



Escuela
Politécnica
Superior

Desarrollo de una aplicación para reservar citas online mediante Wordpress



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:

Jose María Muela Bernabeu

Tutor/es:

David Tomás Díaz

Jose Vicente Vicedo



Universitat d'Alacant
Universidad de Alicante

Enero 2022

Índice

Agradecimientos	7
1. Introducción y justificación del proyecto	9
2. Objetivos	13
3. Metodología	15
4. Análisis de requisitos	21
4.1 Tipos de requisitos	21
4.2 Pirámide de requisitos	22
4.3 Distribución de los requisitos	23
4.3.1 Requisitos funcionales dinámicos (Cliente)	23
4.3.2. Requisitos funcionales estáticos (Cliente)	24
4.2.3 Requisitos funcionales dinámicos (Empresa)	24
4.3.4 Requisitos no funcionales	25
4.4 Diagramas de casos de uso	27
5. Estado de la cuestión	29
5.1 Diferencias en plataformas de bajo coste	29
5.2 Basado en el caso de uso	33
5.2.1 La Tijera	33
5.2.2 SalonBooking	35
5.2.3 ZittApp	36
5.2.4 Diferencias con la aplicación	38
6. Herramientas y tecnologías	39
6.1 Mockups	39
6.2 Funcionamiento de la web en local	40
6.3 Uso de WordPress para la aplicación	43
6.4 Plugins de WordPress	45
6.4.1 Proceso de instalación de un plugin	45
6.4.2 Plugins empleados en la aplicación	47
6.5 Tecnologías web	50
7. Arquitectura del sistema	53
8. Funcionalidades del sistema	55
8.1 Acceso a la aplicación multiplataforma	55
8.2 Roles de usuario	55
8.3 Funcionamiento después de registro	59
8.4 Funcionalidades en el usuario cliente	60
8.5 Funcionalidades en el usuario empresa	63
8.6 Ajustes importantes sobre el Calendario de citas con el plugin Easy Appointment	67

9. Dificultades afrontadas y experiencias durante el desarrollo	69
9.1 Problemas encontrados	69
9.2 Soluciones planteadas	70
9.3 Aclaraciones sobre cuándo usar WordPress	77
9.4 Aspectos importantes de los plugins a utilizar	78
9.5 Curva de aprendizaje sobre WordPress	79
10. Conclusiones y trabajo futuro	81
Referencias	83
Anexos	85

Índice de figuras

Figura 1: Gestión y planificación del proyecto con Diagrama de Gantt	18
Figura 2: Vista de inicio para usuario sin loguear	19
Figura 3: Vista de Servicios para cliente logueado	19
Figura 4: Vista de Citas para empresa logueada	20
Figura 5: Fuente: Modelado de casos de uso	22
Figura 6: Caso de uso Inicio de sesión	28
Figura 7: Caso de uso de Empresa	28
Figura 8: Caso de uso de Cliente	28
Figura 9: Pantalla principal de La Tijera y sus servicios	35
Figura 10: Horario de citas de SalonBooking	36
Figura 11: Horario de citas de ZittApp	37
Figura 12: Panel de control de XAMPP con servicios corriendo	43
Figura 13: Ruta de archivo WordPress en XAMPP	44
Figura 14: Búsqueda e instalación de un plugin	45
Figura 15: Plugin activado en el panel de control de WordPress	46
Figura 16: Arquitectura del sistema	53
Figura 17: Formulario de registro para clientes	57
Figura 18: Formulario de registro para empresas	58
Figura 19: Listado de miembros registrados en la aplicación	59
Figura 20: Menú de un usuario registrado con rol de cliente	60
Figura 21: Menú de un usuario registrado con rol de empresa	60
Figura 22: Catálogo de productos	61
Figura 23: Valoración de un usuario a un establecimiento del catálogo	61
Figura 24: Reserva de cliente a establecimiento	63
Figura 25: Descripción de producto de empresa	64
Figura 26: Calendario de reservas de establecimiento	66
Figura 27: Ajustes generales del sistema de reservas de WordPress	67
Figura 28: Datos de la cita en ventana emergente del calendario de reservas	68
Figura 29: Código para cambiar credenciales de usuario en phpMyAdmin	72
Figura 30: Ruta de archivos de la base de datos en XAMPP	74
Figura 31: Archivos para volcar copia de seguridad	75
Figura 32: Pasos para volcar backup	76
Figura 33: Curva de aprendizaje de los CMS	80
Figura 34: Mockups 1	85

Figura 35: Mockups 2	86
Figura 36: Mockups 3	86

Índice de tablas

Tabla 1: Requisitos funcionales dinámicos de Cliente	23
Tabla 2: Requisitos funcionales estáticos de Cliente	24
Tabla 3: Requisitos funcionales de Empresa	24
Tabla 4: Requisitos no funcionales de rendimiento y eficiencia	25
Tabla 5: Requisitos no funcionales de disponibilidad	25
Tabla 6: Requisitos no funcionales de seguridad	26
Tabla 7: Requisitos no funcionales de fiabilidad	26
Tabla 8: Requisitos no funcionales de usabilidad	27
Tabla 9: Problemas encontrados y soluciones planteadas a lo largo del desarrollo	71

Agradecimientos

Quiero agradecer toda la ayuda depositada en este trabajo tanto a nivel técnico como emocional. En primer lugar, quiero agradecer toda la labor a mis tutores David Tomás y José Luis Vicedo por prestar toda la ayuda posible para realizar este trabajo de la mejor forma.

En segundo lugar, y no por ello menos importante, agradezco de corazón el empuje moral que necesitaba en varios momentos de mis padres, mi hermano y mi pareja, que han estado apoyándome desde el primer minuto hasta el último sin cesar y, sin duda, no lo habría logrado sin ellos. También quiero hacer mención a un ex compañero de grado (Iván Mañús Murcia), un gran compañero y amigo que también supo afrontar este trabajo con entereza y coraje.

1. Introducción y justificación del proyecto

En los últimos años las empresas han ido creciendo a todos los niveles, ya sea a comercial como a tecnológico entre otros. El crecimiento fue exponencial en las grandes empresas, aunque no en todas. Sin embargo, el mayor perjudicado siempre ha sido el pequeño y/o mediano comercio, que también podemos nombrar como **pymes** (pequeña y mediana empresa). Los ingresos económicos y el uso de tecnologías de estas últimas no se puede comparar con las de las grandes empresas en ningún aspecto. Por ello mismo hay que apoyar al pequeño comercio, como puede ser el caso de gimnasios, panaderías y peluquerías de barrio, entre otros sectores de trabajo, que se pueden hallar tanto en pueblos como ciudades de España. Además, por si fuese poco, el agujero económico formado por la pandemia que actualmente estamos sufriendo, ha llevado a cabo que varios de esos pequeños comercios hayan tenido que cesar definitivamente debido a la imposibilidad de reflotar o mantener el negocio por parte de los propietarios. Las distintas formas de apoyar al pequeño comercio es ayudarles a amoldarse y a actualizarse a los tiempos que corren a nivel tecnológico. Una de esas formas es darse a conocer por medios que hace años era impensable, es decir, por medio de páginas web multiplataforma y las redes sociales, entre otras.

