PROYECTO DE FIN DE GRADO

BRULECAR

**TRABAJO REALIZADO POR:**

**ADRIÁN PÉREZ CASTAÑO**

INTRODUCCIÓN

A continuación se describirá toda la información y el proceso realizado en el proyecto final de desarrollo de aplicaciones web. Dicho sitio web consiste en compra y venta de coches.

Este sitio web está pensado para usarse desde cualquier tipo de navegador y conexión además del sistema operativo Windows. Sin estar en desarrollo todavía para dispositivos móviles y Linux.

La página esta pensada para que solo los usuarios registrados puedan ver, comprar o vender los coches, ya que así se les da a los usuarios registrados la oportunidad de que solamente puedan ver sus coches en venta ya que así le damos a dichos usuarios la protección de datos de los vehículos.

El registro es totalmente gratis, pero está dedicada para personas que superen la mayoría de edad.

El proyecto está realizado por Adrián Pérez Castaño, esta idea fue pensada hace tiempo ya que estaba en un punto donde quería comprar un coche y me dio la idea para hacer dicho proyecto.

PLANIFICACIÓN PREVIA

La planificación a realizar fue bastante sencilla ya que al ser de un solo integrante, yo puse las pautas a seguir para dividir así mejor el trabajo.

Empecé haciendo la base de datos después de una idea que tuve acerca de como lo quería hacer, a continuación empecé a hace la página principal para así tener una mejor idea y visualización de como lo quería hacer y de los colores a utilizar, después de ello hice el header y el footer que van como conjunto en todas las páginas del proyecto.

Poco después realicé el login, logout y register para los usuarios en una página dedicada a ello.

Al realizar lo anterior comencé a hacer las distintas páginas que hay, como son las de contacto, compra y venta.

ANÁLISIS

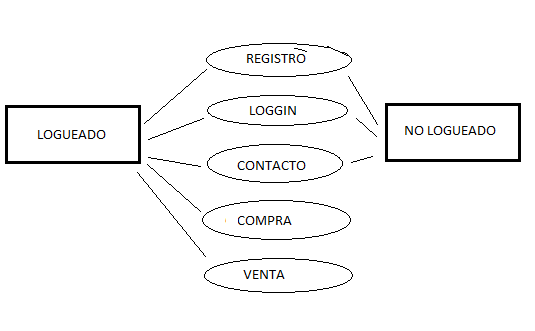
Tras la planificación del proyecto, a continuación se explicarán los requisitos y casos de uso dando así una vista previa de como serán los roles de los distintos usuarios.

REQUISITOS GENERALES

1. Acceso a internet
2. Tipo de explorador a elegir.
3. Sistema operativo Windows 10.
4. El dispositivo donde se vaya a proyectar la página deberá tener:

* Procesador: Procesador a 1 GHz o más rápido o sistema en un chip (SoC)
* RAM: 1 GB para 32 bits o 2 GB para 64 bits.
* Disco duro: 16 GB para un SO de 32 bits o 32 GB para un SO de 64 bits.

CASOS DE USO



* Tras entrar a la página, siendo un usuario no registrado, aparecerá la página home, donde se explica brevemente la página web, también aparecerá la página contacto, donde se mostrará un formulario para cualquier duda que se enviará por correo y por último pero más importante la página de loggin y register, donde el usuario al no estar registrado, se registrará en un formulario facilitando distintos datos.
* El usuario al registrarse volverá a la página de inicio sesión, donde pondrá sus credenciales y accederá a su perfil. Dicha página se transforma en su página de perfil donde aparecerán sus datos, además de estos, abajo aparecerán los coches que tenga en venta ese usuario.
* Tras ver el perfil del usuario, aparecerán dos nuevas páginas, que son las de venta, donde el usuario tras un formulario podrá poner en venta su vehículo y la de compra, donde aparecerán todos los vehículos que han puesto todos los usuarios en venta, también podrán filtrar los coches para una mejor accesibilidad a ellos.
* Al comprar un coche, este desaparecerá de la página de compra y se pondrá automáticamente en la lista de ventas de la página de perfil del usuario.

