la tabla subs sirve para almacenar los tipos de suscripción y toda información al respecto, de forma que se pueden agregar planes de suscripción o eliminar sin afectar a las demás y que se actualice automáticamente en la vista, el registro_subs, en cambio, lleva un registro que más tarde se usara para de forma dinámica actualizar la gráfica de la sección de gestión de usuarios donde se puede visualizar el progreso de la web en cuanto a usuarios totales y su subdivisión en usuarios VIP o NO VIP.

TECNOLOGÍAS

A continuación, se detallan las diferentes tecnologías que han sido utilizadas durante la ejecución del proyecto:

	Usado en toda la web para procesar información, gestionar la interacción, realizar operaciones con BBDD.
	Estructuración MVC y uso de herramientas como son las sesiones, conexiones a BBDD, paginación etc...
	Gestión de BBDD
	Recuperación de contraseña mediante un mail.



Sistema de Pago.



Uso en gráficas en las secciones de administración de los usuarios

The logo for JavaScript, featuring the letters "JS" in a bold, black, sans-serif font on a yellow background.

Gestión de actualizaciones dinámicas, formularios y demás interacciones frontend.



Envío de peticiones y actualización dinámica de información



Librería de Python para navegar en la web, usado para sacar información e introducir datos en inputs

	<p>Librería de Python para enviar peticiones directamente sin necesidad de usar la interfaz de una web, usado para introducir información en la base de datos</p>
	<p>librería de python usada para conectar python con la base de datos</p>
	<p>Scripts de automatización de procesos.</p>

METODOLOGÍA

Para mi proyecto, determiné usar un enfoque rápido y flexible utilizado para ello metodología ágiles. Los sprint han tenido una periodicidad semanal, comenzando en cada uno de ellos con la planificación de tarea para dicho sprint y terminaba con una reunión de revisión. Durante las reuniones regulares, aseguramos un diálogo continuo entre el grupo, facilitando el seguimiento de los avances y la identificación rápida de impedimentos.

Después de cada sprint, teníamos una reunión para mostrar el trabajo realizado esa semana. Después, los miembros del equipo discutían lo que habían logrado y hablaban sobre maneras de mejorar. Por último, se establecen los objetivos para la siguiente fase del proyecto, asegurándonos de que todos conocieran sus responsabilidades y las tareas importantes en las que debían trabajar.

Implementar esta metodología nos permite mantener la maleabilidad necesaria para la modificación en respuesta a las evoluciones, junto con la mejora continua de la calidad del resultado final.

Configurar el proyecto con ideas clave simplifica las cosas y funciona bien con las metodologías ágiles. "Este marco permite que el desarrollo se segmente en partes manejables y visualmente enfocadas, destacando las funciones clave del sistema, facilitando el despliegue." Reemplacé 'estructura' por 'marco', 'permitir' por 'permite', 'dividido' por 'segmentado', 'visual' por 'visualmente enfocadas', 'funciones'

Organizar el proyecto en torno a las perspectivas principales busca segmentar las tareas en unidades coherentes, donde cada perspectiva encarna una tarea o aspecto distinto del proyecto. Simplificando la planificación, la asignación de trabajo y la realización de estas tareas, ya que cada pieza se construye, pero permanece como parte del mismo proyecto general. Este método también permite al grupo identificar sin

esfuerzo qué sección del proyecto se está desarrollando en cada momento, y amplifica la capacidad de evaluar progresivamente.

Además, concentrarse en las perspectivas centrales ayuda a crear una imagen más clara del compromiso del consumidor y su interacción general con la interfaz. Cada vista tiene un papel que apoya el objetivo principal del proyecto, ayudando a que el desarrollo sea más fácil y amigable para el cliente.

Algunos beneficios que puedo nombrar de esta práctica son:

- Cada parte es como una pieza separada que se puede hacer, comprobar y ajustar por separado antes de integrarse en el proyecto grande. De esta manera, podemos trabajar más fácilmente y controlar cómo van las cosas.
- Clarificar las partes importantes, para que el diseño funcione bien para todos. De esta manera, cada parte ayuda a la persona que lo usa.
- Este método ayuda a realizar entregas rápidas en cada parte, ya que el progreso se planifica por partes, es más sencillo comprobar el progreso continuo y solucionar problemas según sea necesario, lo que proporciona un flujo más fluido de mejoras.
- Mantener las partes de un sistema juntas una por una reduce las posibilidades de **problemas y ayuda a detectar problemas desde el principio.**

IMPLEMENTACIONES

Este es el registro final de todas las implementaciones del proyecto:

Login

- **30/09:** Inicio del desarrollo del login.
- **03/10:** Finalización del desarrollo del registro de usuarios (sign-up).
- **04/10:** Finalización de la funcionalidad del login.