Hoy en día la mayoría de comercios son negocios familiares que se ven necesitados tanto a nivel económico como tecnológico y que, pese a darse a conocer por la clientela fiel y la más cercana a sus establecimientos, además de desempeñar una buena labor como profesionales, con el paso del tiempo necesitan ayuda para mantener y modernizar sus negocios mediante su adaptación a las nuevas tecnologías. De esta forma, podrán seguir creciendo a nivel profesional.

Por tanto, en este trabajo se pretende llevar a cabo el desarrollo de una aplicación multiplataforma basada en *WordPress* en la que se pueda realizar un sistema de citas a diferentes negocios desde la perspectiva de un cliente y, por parte de cada empresa, llevar un control de las citas confirmadas. El motivo por el que se decide utilizar este *framework* de desarrollo es por la facilidad y rapidez que tiene para implementar una aplicación que solucione el problema de desconocimiento de las nuevas tecnologías. Además, también se puede llevar a cabo un mantenimiento y desarrollo adecuados al tratarse de una aplicación realizada a través de una plataforma de bajo coste.

La intención con este trabajo es apoyar al pequeño y mediano empresario, centrándose en un nicho de mercado concreto, pero que a su vez sirva de ayuda a todos los servicios que pasen por una situación similar.

El nicho de mercado en el que me he basado para desarrollar la aplicación multiplataforma es en el sector de las peluquerías, dado que son establecimientos pequeños cuyos recursos tecnológicos son insuficientes para adaptarse a la actualidad. Además, no se pueden permitir ningún tipo de sistema de reservas *online* dado los elevados costes tanto de desarrollo como de mantenimiento del mismo. Por tanto, se considera necesario que, al igual que los grandes comercios ya poseen, los pequeños tengan alguna plataforma para llevar a cabo la gestión de citas.

El apoyo proporcionado con este trabajo es mediante una aplicación intuitiva y sencilla con la cual los clientes puedan realizar reservas en un día y hora concretos para cualquiera de los servicios que ofrezca el negocio. Se trata de un sistema multiplataforma utilizando las diferentes tecnologías y herramientas actuales para dar a conocer a los pequeños negocios que se adhieran más allá de los clientes habituales y ciudadanos de la ciudad ubicada. Además, la propuesta de realizar aplicaciones para negocios con el fin, entre otros, de pedir citas o reservas online, llegó para quedarse sobre todo cuando emergió en nuestras vidas la pandemia de la COVID-19 [1].

Otro aspecto a tener en cuenta es que el cliente podrá tener una comunicación con la empresa de una forma más fluida y a cualquier hora por medio de la aplicación, debido a que los métodos de contacto con este tipo de negocio suele ser por llamada o asistiendo presencialmente. El primer servicio no asegura un contacto a todas horas, puesto que se puede dar el caso de que el servicio telefónico no esté disponible en un momento determinado o un período más prolongado, mientras que el segundo es ideal para aquellos clientes habituales que residen cerca del negocio, pero no para los que dependen de un vehículo con el que desplazarse. Por tanto, hay que considerar que, para la gran mayoría del público en la actualidad, el uso de una aplicación es el método ideal para contactar con una empresa en cualquier momento y lugar.

Gracias a la aplicación que se quiere realizar con este trabajo, hay que considerar que subsanaremos la problemática que sufren los pequeños comercios que no tienen medios

para darse a conocer más allá de un pueblo o ciudad y de sus clientes habituales. Además, se seguirán manteniendo las medidas sanitarias e higiénicas pertinentes para que, al haber citas para cada cliente en diferentes franjas horarias los días de la semana que trabaja el negocio, no haya exceso de aforo ni clientes esperando en la entrada del local.

Con respecto al resto de la memoria, se ha estructurado de forma en la que se ha seguido todos los pasos a nivel de análisis y desarrollo para obtener el resultado final. En la segunda sección se comentan los objetivos que tiene este trabajo, pasando por la tercera donde se explica la metodología seguida para completar cada uno de ellos. Se pasa por las secciones cuatro para tratar el análisis de requisitos, cinco para comparar esta aplicación con algunas similares en el mercado y seis, siete y ocho para tratar las herramientas utilizadas para el desarrollo del trabajo, la arquitectura del sistema y el correcto funcionamiento de la aplicación. En la sección nueve se explican los problemas ocasionados a lo largo del desarrollo, las soluciones propuestas para cada una y la curva de aprendizaje de la plataforma usada. En la última sección se hace una conclusión del trabajo realizado con unas posibles implementaciones futuras.

2. Objetivos

El objetivo principal de este proyecto es analizar, diseñar, implementar y desarrollar una aplicación de citas multiplataforma basada en *WordPress*. El propósito principal es que un cliente que accede a la aplicación sea capaz de localizar un negocio que contenga un servicio y trabajador concretos y así poder realizar una reserva para una fecha y hora. Con ello también se le da visibilidad a los servicios que realizan los pequeños comercios. Para alcanzar dicho objetivo, es necesario abordar una serie de objetivos secundarios de forma que, una vez concluidos, se muestre el resultado final esperado. Dichos objetivos son:

1. Analizar los problemas a los que se enfrentan los pequeños comercios, tales como el desconocimiento de la tecnología y el coste elevado tanto del desarrollo como del mantenimiento.
2. Estudiar y proponer una posible solución de bajo coste para subsanar dichos problemas por medio de una aplicación.
3. Proponer un caso de uso y desarrollar una aplicación multiplataforma que permita la digitalización de un pequeño negocio a un coste asumible.
4. Experimentar *WordPress* como *framework* de diseño de aplicaciones multiplataforma de bajo coste.
5. Evaluar *WordPress* como *framework* de diseño de aplicaciones multiplataforma.

3. Metodología

En este punto, se explicará la metodología de trabajo seguida de principio a fin durante el transcurso del TFG, la cual se dividirá en una serie de pasos a seguir para la consecución de los objetivos del proyecto. En primer lugar, hay que comentar cómo surgió la idea para llevar a cabo el trabajo sobre este tema y cuáles son los pasos que se han ido siguiendo para cubrir los objetivos previstos y, en segundo lugar, cómo se ha desarrollado e implementado esta idea y con qué programas, herramientas y aplicaciones se han llegado a trabajar para cumplir con expectativas el proyecto.