DISEÑO

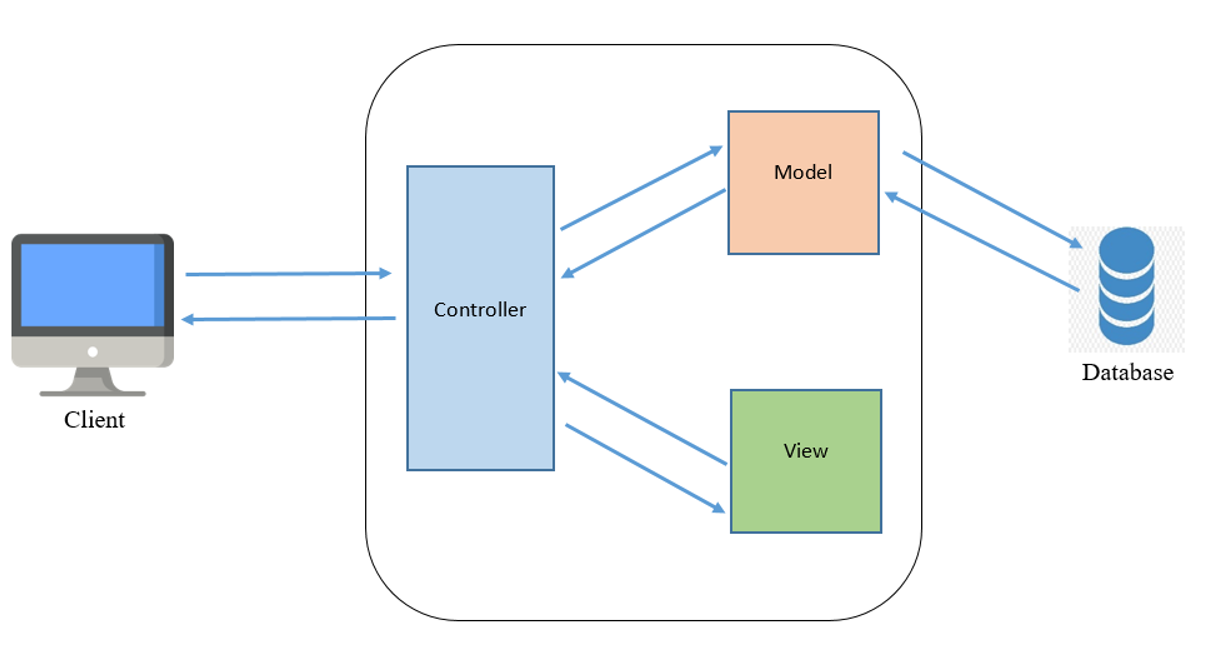
A continuación se detallará el diseño de aplicación que se ha llevado a cabo para este proyecto, tanto como la arquitectura de la aplicación, modelo de datos, interfaz gráfica, etc.

ARQUITECTURA

En este proyecto se ha utilizado el patrón de arquitectura de Modelo-Vista-Controlador (MVC), dividen la lógica del programa relacionado en tres elementos interconectados. Esto se hace para separar las representaciones internas de la información de las formas en que se presenta y acepta la información del usuario.

A continuación se explicaran los 3 componentes de esta arquitectura :

* **Modelo**: El componente central del patrón. Es la estructura de datos dinámica de la aplicación, independiente de la interfaz de usuario. Gestiona directamente los datos, la lógica y las reglas de la aplicación; se encarga de gestionar los datos de la aplicación. Recibe la entrada del usuario del controlador.
* **Vista**: Cualquier representación de información como un gráfico, diagrama, tabla, etc. Son posibles múltiples vistas de la misma información; representa la presentación del modelo en un formato particular.
* **Controlador**: Acepta la entrada y la convierte en comandos para el modelo o la vista; responde a la entrada del usuario y realiza interacciones en los objetos del modelo de datos. El controlador recibe la entrada, opcionalmente la valida y luego pasa la entrada al modelo.