Tiempo total: 24 horas (Del 30/09 al 04/10).

Buscador

- **21/10:** Inicio del desarrollo del buscador.
- **22/10:** Buscador funcional. Se implementaron consultas vacías y se propusieron ideas para filtros.
- **23/10:** Se configuró el buscador como menú principal. Estados logueados y restricciones implementadas.
- **28/10:** Filtros completados.
- **29/10:** Implementación de filtros dinámicos (etiqueta y combustible). Corrección de errores en la funcionalidad de agregar coches.

Tiempo total: 40 horas

Menú perfil

- **07/10:** Inicio de desarrollo del menú (controladores y modelos).
- **08/10:** Avances en la funcionalidad del menú.
- **09/10 - 10/10:** Implementación de la funcionalidad de visualización de listas. Limpieza de código y avance en el maquetado.
- **14/10:** Mejora del maquetado del menú e inicio de la funcionalidad de eliminar.

- **15/10 - 17/10:** Desarrollo de modales, implementación de eliminación de elementos y revisión de la vista del menú.
- **18/10:** Remake del maquetado.
- **23/10 - 24/10:** Funcionalidades para agregar listas y coches, y eliminar listas completas.

Tiempo total: 88 horas (Del 07/10 al 24/10).

Pasarela de pago

- **30/10:** Inicio del desarrollo de la vista para la pasarela de pago.
- **31/10:** Revisión de documentación técnica de la pasarela.
- **04/11:** Resolución de problemas técnicos para la transferencia de datos.
- **05/11:** Desarrollo finalizado. La suscripción quedó completamente funcional.

Tiempo total: 24 horas (Del 30/10 al 05/11).

Sección de gestión de coches

- **08/11:** Inicio del desarrollo.
- **11/11:** Implementación de funcionalidades: buscador, modificación de coches, creación de modelos y marcas.
- **12/11:** Desarrollo de la funcionalidad para crear y eliminar coches (sin gestión de imágenes).
- **13/11:** Integración de API para alimentar la base de datos.
Desarrollo del mailer para restablecer contraseñas.
- **15/11:** Resolución de problemas con el mailer y finalización de la integración con la API.

Tiempo total: 40 horas (Del 08/11 al 15/11).

Sección de gestión de usuarios

- **06/11:** Inicio de la funcionalidad para buscar usuarios, cambiar estado y asignar privilegios VIP.
- **07/11:** Implementación de la paginación y avance en gráficos.
- **08/11:** Finalización de gráficos e inicio de funcionalidades adicionales.
- **18/11:** Desarrollo de validaciones y restricciones para usuarios inactivos.
- **28/11:** Finalización de funcionalidades: mostrar, editar, eliminar y crear usuarios. Corrección de errores en búsquedas y permisos.

Tiempo total: 64 horas (Del 06/11 al 28/11).

Tiempo total proyecto completo: 280 horas

DESPLIEGUE/ INSTALACIÓN

Para R el proyecto, instale primero las herramientas y servicios necesarios para los entornos de desarrollo y producción. A continuación, detallo los pasos y elementos imprescindibles.

Instalación de WAMP

WAMP (Windows, Apache, MySQL y PHP) es un paquete que proporciona un entorno de desarrollo completo y funcional en sistemas operativos Windows. Este paquete incluye:

- **Apache:** Servidor web necesario para alojar y servir las vistas del proyecto.
- **MySQL:** Sistema de gestión de bases de datos utilizado para almacenar toda la información del proyecto, como datos de usuarios, vehículos, listas personalizadas, entre otros.
- **PHP:** Lenguaje de programación utilizado en el backend para la lógica del proyecto.

Para instalar WAMP:

1. Descarga el instalador desde el sitio oficial de WAMP.
2. Sigue los pasos del instalador seleccionando los componentes deseados (en este caso, asegúrate de incluir Apache, MySQL y PHP).
3. Configura la instalación asignando los puertos y directorios necesarios para el correcto funcionamiento del servidor.

Configuración de Apache

El servidor Apache será el encargado de servir las vistas del proyecto. Es importante verificar que esté configurado correctamente:

1. Asegúrate de que el puerto 80 (o el puerto configurado) esté libre para evitar conflictos con otras aplicaciones.
2. Añade los archivos del proyecto al directorio raíz del servidor (habitualmente, [www](#)).
3. En el archivo de configuración de Apache ([httpd.conf](#)), verifica que el módulo PHP esté habilitado y que los permisos de acceso sean correctos.