Comenzamos por comentar el nacimiento de esta idea para llevar a cabo este proyecto. Todo comenzó a raíz de un análisis de mercado en el que se puede ver cómo en los últimos años los ingresos, tecnologías, prestigio, etc. es mayor en las grandes empresas y de mayor renombre que en las pymes. Además, el daño que ha provocado la pandemia COVID-19 ha hecho ver todavía más esa diferencia que estamos comentando, ya que hay pequeños negocios que no han podido reabastecerse ni han recibido ayudas económicas, lo que les ha condenado a cesar. El negocio en particular sobre el que aplicar mi propuesta de TFG surgió de forma que, en la búsqueda de un nicho de mercado y un pequeño local necesitado de nuevas tecnologías para darse a conocer y realizar reserva de citas online, me encontré con una peluquería. Concretamente, un local regentado por alguien que conozco personalmente y me facilitaba llevar a cabo este proyecto idea por dos razones:

1. Poder realizar el TFG sobre un caso real, comentando previamente esta propuesta con mis tutores.
2. Poder ayudar a un pequeño negocio necesitado de tecnología y explorar la posibilidad de utilizarlo en ayudar a más negocios relacionados con el mismo nicho de mercado.

En segundo lugar y una vez dado con la idea del proyecto, las reuniones con los profesores y el cliente por separado y el análisis de mercado realizado anteriormente, comienza el periodo de desarrollo de la aplicación. Una vez se ha realizado la reunión con el cliente y se le ha comentado la idea de la aplicación, éste ha transmitido las necesidades que le gustaría cubrir con la aplicación. El diseño lo deja el cliente a nuestra disposición

siempre y cuando cumpla lo solicitado. Antes de comenzar con el desarrollo, primero se realiza el análisis de requisitos con la idea de qué debería de proporcionar y tener la aplicación que se va a desarrollar con la intención de cubrir las necesidades básicas del cliente. Acto seguido, se ha decidido dibujar (a lápiz y papel primero y con un programa *online* después), un boceto o también conocido en este ámbito como **mockup**, el cual es una muestra previa de cómo se mostraría la aplicación ya desarrollada tanto en versión web como en móvil y/o tablet, ya que otro propósito de este trabajo es que el sistema sea multiplataforma. Después de que, tanto el cliente del negocio como los tutores de este trabajo, den su pertinente aprobación a continuar con el proyecto, se lleva a cabo una búsqueda de plantillas (como por ejemplo con la herramienta *BootStrap*¹) para obtener una editable y cómoda que podamos adaptar y moldear a lo que realmente queremos para nuestra aplicación.

Finalmente, el desarrollo e implementación de la aplicación, la cual se trata de la cuna del proyecto, se lleva a cabo por medio de la herramienta **WordPress**² y varios *plugins* que proporciona, debido a que presenta las siguientes ventajas:

- Es tendencia y tal y como indica *WebEmpresa* [2], está presente en más del 40% de la web, considerándose como una solución versátil para muchas empresas.
- Se trata de un programa sencillo de usar si se tienen conocimientos de algunos lenguajes de programación.
- Intuitivo e ideal para pequeñas empresas que no requieran un elevado coste por mantenimiento web.

Sin embargo, como toda herramienta, no todo son ventajas, debido a que las aplicaciones web realizadas con este programa carecen de un buen sistema de seguridad. De todas formas, para este trabajo no es un problema grave, ya que no se pedirán ni almacenarán una gran cantidad de datos personales para registrarse en la aplicación, como puede ser el caso del número de tarjeta de crédito u otros muy reseñables.

Para llevar a cabo el proyecto en condiciones, es idóneo marcarse una distribución del tiempo que se le tiene que emplear a cada uno de los objetivos que se marca en éste. Pese a ser prácticamente imposible marcar el tiempo exacto que se tarda en cumplir cada

¹ <https://getbootstrap.com/>

² <https://es.wordpress.org/>

objetivo, hay que intentar realizar espacios temporales que sean aproximadamente precisos a la realidad.

Esta tarea se puede llevar a cabo mediante herramientas de gestión y planificación de proyectos, en la que se realizará una distribución de los puntos del proyecto en distintos espacios de tiempo. Estos se encontrarán ordenados desde el primer punto hasta el último, teniendo cada uno una duración determinada según el tiempo que se requiere para realizar cada uno de los objetivos.

Esta planificación de tiempo-trabajo para la realización del proyecto se ha realizado mediante el programa **OpenProject**³. Es una herramienta que se puede descargar para sistemas operativos Linux o MacOS de forma gratuita y también utilizar de forma *online* mediante un navegador. Tiene varias funcionalidades con la que poder planificar un proyecto de distintas temáticas, como pueden ser:

- **Tablero Kanban** que sirve para clasificar cada tarea u objetivo a realizar en el proyecto por grupos o fases, es decir, según si esas tareas por separado están en espera de ser aceptadas por el cliente, en proceso, aceptada, rechazada, etc.
- Un **diagrama de Gantt** en el que poder comprobar de forma gráfica los espacios temporales de cada tarea marcada en el trabajo en cuestión, diferenciándose entre sí por las fechas marcadas e incluso por colores. Además, mediante unas flechas que van de un espacio a otro podremos apreciar si una tarea depende de su predecesora para poder iniciarla.
 - Dentro de cada tarea con su espacio temporal correspondiente, podemos desmembrar la misma en pequeños espacios temporales de otro color para diferenciarlo del anterior, como por ejemplo:
 - **Herramientas y tecnologías:** Cada subtarea sería cada herramienta y tecnología alternativas con la que poder desarrollar la aplicación que se pretende en este trabajo.
 - **Arquitectura del sistema:** Se puede dividir esta tarea en varias, siendo una el formulario y registro de un usuario, otra el apartado para pedir citas de cada servicio, otra el índice de una web o landing page, etc. de forma que el diagrama sea más concreto con las tareas a realizar en el proyecto.

³ <https://www.openproject.org/>

- Marcar una **hoja de ruta** en la que los paquetes de trabajo (tareas) se diferencian por su ID y, de esta forma y por escrito, cuáles se realizan primero y cuáles después.
- Englobar tareas en grupos (**sprints**), de forma que para pasar de un *sprint* a otro, se ha de realizar por completo todas las tareas que conforman el *sprint* anterior.

A efecto gráfico, se puede apreciar en la Figura 1 un diagrama de Gantt por medio del programa mencionado en este punto, con los pasos a seguir divididos en líneas y con sus fechas tanto de inicio como de finalización. Todo esto se realiza con la intención de planificar paso por paso lo que será la realización del proyecto.

