



# SAN JUAN DE LA CRUZ

Grado en Desarrollo de Aplicaciones Web

Trabajo de fin de grado



# Sanova

**Presentado por:** Guillermo Pulido López

**Tutor:** Llanos Soro Moratalla

**Curso:** 2020-2021

## Contenido

1.- Introducción .....	2
1.1.- Descripción del proyecto y funcionalidades .....	2
1.2.- Motivación del proyecto .....	2
2.- Objetivos.....	3
3.- Metodología .....	3
3.1.- Planificación temporal del trabajo .....	3
3.2.- Etapas en la realización del proyecto.....	4
4.- Análisis y diseño lógico de la BD .....	5
5.- Diseño de la aplicación según el patrón MVC .....	6
5.1.- Modelo.....	6
5.2.- Vista .....	7
5.3.- Controlador .....	8
6.- Implementación e integración .....	9
6.1.- Herramientas y tecnologías utilizadas e investigadas .....	9
7.- Detalles de implementación .....	10
7.1.- Página de inicio .....	10
7.2.- Página de login .....	11
7.3.- Página de registro.....	11
7.4.- Zona de usuario.....	12
7.5.- Zona de administración.....	13
8.- Pruebas .....	15
9.- Aportaciones originales.....	15
10.- Conclusión .....	15
11.- Bibliografía y enlaces .....	16

## 1.- Introducción

### 1.1.- Descripción del proyecto y funcionalidades

Este proyecto trata de emular la web de un centro de salud encargado de gestionar la vacunación de la población, pero a pequeña escala.

Esta web permite a los usuarios que, tras registrarse, puedan solicitar una cita en el día y hora que les resulte más conveniente. Una vez solicitada la cita, un médico será asignado a cada usuario, permitiendo a partir de ahí la comunicación directa entre ambos mediante un chat en tiempo real.

Por otro lado, el área de los médicos cuenta con una gran cantidad de herramientas de gestión para: las solicitudes de cita, los chats con los pacientes, los informes de los pacientes...

Los médicos tendrán la capacidad de ver un listado completo con todos los usuarios que hayan solicitado una cita. A partir de este, podrán aceptar las citas que más les convengan, estableciendo a su vez la vacuna que va a ser administrada a cada usuario.

Una vez asignado el usuario, el médico contará con una pantalla donde puede filtrar por pacientes vacunados, no vacunados, edad, nombre... A través de esta pantalla, el médico tendrá la capacidad de acceder a toda la información del paciente, su informe médico, su chat y su información de usuario.

A parte de las funcionalidades anteriormente mencionadas, todos los usuarios dados de alta en la web, cuentan con una página de perfil, con la que pueden editar información básica de su usuario, y ver su médico asociado.

En conjunto, este proyecto trata de facilitar interacción entre pacientes y médicos, y acelerar el proceso de vacunación.

### 1.2.- Motivación del proyecto

Este proyecto nació de la necesidad de facilitar y acelerar la gestión de la vacunación a nivel mundial, debido a la situación de pandemia propiciada por el COVID-19.

La idea del proyecto vino dada tras un análisis de los métodos y herramientas utilizadas para gestionar la vacunación. La conclusión de este fue que los recursos para llevarlo a cabo eran muy limitados y arcaicos – al menos en el periodo de tiempo en el que se hizo el análisis –.

Todos estos factores, junto al abanico de posibilidades que me ofrecían los conocimientos adquiridos en el grado, fueron los que dieron lugar al proyecto “Sanova” como trabajo de fin de grado.

## 2.- Objetivos

Tal y como ha quedado definido en los apartados anteriores, el objetivo del proyecto es abordar una casuística existente en la actualidad, como es el proceso de vacunación a nivel global. El proyecto tiene como objetivo dar un servicio accesible, multidispositivo y eficaz tanto a profesionales sanitarios como a usuarios registrados. En el caso de los médicos, este proyecto también les ofrece un abanico de herramientas y paginas a través de las cuales acceder a información, comunicarse con pacientes y gestionar labores medicas de forma muy rápida y sencilla.