CAPA DE PRESENTACIÓN

Esta capa se encarga de desarrollar e implementar toda la lógica de negocio de la nueva aplicación. Proporciona servicios web a la capa de presentación u a otros posibles sistemas externos que lo necesiten.

Las tecnologías utilizadas han sido HTML, CSS y un framework llamado Boostrap para dar calidad a la página.

* **HTML**: HyperText Markup Language (HTML), es un lenguaje estándar amplio utilizado para la elaboración de páginas web y altamente soportado por los navegadores. Es el elemento de construcción más básico para la creación y presentación visual del contenido de una página web.
* **CSS**: CSS es el acrónimo de Cascading Style Sheets (Hoja de estilos en cascada). Es un elemento básico complementario al HTML cuya funcionalidad es establecer la apariencia de la página web. Junto con HTML, es una tecnología muy popular para crear páginas web visualmente atractivas. Nace con la finalidad de separar el contenido de la forma de presentarse.
* Framework Boostrap: Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto) para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en [HTML](https://es.wikipedia.org/wiki/HTML) y [CSS](https://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada), así como extensiones de [JavaScript](https://es.wikipedia.org/wiki/JavaScript) adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo [front-end](https://es.wikipedia.org/wiki/Front-end). Bootstrap es el segundo proyecto más destacado en [GitHub](https://es.wikipedia.org/wiki/GitHub)[1](https://es.wikipedia.org/wiki/Bootstrap_(framework)#cite_note-1)​ y es usado por la [NASA](https://es.wikipedia.org/wiki/NASA) y la [MSNBC](https://es.wikipedia.org/wiki/MSNBC), entre otras organizaciones.

Detallaré esta parte con una imagen del proyecto:

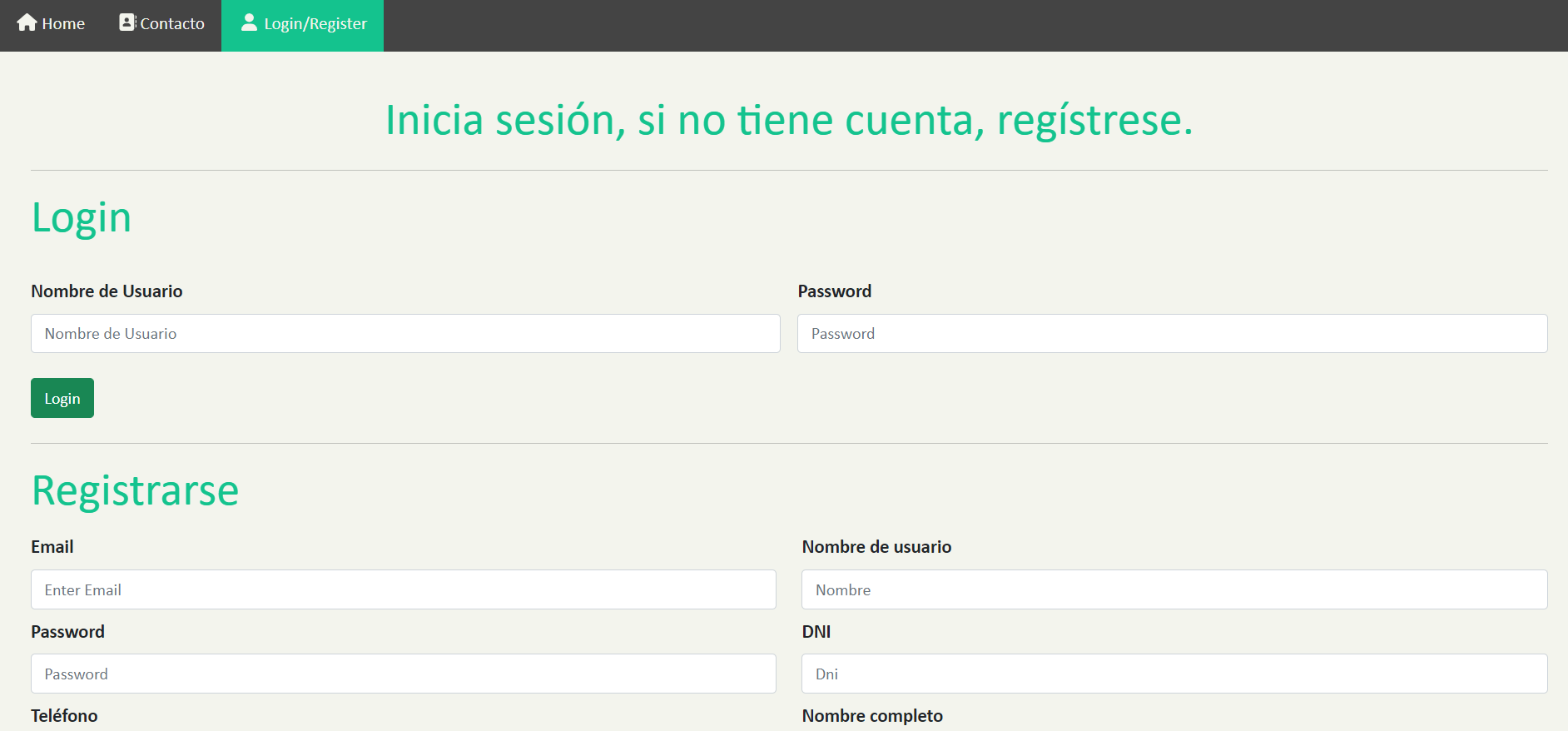


CAPA DE DOMINIO O APLICACIÓN

Esta capa se encarga de desarrollar e implementar toda la lógica de negocio de la nueva aplicación. Proporciona servicios web a la capa de presentación u a otros posibles sistemas externos que lo necesiten.

Las tecnologías que se han utilizado son las siguientes:

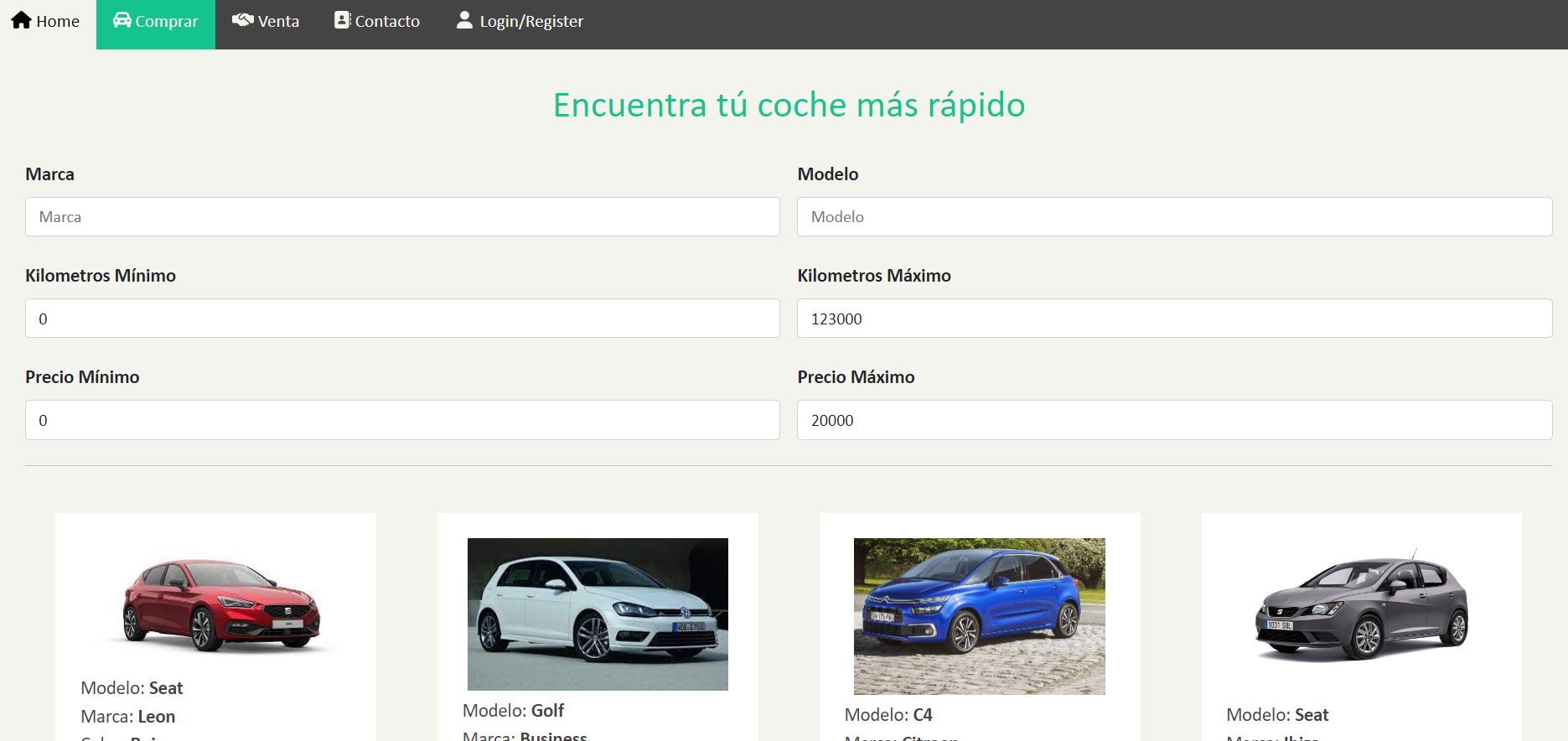
* **PHP**: PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. El código PHP suele ser procesado en un [servidor web](https://es.wikipedia.org/wiki/Servidor_web) por un [intérprete](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)) PHP implementado como un  o como un ejecutable de [interfaz2 de entrada común](https://es.wikipedia.org/wiki/Interfaz_de_entrada_com%C3%BAn) (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado —que puede ser cualquier tipo de datos, como el [HTML](https://es.wikipedia.org/wiki/HTML) generado o datos de imágenes binarias— formaría la totalidad o parte de una respuesta [HTTP](https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto).