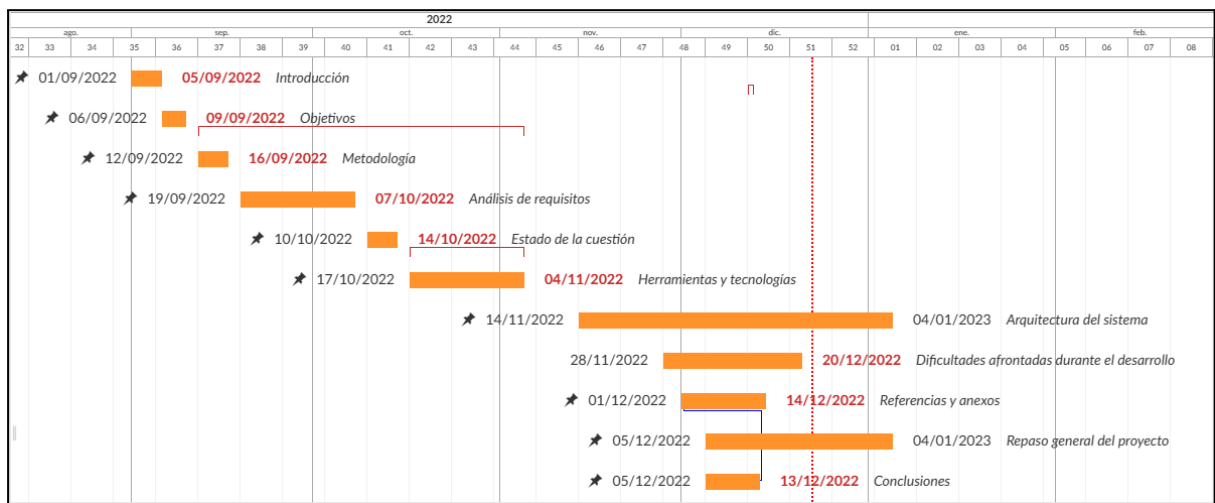


Figura 1: Gestión y planificación del proyecto con Diagrama de Gantt

En cuanto a la realización de los bocetos de la aplicación, para que el cliente tenga una idea del diseño y funcionamiento de la misma, se ha utilizado la herramienta **Moqups**⁴, en la que se puede apreciar una serie de páginas reconocidas cada una por su url. Contiene vistas para usuarios no registrados, clientes y empresas. Algunas de ellas tendrán aspectos en común y otras serán completamente distintas según el tipo de usuario que acceda a la aplicación tanto vía ordenador como móvil.

Podemos apreciar, en las Figuras 2, 3 y 4, algunas vistas para comprobar cómo sería la aplicación, a nivel de boceto, utilizando la herramienta comentada anteriormente.

⁴ <https://app.moqups.com/>

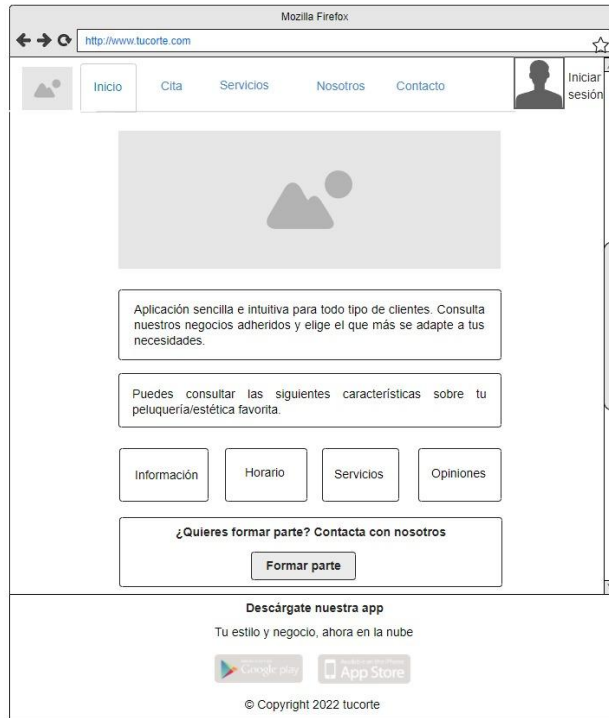


Figura 2: Vista de inicio para usuario sin loguear

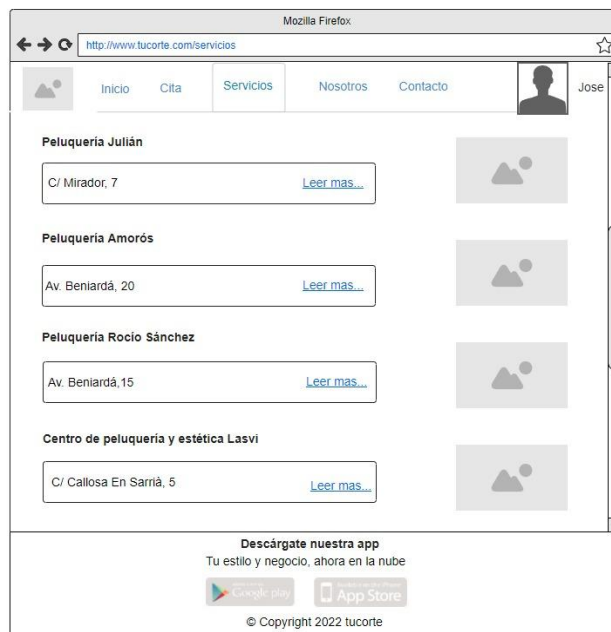


Figura 3: Vista de Servicios para cliente logueado

Mozilla Firefox
<http://www.tucorte.com/citas>

Inicio Usuarios Añadir Servicios Mi negocio Citas Jose

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
09:00						
10:00						
11:00				ana	oscar	
12:00						
13:00			pablo			
14:00						
15:00						
16:00		ana		julia		
17:00				julia		
18:00						
19:00						
20:00						

Peluquería Otros servicios
 Estética

Descárgate nuestra app
 Tu estilo y negocio, ahora en la nube
[Google play](#) [App Store](#)
 © Copyright 2022 tucorte

Figura 4: Vista de Citas para empresa logueada

4. Análisis de requisitos

En este punto vemos el análisis de requisitos, en el cual se lleva a cabo la determinación de expectativas del usuario sobre un producto o servicio que se encuentra en el mercado o que acaba de llegar al mismo. En otras palabras, consiste en la capacidad de resolver un problema o un conjunto de ellos por parte del usuario o bien que el sistema en sí cumpla o tenga una serie de condiciones para su correcto funcionamiento. Los requisitos también son conocidos como **especificaciones funcionales**. Se trata de un elemento fundamental y básico para llevar a cabo un proyecto cualquiera, en el que se analiza las funcionalidades que puede realizar un usuario sobre ese producto, servicio o sistema. Por esa razón, se utiliza de forma asidua en la gestión de proyectos. El objetivo es conseguir que el sistema se ajuste a las necesidades del cliente en lugar de que el propio cliente tenga que hacer un esfuerzo para adaptarse al funcionamiento del sistema.