## 3.- Metodología

### 3.1.- Planificación temporal del trabajo

La planificación temporal del proyecto no vino dada en forma de plazos de tiempo preestablecidos, si no en forma de metas y objetivos. La decisión de planificar el desarrollo del proyecto de esta forma, se debió a la inexperiencia en la realización de proyectos fullstack con implementación de nuevas tecnologías.

Sin embargo, tras un análisis del proceso de desarrollo dado para llevar a cabo el proyecto Sanova, se distinguen varias fases en la temporalización que este ha tomado:

- Dos semanas: análisis de requisitos de la web, análisis de futuros problemas e implementación de nuevas tecnologías, esquematización de la base de datos, layout de pantallas...

- Dos semanas y media: analizar las tecnologías con las que desarrollar el proyecto, adquirir información sobre las nuevas tecnologías a implementar, y probar su funcionamiento e integración en el proyecto.
- Un mes: Desarrollo de la estructura, funcionalidades básicas y front de la web.
- Un mes: Desarrollo del backend del proyecto, ampliación de funcionalidades y corrección de errores.

### 3.2.- Etapas en la realización del proyecto

Al comienzo del proyecto, se establecieron varias etapas a seguir para el correcto desarrollo del mismo. Estas son:

1. Análisis de requisitos, desarrollo de un layout básico sujeto a cambios futuros, desarrollo de modelo E/R de la base de datos.
2. Investigación e implementación de nuevas tecnologías y librerías de frontend.
3. Desarrollo de la página principal del proyecto para la familiarización con la implementación de la librería de estilos de Bootstrap.
4. Desarrollo del área para pacientes.
5. Desarrollo del área para médicos.
6. Implementación de AJAX para obtener información del servidor por requests HTTP sin recargar las páginas.
7. Desarrollo de scripts implementando SSE y AJAX para el desarrollo del chat en tiempo real.
8. Implementación de editor de texto WYSIWYG en el área médicos para la redacción de informes.
9. Desarrollo del área de administrador con el patrón MVC.
10. Adaptación de la web para móviles (responsive).
11. Testeo y corrección de errores.

## 4.- Análisis y diseño lógico de la BD

Durante el análisis de requisitos para el desarrollo del proyecto, se tomó la decisión de crear una base de datos capaz de cubrir las necesidades básicas de la web, sin llegar a ser demasiado extensa o compleja y dificultar su uso.

Es por esa razón que se tomó la decisión de crear únicamente tres entidades, las cuales, tras aplicar una generalización, darían lugar a cinco entidades en total: “mensajes”, “usuarios”, “usu\_pacientes”, “usu\_medicos”, “vacunas”.

Las relaciones que toman estas entidades pueden apreciarse en la imagen 4.a, dando lugar al modelo E/R.