**JavaScript**: JavaScript (abreviado comúnmente JS) es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [interpretado](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)), dialecto del estándar [ECMAScript](https://es.wikipedia.org/wiki/ECMAScript). Se define como [orientado a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos),[2](https://es.wikipedia.org/wiki/JavaScript#cite_note-2)​ [basado en prototipos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_basada_en_prototipos), imperativo, [débilmente tipado](https://es.wikipedia.org/wiki/Tipado_fuerte#Lenguajes_no_tipados) y [dinámico](https://es.wikipedia.org/wiki/Tipado_din%C3%A1mico).

Se utiliza principalmente del [lado del cliente](https://es.wikipedia.org/wiki/Cliente_(inform%C3%A1tica)), implementado como parte de un [navegador web](https://es.wikipedia.org/wiki/Navegador_web) permitiendo mejoras en la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) y [páginas web](https://es.wikipedia.org/wiki/P%C3%A1gina_web) dinámicas[3](https://es.wikipedia.org/wiki/JavaScript#cite_note-3)​ y JavaScript del [lado del servidor](https://es.wikipedia.org/wiki/Script_del_lado_del_servidor) (Server-side JavaScript o SSJS). Su uso en [aplicaciones](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica) externas a la [web](https://es.wikipedia.org/wiki/World_Wide_Web), por ejemplo en documentos [PDF](https://es.wikipedia.org/wiki/PDF), aplicaciones de escritorio (mayoritariamente [widgets](https://es.wikipedia.org/wiki/Widget)) es también significativo.

En este proyecto se ha utilizado este lenguaje en casi todos los apartados.



Para el desarrollo y la aplicación de este proyecto ha sido indispensable la utilización de un servidor web, como lo es **Apache.**

El servidor web tiene la función de permanecer a la espera de peticiones HTTP de los clientes, que suelen llevarlas a cabo a través de un navegador web. Los clientes realizan peticiones HTTP al servidor, y este les responde con el/los contenido/s que los clientes solicitan. Los contenidos devueltos por el servidor suelen ser interpretados por los navegadores web de los clientes, que los muestran por pantalla.