4.1 Tipos de requisitos

Hay varios tipos de requisitos que podremos diferenciar claramente una vez los vayamos localizando gracias al estudio que se realiza sobre las acciones, condiciones y necesidades del sistema final y del usuario. Se dividen en los siguientes:

- **Características** (*Features*): Representan requisitos de alto nivel del sistema software. En programas de Modelado de Casos de Uso (como por ejemplo **Rational Rose**⁵), se consideran un tipo de requisitos debido a que son fáciles de documentar, discutir y comunicar.
- **Requisitos funcionales**: Son características que necesitan un mayor nivel de detalle para poder diseñar, implementar y verificar. En *softwares* como **Rational Rose** o **Bonita**⁶ entre otros, son considerados casos de uso.
- **Requisitos no funcionales**: Son especificaciones textuales que se asocian a los casos de uso. También son conocidos como requisitos suplementarios.

⁵ <https://www.ibm.com/support/pages/ibm-rational-rose-enterprise-7004-ifix001>

⁶ <https://es.bonitasoft.com/>

4.2 Pirámide de requisitos

Para realizar un análisis de requisitos, es conveniente conocer la **Pirámide de requisitos [3]** y su pertinente **trazabilidad**.

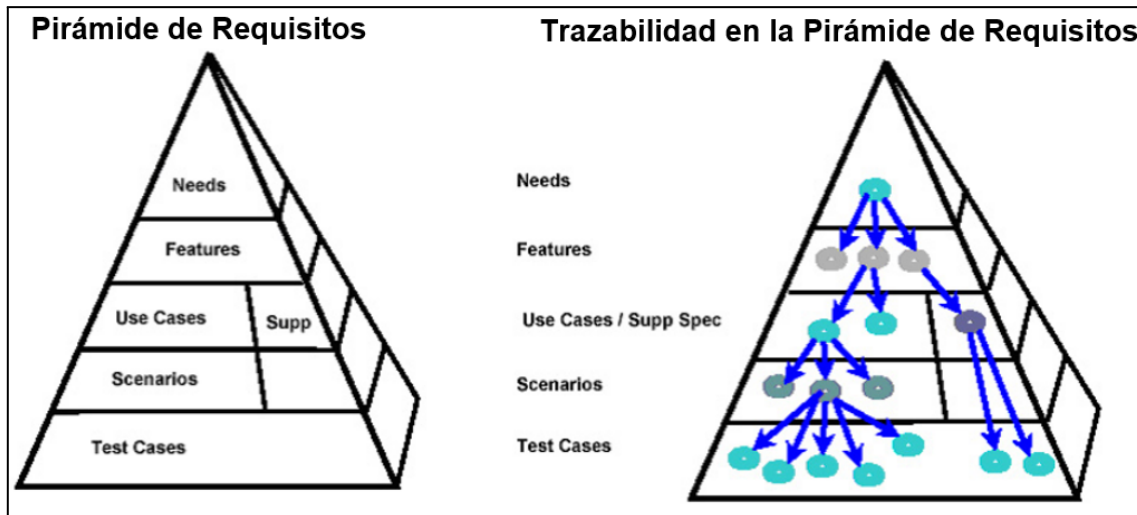


Figura 5: Fuente: Modelado de casos de uso [3]

A lo que hace referencia la Figura 5 es a la composición de una pirámide formada por una serie de elementos, los cuales son:

- **Needs:** Necesidades de usuarios o del negocio, es decir, los objetivos.
- **Features** (características): Funcionalidad de alto nivel que puede agrupar varias más específicas.
- **Use Cases** (casos de uso): Hacen referencia a los requisitos funcionales.
- **Supp Spec:** Requisitos No Funcionales o suplementarios.
- **Scenarios** (escenarios): Descripción de la funcionalidad de cada requisito (funcional o no funcional) o de casos de uso.
- **Test Cases** (test de caso de uso): Pruebas que se realizan para comprobar cada requisito funcional y no funcional.

Estos componentes dependen unos de otros, en los que de cada uno se puede extraer uno o más elementos de los demás. En el ejemplo de la Figura 5, de una necesidad se pueden obtener tres características, de cada una de éstas se consiguen dos casos de uso, que a su vez de estos se hallan una serie de escenarios y, finalmente, se logran test finales que se realizan para comprobar requisitos tanto funcionales como no funcionales. El

número específico de cada uno de los componentes variará según el problema y número de necesidades que se tenga por parte de una empresa o negocio.

4.3 Distribución de los requisitos

Una vez conocido de dónde surgen los requisitos, la distribución que realizaremos sobre los mismos, en base a la necesidad de este trabajo, es por medio de tablas, en las que estas se diferencian por el tipo de requisito y, a su vez, cada una contendrá tres elementos: **ID**, **descripción** y **prioridad**. Se realiza de esta forma con el objetivo de identificar cada característica a qué tipo de requisito pertenece. El **ID** simplemente es el identificador con el que detectar cada uno de los requisitos. La **descripción** hace indicar a qué se refiere cada requisito. Mientras que la **prioridad**, cuyos valores son alto, medio o bajo, nos hace ver cuán importante y necesario es su uso en nuestro sistema.

Para ello, comenzamos haciendo la distribución comentada anteriormente con los **requisitos funcionales**, diferenciándolos entre **dinámicos** y **estáticos** y, además, entre si son realizados por parte del usuario cliente o bien por medio de la empresa. Los requisitos estáticos son aquellos cuya información que contienen se deben de almacenar, como puede ser el caso de los datos personales de un cliente. Mientras que los dinámicos consisten en una serie de procesos que hacen evolucionar la información que contiene dicho requisito, como son las transacciones.