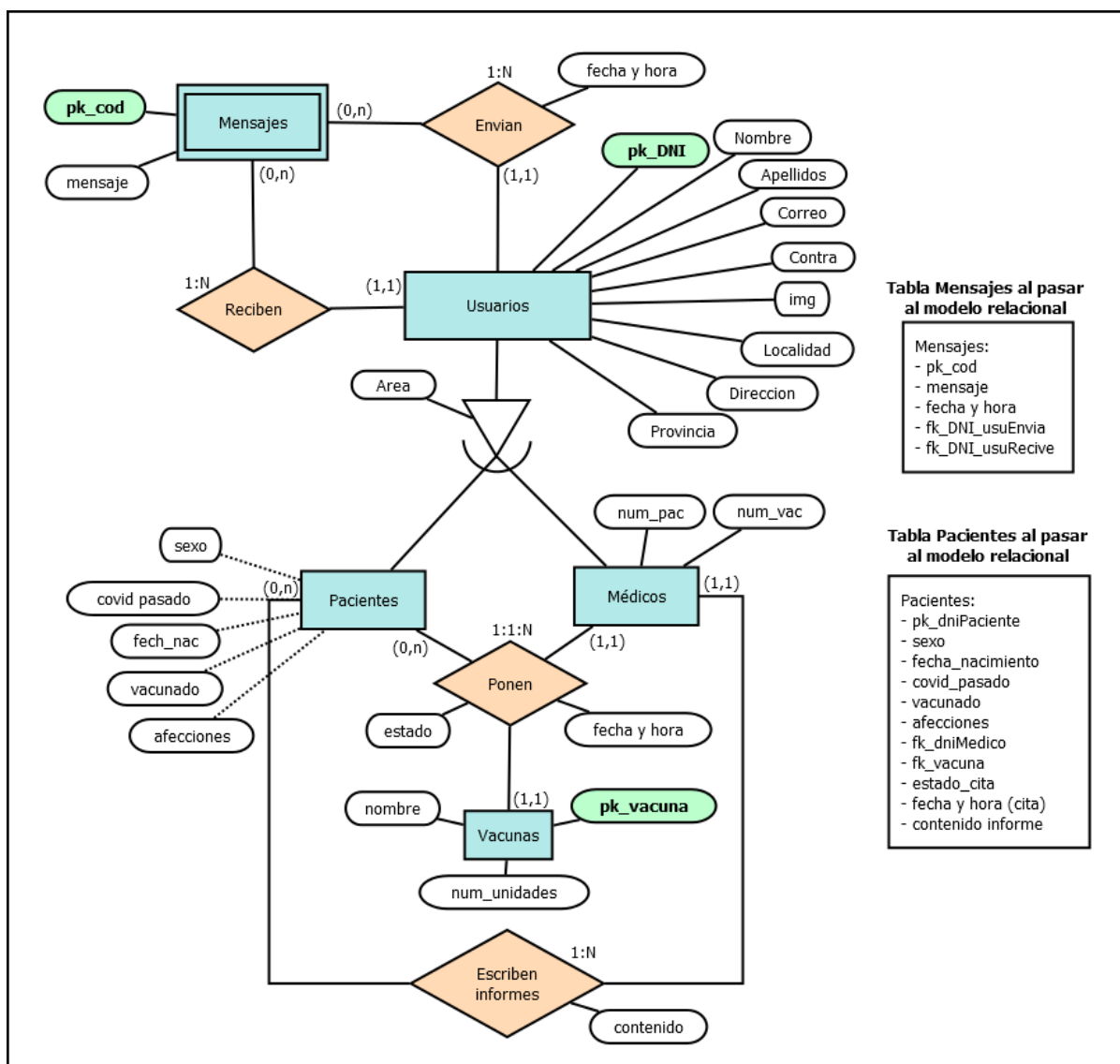


Imagen 4.a - Modelo E/R

En el caso de que quisiéramos dar un uso real a esta aplicación, sería conveniente diversificar la entidad “mensajes” en una entidad o tabla llamada “chat” por cada paciente asociado a un médico. Esto se debe a que actualmente la entidad “mensajes” almacena los registros de todos los chats de los pacientes con los médicos, y en el caso de existir una gran cantidad de los mismos, empezarían a surgir problemas de espacio, tiempo de carga y de respuesta...

## 5.- Diseño de la aplicación según el patrón MVC

Es necesario remarcar que debido a las características propias del lenguaje en el que ha sido desarrollada la web – PHP –, los apartados de médicos y pacientes fueron desarrollados sin seguir este patrón de diseño. Esto se debe a que PHP permite, de una forma mucho más sencilla, integrar el controlador en las vistas o el modelo, quedando como resultado únicamente dos archivos, en vez de tres.