Configuración de MySQL

El proyecto utiliza una base de datos MySQL para gestionar toda la información. Para configurarlo:

1. Accede a **phpMyAdmin**, incluido en WAMP, para gestionar la base de datos.
2. Importa el archivo [.sql](#) que contiene las estructuras y datos iniciales necesarios para el proyecto.

Instalación de PHP

PHP es el lenguaje en el que está desarrollado el backend del proyecto. Asegúrate de que la versión instalada de PHP sea compatible con el proyecto.

Descompresión del archivo del proyecto

El proyecto está contenido en un archivo comprimido que incluye todo el código fuente necesario:

1. Descomprime el archivo en el directorio raíz del servidor ([www](#)).
2. Verifica que todos los archivos estén correctamente organizados y que no haya errores de permisos.

Configuración del proyecto

1. Edita los archivos de configuración de la base de datos (`application\config\database.php`) para que coincidan con los valores de tu entorno local, como:
 - Hostname
 - Username
 - Password
 - Database
 - Dbdriver

2. Reinicia los servicios de WAMP para aplicar todos los cambios.

Con todo esto, si se accede a `localhost/carspecs_app` debería aparecer el buscador como página de inicio.

TRABAJOS FUTUROS

La iniciativa se ha concebido con un esquema adaptable que posibilita ajustarse con facilidad a distintas modificaciones y evoluciones, tanto en la gestión como en la experiencia del usuario. Esta **flexibilidad para adaptarse** destaca como uno de los factores cruciales que favorecen la expansión y progresión del proyecto conforme se transforman las exigencias comerciales o las demandas de los usuarios.

Cambios en la gestión:

La flexibilidad se ha proyectado en la adaptabilidad de la gestión de información y procedimientos internos. En un principio, se pensó en emplear hojas de cálculo para facilitar el intercambio de datos en gran cantidad. Esta elección brindaría a los usuarios la capacidad de manejar extensas cantidades de datos de forma veloz y simple. También haría más fácil la conexión con otros sistemas y la importación de información desde fuentes externas.

Igualmente, se consideró la posibilidad de incorporar APIs de terceros para enriquecer la base de datos, lo que resultaría beneficioso para acceder a detalles actualizados y exactos acerca de automóviles, marcas, valores, entre otros. Esta característica fue añadida previamente, pero no funcionó como se esperaba, por lo tanto, se optó por descartarla de la propuesta. A pesar de esto, la estructura del proyecto aún es adecuada para integrarla de nuevo si fuese requerido en un futuro.

Cambios para el usuario:

En cuanto a las necesidades de los usuarios, el proyecto también se ha diseñado con un alto grado de adaptabilidad. La plataforma está pensada para incorporar nuevas funcionalidades que puedan mejorar la experiencia

del usuario, tales como enlaces a portales de compras, comparaciones de vehículos, entre otras.

- **Enlaces a portales de compras:** Se puede añadir fácilmente la funcionalidad que permita a los usuarios acceder a portales de compra de vehículos o a tiendas en línea, facilitando la compra o el acceso a más opciones. Esta integración podría estar relacionada con las preferencias de búsqueda de vehículos o con la comparación de precios y ofertas disponibles.
- **Comparaciones de vehículos:** Una posible mejora sería la implementación de una herramienta que permita a los usuarios comparar diferentes vehículos según criterios específicos, como el precio, las características técnicas, las valoraciones de otros usuarios, etc. Esto enriquecería la experiencia de navegación del usuario, permitiéndole tomar decisiones más informadas y personalizadas.

Flexibilidad en el futuro:

La estructura modular y escalable del proyecto asegura que se puedan agregar nuevas características o modificar las existentes sin afectar significativamente el rendimiento general del sistema. Además, se puede ajustar a futuras necesidades o tendencias del mercado, lo que permite que el proyecto se mantenga actualizado y competitivo a lo largo del tiempo.

En resumen, la capacidad del proyecto para adaptarse a diversos cambios es fundamental para su éxito a largo plazo. Aunque algunas funcionalidades inicialmente propuestas no se implementaron debido a limitaciones técnicas o de resultados, el diseño subyacente sigue permitiendo que nuevas mejoras sean integradas de manera ágil y eficiente, tanto para la gestión interna como para mejorar la experiencia del usuario final. Esto proporciona una base sólida para el crecimiento del

proyecto, permitiendo su evolución conforme cambian las necesidades tecnológicas o las expectativas de los usuarios.

CONCLUSIONES

En cuanto a mis pensamientos finales sobre este proyecto, puedo afirmar con satisfacción que he logrado cumplir con los objetivos que me propuse desde el inicio: hacer que la información sobre los vehículos del mercado sea más accesible e interactiva para los usuarios. A través del desarrollo de esta plataforma, he conseguido transformar la experiencia de navegación sobre vehículos, alejándome de las referencias tradicionales que a menudo no ofrecían interacción alguna con el usuario y contaban con interfaces caóticas que dificultan su uso.