CAPA DE DATOS

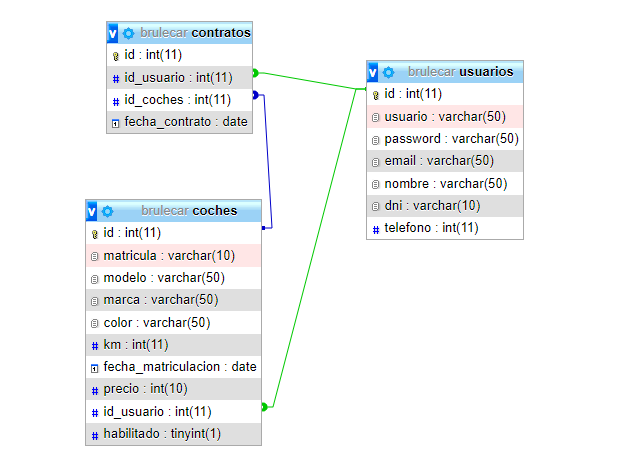
Esta capa se nutre del sistema gestor de base de datos (SGBD) a partir de peticiones de la capa de dominio. La organización consiste en el diseño de la base de datos donde se almacenan los datos de nuestra aplicación.

Para este proyecto se ha utilizado la base de datos de PHPMyAdmin.

**PHPMyAdmin**: **es una aplicación web** que sirve para**administrar bases de datos MySQL** de forma sencilla y con una interfaz amistosa. Se trata de un software muy popular basado en PHP. La ventaja de usar una aplicación web es que nos permite conectarnos con servidores remotos, a los cuales no siempre se puede acceder usando programas de interfaz gráfica.

Para usar phpMyAdmin simplemente necesitas subir el conjunto de archivos PHP que componen la aplicación a un servidor web, configurar con los datos de acceso a MySQL y empezar a administrar las bases de datos. Con phpMyAdmin puedes hacer todo tipo de operaciones, desde la creación y borrado de bases de datos a la administración de las tablas.

El modelo de Entidad -> Relación es el siguiente:



* Datos para la creación y cambios en la base de datos:

CREATE TABLE `coches` (

`id` int(11) NOT NULL,

`matricula` varchar(10) NOT NULL,

`modelo` varchar(50) NOT NULL,

`marca` varchar(50) NOT NULL,

`color` varchar(45) NOT NULL,

`km` int(11) NOT NULL,

`fecha\_matriculacion` date NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `contratos`

--

CREATE TABLE `contratos` (

`id` int(11) NOT NULL,

`id\_usuario` int(11) NOT NULL,

`id\_coche` int(11) NOT NULL,

`fecha` date NOT NULL,

`tipo` varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `direcciones`

--

CREATE TABLE `direcciones` (

`id` int(11) NOT NULL,

`id\_usuario` int(11) NOT NULL,

`direccion` varchar(150) NOT NULL,

`municipio` varchar(50) NOT NULL,

`provincia` varchar(50) NOT NULL,

`cp` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `usuarios`

--

CREATE TABLE `usuarios` (

`id` int(11) NOT NULL,

`usuario` varchar(20) NOT NULL,

`password` varchar(50) NOT NULL,

`email` varchar(50) NOT NULL,

`nombre` varchar(50) NOT NULL,

`dni` varchar(10) NOT NULL,

`usuarioscol` varchar(45) NOT NULL,

`telefono` int(11) NOT NULL,

`tipo` varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Índices para tablas volcadas

--

--

-- Indices de la tabla `coches`

--

ALTER TABLE `coches`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `contratos`

--

ALTER TABLE `contratos`

ADD PRIMARY KEY (`id`),

ADD KEY `fk\_usuario` (`id\_usuario`),

ADD KEY `fk\_coche` (`id\_coche`);

--

-- Indices de la tabla `direcciones`

--

ALTER TABLE `direcciones`

ADD PRIMARY KEY (`id`),

ADD KEY `fk\_idusuario` (`id\_usuario`);