El momento en el que se detectó esta falla en el patrón de diseño, se dio cuando este ya estaba en un estado muy avanzado de desarrollo como para hacer una reestructuración completa del mismo. Esta fue la razón por la que se decidió optar por no reestructurar todo el proyecto implementando MVC, si no utilizar este patrón en el desarrollo del apartado de administración de la web, ya que era el único que todavía no estaba en fase de desarrollo hasta ese momento.

Sabiendo esto, a continuación se describirá del patrón de diseño MVC implementado únicamente en el apartado de administración de Sanova.

### 5.1.- Modelo

En el modelo se encuentra un único archivo, llamado *clases\_model.php*, el cual contiene tres clases encargadas de obtener información de la base de datos. Estas clases son:

- Clase conectar: Esta clase contiene únicamente un método estático, el cual será llamado por el resto de clases en el momento de la instanciación de objetos de las mismas para abrir una conexión con la base de datos.

- Clase obtencion\_tablas: Los objetos de esta clase son instanciados por el archivo controlador llamado “paginas\_controller.php” utilizado para el filtrado de las peticiones entrantes para cargar una vista determinada.

Esta clase es la encargada de obtener información de las tablas “usuarios”, “usu\_pacientes”, “usu\_medicos” y “vacunas” para después ser presentada en el front a través de las vistas. Esta clase también cuenta con dos métodos que ayudan a presentar la información en las vistas en forma de select, llamados “select\_medicos” y “select\_vacunas”.

- Clase interaccion\_bbdd: Los objetos de esta clase son instanciados por el archivo controlador llamado “peticionesAjax\_controller.php”, utilizado para gestionar las peticiones entrantes a través de AJAX, que se realizarán a través de los archivos javascript asociados a cada una de las vistas.

Esta clase cuenta con un método encargado de controlar el tipo de interacción que se quiere hacer en con la BD, llamado “filtrar\_peticion”, el cual llama a los métodos encargados realizar las mismas. El resto de métodos de la clase son los que realizan la interacción con la base de datos, con acciones de tipo inserción, borrado o actualización de forma recursiva de todos los datos.

## 5.2.- Vista

En la carpeta de vistas, se encuentran únicamente tres archivos, que serán capaces de generar cinco páginas distintas en el área de administración de Sanova. Estos son:

- Dashboard: Este archivo compone una única vista, la del menú principal del área de administración de Sanova. Desde esta vista se puede acceder al resto de vistas, como son: registro de médicos, registro de vacunas, información de médicos, pacientes y vacunas. Es importante tener en cuenta que esta vista es la única desde la cual se puede acceder al registro de médicos y el registro de vacunas.



- Tablas: Este archivo compone tres vistas, mostrando en todas ellas información sobre pacientes, médicos y vacunas en forma de tabla. Estas vistas permiten consultar información, actualizar registros de la base de datos, o incluso borrar registros.

Todos los registros de la bases de datos que se presentan son susceptibles de ser borrados y sus campos actualizados, y en muchos de estos casos, los registros están condicionados o condicionan a terceros registros de diferentes tablas, es decir, están relacionados entre sí. Es por eso que todos los casos en los que existen relaciones entre registros, están contemplados para que al realizarse un cambio en la base de datos, se actualicen todas las relaciones de forma recursiva o en cascada automáticamente.

- Formularios: Este archivo compone dos vistas, mostrando en ambas un formulario de registro, con el cual añadir nuevos médicos en la base de datos, o registrar nuevas vacunas. Estas vistas únicamente se componen de un formulario y un control de validación del mismo, que finaliza enviando la información del registro al modelo utilizando AJAX.