Accesibilidad y Simplicidad:

El objetivo principal ha sido mejorar la accesibilidad de la información. Al hacerlo, me aseguré de que los usuarios pudieran encontrar lo que buscan de manera rápida, clara y eficiente, sin verse abrumados por una interfaz desordenada o complicada. Opté por diseñar un sistema que sea intuitivo y fácil de usar, donde los usuarios puedan navegar entre las diferentes categorías de vehículos, filtrar opciones según sus preferencias y acceder a la información de forma estructurada.

Interactividad:

Un aspecto esencial de este proyecto ha sido incrementar la interactividad de la plataforma. A diferencia de las soluciones anteriores, donde la información era estática y sin posibilidades de interacción, he introducido diversas funcionalidades que permiten al usuario participar activamente. Estas funciones incluyen la capacidad de guardar vehículos favoritos, crear y gestionar listas personalizadas, y realizar búsquedas avanzadas. Además, el sistema ha sido diseñado para permitir una experiencia dinámica, con filtros y opciones que facilitan la personalización de la búsqueda, lo que ayuda a los usuarios a tomar decisiones más informadas y adaptadas a sus necesidades.

Uso de Nuevas Herramientas y Métodos:

Durante el desarrollo, utilicé una serie de herramientas nuevas que no había implementado previamente en otros proyectos. Esto me permitió ampliar mis conocimientos y experimentar con tecnologías y técnicas innovadoras que no solo mejoraron la funcionalidad del proyecto, sino que también optimizan su rendimiento y usabilidad. Además, implementé metodologías ágiles, que me proporcionaron la flexibilidad para adaptarme a cambios rápidos, coordinar eficazmente con el equipo y entregar el producto final en tiempo y forma.

Por otro lado, también utilicé métodos no experimentados anteriormente que me permitieron abordar problemas de manera diferente. La integración de APIs externas, la creación de vistas interactivas basadas en el comportamiento del usuario y la incorporación de herramientas como el scraping de datos fueron técnicas nuevas que enriquecieron significativamente el proyecto.

REFERENCIAS

AUTODATA

Aplicado en la investigación del tema de la web y a la hora de sacar información de los coches mediante automatizaciones.

<https://www.auto-data.net/> Technical specs, data, fuel consumption of cars

KM77

Aplicado en la investigación del tema de la web.

<https://www.km77.com> Revista de coches, novedades y pruebas de coches. Reportajes, noticias y artículos técnicos.

UltimateSpecs

Aplicado en la investigación del tema de la web.

<https://www.ultimatespecs.com/es> Ficha técnica de coches, dimensiones y consumo de combustible

Documentación PHP

Se utilizó como referencia para comprender las funciones y métodos de PHP, base del desarrollo backend del proyecto.

PHP.net. (s. f.). *PHP Manual*. Recuperado de <https://www.php.net/manual>

Documentación CodeIgniter 3

Framework principal para estructurar el desarrollo del proyecto, ofreciendo un entorno ágil y organizado.

CodeIgniter Foundation. (s. f.). *CodeIgniter 3 User Guide*. Recuperado el de https://codeigniter.com/user_guide

Redsys

Proporcionó la integración de la pasarela de pago necesaria para gestionar las suscripciones VIP.

Redsys. (s. f.). *Documentación técnica de la pasarela de pago Redsys*. Recuperado de <https://www.redsys.es>

Highcharts

Herramienta utilizada para generar gráficos interactivos del avance del número de usuarios.

Highcharts. (s. f.). *Highcharts Documentation*. Recuperado de <https://www.highcharts.com/docs>

WAMPServer

Se usó para crear el entorno local de desarrollo, integrando Apache, MySQL y PHP de manera eficiente.

WAMPServer. (s. f.). *WAMP Documentación*. Recuperado de <https://www.wampserver.com>

PHPMailer

Biblioteca usada para hacer la recuperación de contraseña mediante un mail.

PHPMailer. (s. f.). *PHPMailer Documentación*. Recuperado de <https://github.com/PHPMailer/PHPMailer>

MySql

Sistema de gestión de bases de datos empleado para almacenar y gestionar la información del proyecto.

Oracle Corporation. (s. f.). *MySQL Documentación*. Recuperado de <https://dev.mysql.com/doc>

Selenium

Herramienta usada para sacar la información y almacenarla en la base de datos

Selenium Foundation. (s. f.). *Selenium Documentación*. Recuperado de <https://www.selenium.dev/documentation>