4.3.1 Requisitos funcionales dinámicos (Cliente)

Requisitos funcionales dinámicos de Cliente		
ID	Descripción	Prioridad
RFD_C1	Login/registro para el acceso a la plataforma para el usuario, mediante la introducción de su email y contraseña.	Alta
RFD_C2	Puede modificar campos correspondientes a su propia cuenta (nombre, apellidos, usuario, teléfono y contraseña).	Alta
RFD_C3	Recibe un correo electrónico con la confirmación de la cita solicitada en el negocio, servicio y por el empleado concreto elegido.	Media

RFD_C4	Mediante el apartado Contacto , el usuario cliente puede pedir asistencia a la empresa para solucionar un problema o realizar una sugerencia mediante un formulario que rellena el propio usuario.	Media
--------	---	-------

Tabla 1: Requisitos funcionales dinámicos de Cliente

4.3.2. Requisitos funcionales estáticos (Cliente)

Requisitos funcionales estáticos de Cliente		
ID	Descripción	Prioridad
RFE_C1	El usuario administra los campos de su propio perfil.	Alta
RFE_C2	Puede pedir cita, confirmar y/o anularla en cualquier negocio adherido a la aplicación.	Alta
RFE_C3	A la hora de pedir cita, puede rellenar los siguientes campos, eligiendo una de las opciones disponibles en cada uno: ubicación del negocio, servicio y empleado.	Alta
RFE_C4	En el catálogo de productos, es capaz de localizar los servicios ofertados en la plataforma por medio de un buscador y un listado.	Media

Tabla 2: Requisitos funcionales estáticos de Cliente

4.2.3 Requisitos funcionales dinámicos (Empresa)

Requisitos funcionales de Empresa		
ID	Descripción	Prioridad
RFD_E1	Añadirá ubicación, servicios y empleados que desempeñan las diferentes funciones del negocio.	Alta
RFD_E2	La empresa podrá cancelar o modificar citas de los usuarios registrados.	Media
RFD_E3	Podrán registrarse a través de un formulario diferente al de los clientes.	Alta

RFD_E4	Recibe un correo electrónico con la confirmación de la cita solicitada por un cliente en su respectivo negocio, saliendo reflejado en su horario de citas.	Media
RFD_E5	Cada negocio podrá definir sus propios horarios.	Alta
RFD_E6	La empresa proporciona su propio equipo de profesionales en un servicio.	Media

Tabla 3: Requisitos funcionales de Empresa

4.3.4 Requisitos no funcionales

La distribución que realizaremos sobre los **requisitos no funcionales**, es decir, asociado a las propiedades emergentes del sistema, es también por medio de tablas (al igual que en el caso anterior), en el cual se diferencian de la misma forma que antes, pero con los diferentes requisitos no funcionales, los cuales son: Rendimiento y eficiencia, disponibilidad, seguridad, fiabilidad y usabilidad.

- **Rendimiento y eficiencia:** Posibilidad de un funcionamiento óptimo de la aplicación en dispositivos de bajas prestaciones. Es imprescindible puesto que una de las intenciones con este trabajo es que la aplicación sea multiplataforma.

Requisitos no funcionales de rendimiento y eficiencia		
ID	Descripción	Prioridad
RNF_RE1	Funcionamiento óptimo de la aplicación en dispositivos de bajas prestaciones.	Alta

Tabla 4: Requisitos no funcionales de rendimiento y eficiencia

- **Requisito de disponibilidad:** Que se pueda pedir cita en cualquier momento del día actual, ya sea para el día en cuestión como para el próximo mes.

Requisitos no funcionales de disponibilidad		
ID	Descripción	Prioridad
RNF_D1	Pedir cita en cualquier momento del día (24/7/365).	Alta

RNF_D2	Soporte por parte de la empresa que asegura la correcta gestión del servicio que utiliza en la aplicación.	Alta
--------	--	------

Tabla 5: Requisitos no funcionales de disponibilidad

- **Requisito de seguridad:** A la hora de registrarse, se introducen una serie de datos personales para poder pedir cita con la intención de corroborar la identidad del usuario en cuestión. Estos mismos permanecerán seguros sin caer en manos de terceros, por lo que este requisito es fundamental para cualquier servicio web y/o móvil que requiera el registro de los clientes.

Requisitos no funcionales de seguridad		
ID	Descripción	Prioridad
RNF_S1	Datos personales de cada usuario seguros sin caer en manos de terceros.	Alta
RNF_S2	Copias de seguridad 1 vez al mes.	Alta
RNF_S3	Labores de administración protegidas contra el acceso no autorizado.	Alta
RNF_S3	Proteger el código de desarrollo de la aplicación en un servidor u otro dispositivo de uso privado.	Media
RNF_S4	Contraseñas encriptadas para todos salvo para el propio usuario en sí.	Baja

Tabla 6: Requisitos no funcionales de seguridad

- **Requisito de fiabilidad:** Para que las citas que se quieran pedir vía web y/o móvil se confirmen por cualquier dispositivo, enviando un SMS o mensaje por correo electrónico para confirmar y asegurar la cita de los usuarios.

Requisitos no funcionales de fiabilidad		
ID	Descripción	Prioridad
RNF_F1	Aplicación fiable a la hora de acceder, registrarse, confirmar citas solicitadas vía email o SMS, etc.	Alta

RNF_F2	Sistema fácil de mantener en cuanto a almacenamiento de datos, diseño de interfaz y código de desarrollo debido a la sencillez de la aplicación de cara al cliente.	Alta
--------	---	------

Tabla 7: Requisitos no funcionales de fiabilidad

- **Requisito de usabilidad:** Interfaz sencilla e intuitiva para todo tipo de usuarios (de cualquier franja de edad).

Requisitos no funcionales de usabilidad		
ID	Descripción	Prioridad
RNF_U1	Interfaz sencilla e intuitiva para todo tipo de usuarios.	Alta
RNF_U2	Debe funcionar, por medio de cualquier plataforma o sistema operativo, con cualquier navegador.	Alta

Tabla 8: Requisitos no funcionales de usabilidad

4.4 Diagramas de casos de uso

Es interesante añadir casos de uso para entender de manera visual los comportamientos de los participantes y del sistema o aplicación en sí. Se compone de una serie de **actores** que, en este caso, se podría entender que son el cliente, la empresa y el sistema. La función de estos diagramas es describir la funcionalidad del sistema tal y como se explica por medio de los requisitos explicados en el punto anterior.

Además de los actores, también se componen mediante un conjunto de círculos los casos de uso (requisitos funcionales) que realiza cada actor. Estos pueden ir por medio de <<include>> o <<extend>> (éste último por medio de líneas discontinuas).