### 5.3.- Controlador

En el controlador hay únicamente dos archivos, los cuales se encargan de filtrar las peticiones de vistas entrantes, la instanciación de los objetos pertinentes, y la llamada a las vistas correspondientes. Estos son:

- paginas\_controller.php: Este controlador se encarga de gestionar las peticiones de vistas entrantes, y para cada una de ellas instanciar el objeto correspondiente que estas necesiten para mostrar la información de esa vista. Este controlador conforma una especie de filtro de vistas e instanciación de objetos.
- peticionesAjax\_controller.php: Este controlador se encarga de gestionar las peticiones asíncronas realizadas con AJAX. Por cada una de las

peticiones, el controlador instancia un objeto de tipo *interaccion\_bbdd*, el cual contiene en su interior un método que filtra las peticiones y ejecuta las acciones requeridas por cada una de ellas. El controlador entonces recibe true si todas las acciones o métodos de interacción con la base de datos se han llevado a cabo correctamente, o falso si ha habido algún fallo.

## 6.- Implementación e integración

### 6.1.- Herramientas y tecnologías utilizadas e investigadas

En el desarrollo de este proyecto, han sido utilizadas nuevas tecnologías no aprendidas a través del ciclo, o profundización en tecnologías ya utilizadas:

- AJAX: La necesidad de hacer la web dinámica, implica que en muchos de los casos, será necesario refrescar o actualizar la pantalla para que se recarguen los datos. Siendo este sistema de obtención de datos, hemos optado por utilizar AJAX para realizar peticiones asíncronas al servidor, las cuales serán capaces de actualizar la pestaña en la que se encuentre el usuario en cada momento, sin necesidad de que este refresque o actualice la página.

Esta tecnología ha sido implementada en la gran mayoría de los formularios del proyecto, aumentando la experiencia de usuario, y permitiendo algunas funcionalidades como el chat entre pacientes y médicos.

- SSE o Server Sent Events: Este API está contenido en la interfaz *EventSource*, y se utiliza para abrir una conexión persistente al servidor, a través de la cual el cliente recibe eventos, con un funcionamiento muy similar al AJAX. Para llevar a cabo esta tarea, se utiliza el objeto *EventSource*, en el cual se establece la ruta del script que será el encargado de generar los eventos. Al abrir la conexión, el script podrá recibir información, pero una vez establecida la conexión, el script solo podrá enviar información en formato JSON al cliente que haya abierto la conexión.

En el proyecto, esta tecnología fue utilizada para desarrollar el chat entre pacientes y médicos en tiempo real, ya que al abrir la conexión, estarían recibiendo mensajes entrantes del usuario al otro lado del chat.

- JQuery: Esta tecnología fue mencionada en el grado, sin embargo, la he implementado en el proyecto en el apartado de registro de usuario, para darle funcionalidad a los fieldsets, la barra de progreso y otros elementos.
- Bootstrap: Esta librería de estilos fue mencionada en el grado, sin embargo, he profundizado ampliamente en ella para generar la gran mayoría de estilos, sin necesidad de hojas de estilo CSS externas.
- Tecnologías para el retoque fotográfico o la creación de iconos y otros elementos gráficos, con programas tales como Illustrator y Photoshop.

## 7.- Detalles de implementación

### 7.1.- Página de inicio

La página de inicio de Sanova toma el nombre de index.php, de esta forma el archivo se abre automáticamente una vez se acceda a la carpeta del proyecto.

La página de inicio conforma una breve presentación del contenido y el objetivo de la web, separándolo en secciones diferenciadas, presentando información básica de la web, datos numéricos generales sobre el número de pacientes vacunados, el número de médicos... empresas asociadas y redes sociales.

Aparte de las secciones informativas, esta página también contiene dos modales:

- Contacto: A través de este modal, cualquier persona sin la necesidad de estar registrada en la web podrá ponerse en contacto con Sanova enviando un correo.
- Inicio de sesión: Este modal permite iniciar sesión a los usuarios ya registrados, registrarse, o recuperar la contraseña en el caso de que esta haya sido olvidada por el usuario. El formulario que conforma el modal de

inicio de sesión está controlado ante todo tipo de errores, cross-site scripting o inicios de sesión maliciosos.

## 7.2.- Página de login

Tal y como ha sido expuesto en el punto anterior, la página de login o inicio de sesión está contenida dentro de la página de inicio en forma de modal. La única forma de acceder a este modal de login es a través de la página de inicio, debido a que esta es la única página accesible sin tener una sesión iniciada.