--

-- Indices de la tabla `usuarios`

--

ALTER TABLE `usuarios`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT de las tablas volcadas

--

--

-- AUTO\_INCREMENT de la tabla `usuarios`

--

ALTER TABLE `usuarios`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

--

-- Restricciones para tablas volcadas

--

--

-- Filtros para la tabla `contratos`

--

ALTER TABLE `contratos`

ADD CONSTRAINT `fk\_coche` FOREIGN KEY (`id\_coche`) REFERENCES `coches` (`id`),

ADD CONSTRAINT `fk\_usuario` FOREIGN KEY (`id\_usuario`) REFERENCES `usuarios` (`id`);

--

-- Filtros para la tabla `direcciones`

--

ALTER TABLE `direcciones`

ADD CONSTRAINT `fk\_idusuario` FOREIGN KEY (`id\_usuario`) REFERENCES `usuarios` (`id`);

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

INTERFAZ GRÁFICA

En el desarrollo de nuestra interfaz gráfica se han tenido en cuenta varios factores.

La interfaz es sencilla y se comprende fácilmente para que cualquier usuario pueda usarla con fluidez.

Todo esta al alcance de la vista, por lo que toda la información se encuentra muy fácil.

En cuanto al menú con iconos y con texto hace que navegar por la página sea extremadamente fácil, ya que ha simple vista se ve y se comprende.

Los colores elegidos son escogidos minuciosamente para que el usuario no tenga problemas a reconocer los datos y se detalle con colores los datos o la información más relevante.

USABILIDAD

En cuanto a la usabilidad, se ha intentado crear una aplicación lo más sencilla posible con la que el usuario pueda cumplir con una tarea específica de manera efectiva y eficaz, el usuario puede acceder con facilidad a la aplicación y puede moverse a través de ella sin importar hardware o software.

El sistema es bastante útil e intuitivo para que el usuario pueda navegar sin ningún tipo de problema.

Respecto a la fiabilidad de la aplicación, el sistema cumple con casi todos los requisitos para el uso por parte del usuario.

Se han aplicado lenguajes que no dan problemas para la carga de las páginas y se han probado para que no haya ningún problema en la carga de estos lenguajes.

PRUEBAS

Al terminar cada apartado de la aplicación se han realizado diferentes pruebas para un correcto funcionamiento de la página:

* Comprobación de la conexión a la base de datos, inserción y borrado en distintas tablas.
* Comprobación en la inserción de datos en los formularios de loggin y register.
* Comprobación del filtrado para que no haya problemas de datos incoherentes.
* La muestra de coches se comprueba que salgan todos los coches que los usuarios han puesto a la venta.
* El formulario de venta se comprueba para que se tengan que rellenar todos los campos ya que son importantes.

DESPLIEGUE

Para el despliegue de la aplicación necesitamos un gestor o sistema de base de datos, un servidor web y un entorno de desarrollo.

El sistema operativo recomendado es Windows ya que no se han hecho pruebas para otro.

El explorador recomendado es Google Chrome, aun que se despliega en cualquier otro.

Para que el usuario pueda navegar por ella, no necesita permisos especiales, solamente crear una cuenta con un usuario para así tener acceso a todas las páginas del proyecto.

OPERACIONES

Las operaciones que se pueden realizar son las siguientes:

* Registro
* Inicio de sesión
* Borrado
* Inserción
* Filtrado
* Comprar

MANTENIMIENTO

La aplicación se puede automantener, aunque es recomendable tener en mente las actualizaciones de la base de datos y del navegador.

BIBLIOGRAFÍA

<https://www.arsys.es/blog/phpmyadmin#:~:text=PhpMyAdmin%20es%20una%20aplicaci%C3%B3n%20web,muy%20popular%20basado%20en%20PHP>.

<https://medium.com/linkit-intecs/mvc-architecture-14238e8b93bc>

<https://es.wikipedia.org/wiki/Bootstrap_(framework)>

<https://es.wikipedia.org/wiki/HTML>

<https://es.wikipedia.org/wiki/CSS>

<https://es.wikipedia.org/wiki/JavaScript>

<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>