Los casos de uso utilizados para este trabajo y por medio de una herramienta *online* llamada **Visual Paradigm Online**⁷ son los mostrados en las Figuras 6, 7 y 8:

⁷ <https://online.visual-paradigm.com/es/>

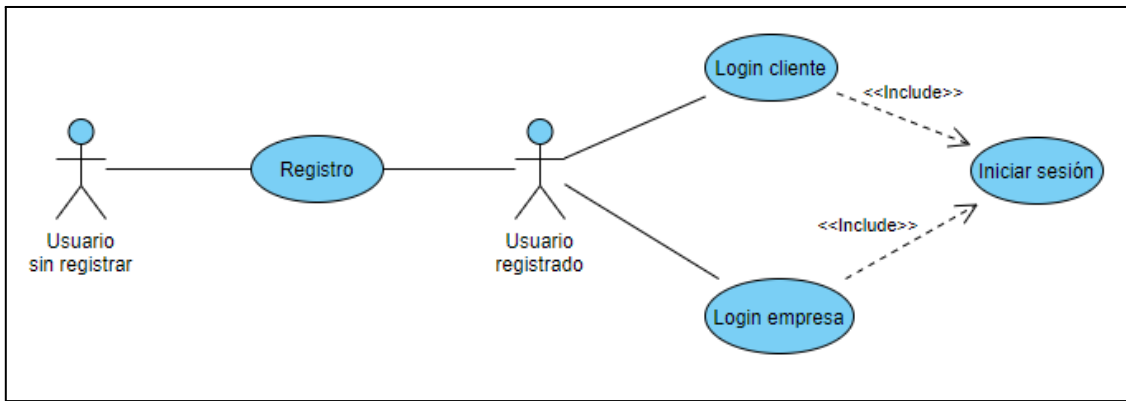


Figura 6: Caso de uso Inicio de sesión

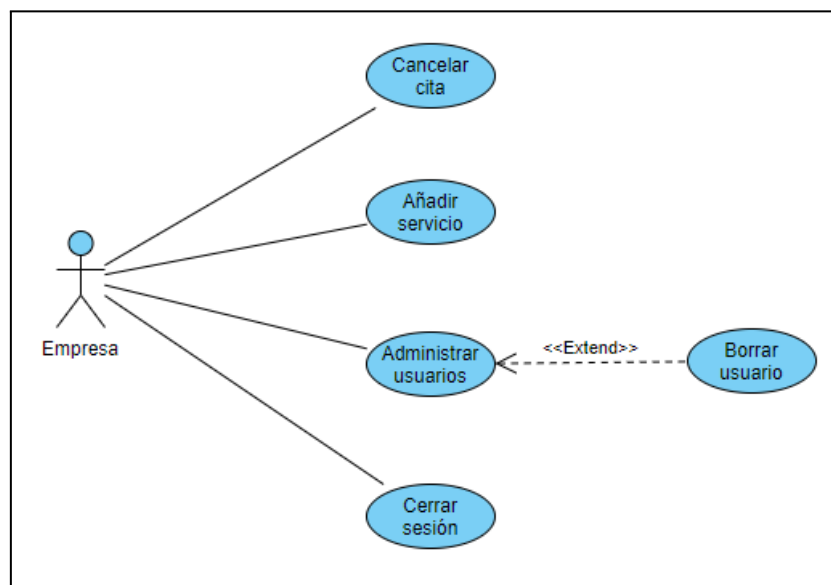


Figura 7: Caso de uso de Empresa

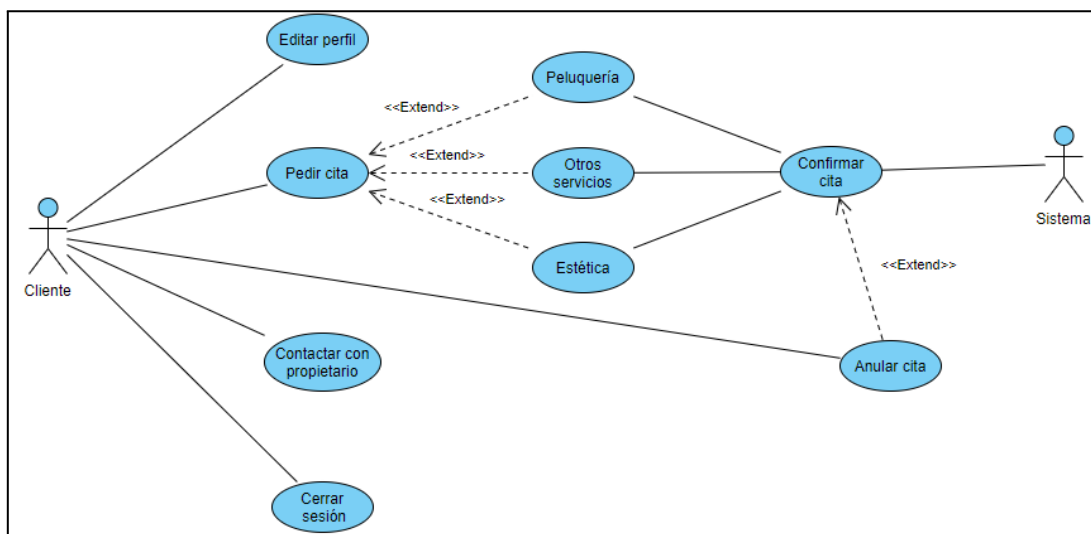


Figura 8: Caso de uso de Cliente

5. Estado de la cuestión

En este apartado se van a comentar algunas herramientas similares en el mercado [4] que se centran en el mismo tipo de negocio que la aplicación para este trabajo. Además, se explican de forma clara los objetivos 2 (Estudiar y proponer una posible solución de bajo coste para subsanar dichos problemas por medio de una aplicación) y 3 (Proponer un caso de uso y desarrollar una aplicación multiplataforma que permite la digitalización de un pequeño negocio a un coste asumible) de la memoria, ya que se tratan las distintas soluciones propuestas de bajo coste para solventar el problema de las empresas, cuál ha sido la escogida dando una explicación clara de ello y también se propone un caso de uso para luego desarrollar la aplicación, diferenciándolo de otras herramientas que se centran en el mismo nicho de mercado.

5.1 Diferencias en plataformas de bajo coste

Uno de los objetivos establecidos es la propuesta de una posible solución de bajo coste con la intención de poner remedio al problema marcado en el objetivo 1 (Analizar los problemas a los que se enfrentan los pequeños comercios, tales como el desconocimiento de la tecnología y el coste elevado tanto del desarrollo como del mantenimiento).

Esta aplicación se va a desarrollar con *WordPress*, pero antes se hará mención de otras plataformas similares de bajo coste que se han propuesto para solucionar los costes de mantenimiento web y la escasa tecnología por parte de los pequeños comercios. Además, se va a diferenciar las mismas con *WordPress* para explicar de una forma más clara cuáles son los detalles a tener en cuenta por el que se elige esta plataforma en lugar de otras.

Aquí podremos diferenciar qué detalles se tuvieron en cuenta para decidir *WordPress* con respecto a otras como *PrestaShop*⁸, *Joomla*⁹ o *Drupal*¹⁰.