El login de inicio de sesión cuenta con dos campos, que requieren el DNI y la contraseña asociada al DNI introducido. Este formulario está protegido contra errores, contra cross-site scripting y contra intentos de inicio de sesión maliciosos. En todos estos casos, se avisa a través de una alerta del tipo de error que se está dando en caso de que el inicio de sesión fallase.

## 7.3.- Página de registro

La página de registro está dividida en cuatro fieldsets entre los que se puede avanzar y retroceder en el caso de que todos los campos del mismo sean correctos. Esta separación del formulario viene dada con la intención de hacerlo más corto horizontalmente, y para generar una barra de proceso haciendo que el usuario tenga un feedback más directo.

Para completar el registro y enviarlo, es necesario:

- Completar todos los campos marcados como obligatorios con un asterisco.
- Escribir un DNI válido, cuyos ocho primeros valores sean dígitos, y finalice con un carácter.
- Escribir una contraseña con una longitud igual o mayor a seis dígitos.
- Repetir la contraseña para establecer una doble comprobación.

Una vez completado y enviado el registro, realizaran las siguientes comprobaciones:

- El usuario debe ser mayor de 18 años.

- El usuario no puede estar vacunado del coronavirus.
- El DNI introducido no puede estar registrado en la base de datos.

Una vez finalizado el registro, el usuario es redireccionado a una página en la que podrá comprobar si el registro se ha llevado a cabo con éxito, o si este ha sido denegado y el usuario no ha podido ser registrado.

#### 7.4.- Zona de usuario

En Sanova existen dos tipos de usuarios, dejando de lado al administrador:

- Pacientes: El área de pacientes cuenta únicamente con dos ventanas, la ventana de usuario y la ventana de cita.
  - Ventana de usuario: En esta ventana, los pacientes pueden consultar información básica de su perfil, consultar su médico asociado - en el caso de que lo tengan -, y modificar o actualizar sus datos personales.
  - Ventana de cita: Esta ventana es variable, dependiendo del estado de la cita del paciente que ingrese a ella.
    - *No solicitada*, el paciente podrá solicitar una cita en el día y la hora que más le convengan.
    - *Solicitada*, el paciente deberá esperar a que un médico acepte su cita, mientras tanto, en esta pantalla puede consultar la fecha y horas solicitadas.
    - *Establecida*, la cita del paciente ya habrá sido aceptada por un médico. A través de esta página, el paciente podrá comunicarse de forma directa con el médico y obtener información sobre la fecha de su cita.
    - *Cancelada*, la cita del paciente puede ser cancelada por varios motivos. En el caso de ser cancelada, el paciente se encontrará con esta pantalla con una breve explicación sobre las posibles causas de cancelación que se hayan podido dar.
    - *Concluida*, en este punto el paciente ya estará vacunado y en esta página solo encontrará información sobre la su cita.

- Médicos: El área de médico cuenta con varios apartados:
  - Solicitudes pendientes: Este apartado permite al médico acceder a un listado con todos los pacientes que hayan solicitado una cita y estén a la espera de ser aceptados. El medico podrá consultar información básica sobre cada paciente, además de la fecha y hora de cita solicitadas. Cuando el medico acepte una cita, podrá elegir entre una de las vacunas disponibles para ser asignada a ese paciente. Este apartado es común para todos los médicos.
  - Chats: Este apartado contiene un buscador a través del cual el médico puede buscar entre los pacientes que tenga asignados. Una vez seleccionado el paciente, el medico verá tanto el informe sanitario del paciente como un chat a tiempo real para comunicarse entre ambos.
  - Citas: Este es el apartado principal del médico, ya que contiene un listado con todos los pacientes asociados al mismo. En esta pestaña, el medico puede filtrar pacientes por: vacunados, no vacunados, edad de mayor a menor, edad de menor a mayor, ordenados por nombre, ordenar por fecha de cita o buscar de forma directa a través de un buscador. A través de este apartado, el medico también puede acceder de forma directa a información del paciente, al chat del paciente, al informe del paciente, aceptar o cancelar la cita con el paciente.
  - Informe: A este apartado solo es posible acceder a través de los apartados anteriores, ya que el acceso determinará el informe del paciente que se va a presentar. Esta página contiene un editor de texto WYSIWYG para que el medico pueda crear o modificar el informe del paciente, el cual quedará registrado en la base de datos en forma de cadena de texto de código HTML.

### 7.5.- Zona de administración

Respecto al apartado de administración de la web, la gran mayoría de su funcionamiento backend y organización MVC ha quedado definido en el punto 5

“Diseño de la aplicación según el patrón MVC”, sin embargo, a continuación se explica brevemente el funcionamiento frontend de esta área.

El administrador, al momento de iniciar sesión, es redireccionado directamente al dashboard, en el que puede encontrar los enlaces a las tablas de pacientes, médicos y vacunas, a través de las cuales puede consultar información, modificar o guardar registros.

A través del dashboard, el administrador también podrá acceder a los formularios de médicos y vacunas a través de los cuales podrá llevar a cabo la inserción de nuevos registros.



## 8.- Pruebas

Las pruebas realizadas en la web de Sanova solo podrán consultarse a través del formato digital de este mismo documento, ya que se trata de dos hojas Excel insertadas en el mismo.



Pruebas Sanova  
Desarrollador.xlsx



Pruebas Sanova  
Usuario.xlsx

## 9.- Aportaciones originales

Anteriormente, en el punto seis llamado *implementación e integración*, se explican las nuevas tecnologías o tecnologías en las que se ha profundizado durante el desarrollo del proyecto.

También hay que remarcar la personalización de la marca llevada a cabo, al diseñar el logo y otros elementos utilizados en el proyecto. Además, este proyecto ha sido diseñado de tal forma, que en el caso de que se quisiese retomar, ampliar, modificar o utilizar como una web real, sería una tarea muy sencilla, ya que estas posibilidades han sido contempladas en la creación del proyecto.

También es necesario tener en cuenta, que ninguna de las funcionalidades o implementaciones de elementos o tecnologías realizadas en el desarrollo del proyecto, ha sido llevada a cabo gracias a tutoriales o guías step by step. Todo el front, back, tecnologías utilizadas, funcionalidades, organización... han sido desarrollados únicamente con documentaciones oficiales, libros y foros de dudas como StackOverflow.

## 10.- Conclusión

En cuanto a conocimientos adquiridos, se pueden destacar algunos como la aplicación del patrón de diseño MVC en un proyecto real a pequeña escala, la implementación de nuevas tecnologías como las ya mencionadas en anteriores puntos de este documento, pero sobre todo, la labor de investigación y



descubrimiento de tecnologías no conocidas hasta el momento, aun sin haber sido implementadas finalmente, las cuales podrían ser utilizadas en futuros trabajos.

Dejando a un lado los conocimientos adquiridos, también cabe mencionar la importancia de una buena planificación inicial a la hora de llevar a cabo un proyecto, y la constancia en el desarrollo, ya que sin ella, la cantidad de funcionalidades finales se vería ampliamente reducida.

## 11.- Bibliografía y enlaces

### **W3School**

<https://www.w3schools.com/>

### **Stackoverflow**

<https://stackoverflow.com/>

### **Documentación de Bootstrap**

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

### **Documentación de PHP**

<https://www.php.net/docs.php>

### **Libro: Desarrollo web en entorno servidor – Xabier Ganzábal**

<https://www.amazon.es/Desarrollo-entorno-servidor-Inform%C3%A1tica-comunicaciones/dp/8491713492>

### **Pexels**

<https://www.pexels.com/>

### **Flaticon**

<https://www.flaticon.com/>