⁸ <https://www.prestashop.com/es>

⁹ <https://www.joomla.org/>

¹⁰ <https://www.drupal.org/home>

Diferencias con *PrestaShop* [5]:

- Al no tratarse de la aplicación que se quiere desarrollar una tienda *online*, es preferible usar *WordPress*. La intención con esta plataforma es, además de establecer un sistema de citas, crear un catálogo de productos que serán los negocios adheridos por las empresas de forma que cada cliente pueda solamente verlos y valorar cada uno. Sin embargo, es preferible *PrestaShop* si se desarrolla una tienda *online* con productos propios y *WordPress* si es de afiliados.
- *WordPress* posee una gran cantidad de temas y *plugins* gratuitos de forma que la aplicación pueda ser plenamente funcional, mientras que con *PrestaShop* es posible crear una tienda *online* muy potente, pero con *plugins* y temas de pago.
- Con *PrestaShop* se puede crear una tienda más elaborada, pero para ello se necesita conocimientos avanzados de HTML, en cambio *WordPress* es ideal para todo tipo de usuarios.

Diferencias con *Joomla* [6]:

- *WordPress* contiene una interfaz más intuitiva y sencilla, además de un aspecto y funcionalidad mejores.
- *WordPress* tiene muchas más descargas, temas gratuitos y complementos a añadir a una página web que *Joomla*.
- *Joomla* es más flexible que *WordPress*, de forma que permite ser creativo con un proyecto, pudiendo personalizar el sistema sin tener que depender de extensiones.
- Panel de control más fácil de entender en *WordPress* que en *Joomla*, ya que las opciones que proporciona es sencillo saber qué función desempeña.
- A nivel de seguridad, *WordPress* ofrece, con un gran número de *plugins*, más posibilidades de prevenir ciberataques.

A diferencia de otro gestor de contenido como es *Drupal* [7], conlleva una serie de diferencias al respecto, las cuales son:

- A nivel de seguridad, *WordPress* es más segura que *Drupal* al contener una mayor cantidad de *plugins* de terceros y contener una mayor cuota de mercado, es decir, es más utilizada. Sin embargo, esto también implica que los ciberdelincuentes se centren más a la hora de realizar sus ataques, con lo que *WordPress* tiene mayor riesgo aunque con mejores prestaciones para corregir problemas por medio de parches de seguridad.
- A la hora de iniciar y gestionar una página web, *WordPress* es favorita debido a que su curva de aprendizaje es menor dado que contiene muchos temas listos para usarse. Además, contiene muchos más manuales e información a la hora de desarrollar una página.
- *WordPress* posee una serie de *plugins* y complementos añadidos con funciones agregadas, de forma que se puede construir una página con muchas funciones sin tocar mucho código. *Drupal* contiene menos *plugins* al ser bastante más limitado.
- *Drupal* en cambio es más ligero dado que requiere menos recursos en la web, conllevando a una carga rápida de la página mayor que con *WordPress* debido a las prestaciones que contiene, sobre todo si se abusa del número de *plugins* instalados.

Se puede decir que *WordPress* es ideal tanto para el mantenimiento de la web para pequeños comercios como para profesionales capaces de realizar un sistema de forma ágil, rápida y de bajo coste, dado los numerosos recursos e información que aporta para realizar una página web con diversas funcionalidades y adaptado a cualquier tema de forma gratuita.

Las características de *WordPress* que benefician para el desarrollo de la aplicación son las siguientes:

- La gestión de contenido es fácil de usar dado que no se necesita usar diseñadores web para realizar cambios en el sitio. Además, se puede crear menús personalizados y elementos de navegación sin tocar código.
- La gestión de usuarios es fácil y segura gracias al control de acceso al contenido. Además, el sistema de roles y capacidades de *WordPress* es extensible de forma

que permite controlar qué acciones están disponibles para los diferentes grupos de usuarios creados.

- Permite instalar muchísimos complementos gratuitos que están en el repositorio de *WordPress* de forma que permite ahorrar tanto tiempo como dinero para añadir funcionalidades a la página.
- Flexibilidad a la hora de agregar cualquier funcionalidad construida en PHP / *MySQL* de forma sencilla en una aplicación de *WordPress*.
- Actualizaciones de seguridad frecuentes de forma que se defiende ante los ataques de ciberdelincuentes.
- Tanto la plataforma como PHP, *MySQL* y los *plugins* de la plataforma son gratuitos.
- Uso de las *API REST* [8] que permiten el desarrollo de aplicaciones móviles, conllevando a que los usuarios consulten sus datos a través de una aplicación de escritorio o de móvil, un servicio externo y una aplicación web externa en lugar de realizar la consulta a través de un sitio web.
- Se trata de un *framework* ideal para redes multisitio de forma que si se ejecuta más de una instalación de *WordPress* con sitios web similares se puede compartir la misma base de datos y un conjunto de complementos y temas, conllevando a un ahorro de tiempo y dinero.
- Es capaz de desarrollar un comercio electrónico de forma que se integren múltiples pasarelas y formularios de pago seguros, además de información de pedido guardada y productos con precios. Todo ello se obtiene gracias a extensiones gratuitas como *WooCommerce*¹¹ y a otras de pago como *Paid Memberships Pro*¹².
- Permite impulsar aplicaciones en dispositivos móviles utilizando tecnología *WordPress*, como es el caso de los contenedores de aplicaciones para *iOS* y *Android*.
- Uso de la tecnología *AJAX (Asynchronous Javascript And Xml)*¹³ para, entre otras funciones, agregar contenido dinámico a un sitio. En *WordPress* se puede utilizar concretamente para enviar formularios de contacto sin recargar la página.

Como se puede observar, *WordPress* tiene muchas características que permiten definirla como un *framework* de desarrollo. Estas características se pueden ampliar, tal y como explica **O'REILLY** [8].

¹¹ <https://es.wordpress.org/plugins/woocommerce/>

¹² <https://wordpress.org/plugins/paid-memberships-pro/>

¹³ <https://developer.wordpress.org/plugins/javascript/ajax/>

Gracias a toda esta información, considero que *WordPress* ha evolucionado considerablemente desde sus inicios, llegando a ser ahora mismo un *framework* muy completo que permite el desarrollo de cualquier tipo de aplicación.

5.2 Basado en el caso de uso

Por qué decido utilizar WordPress por encima de otras plataformas

Si en la anterior sección se ha justificado la elección de *WordPress* como plataforma de desarrollo de bajo coste por delante de otras, aquí la intención es proponer un caso de uso y desarrollar una aplicación multiplataforma que permita la digitalización de un pequeño negocio a un coste asumible. De esta forma, podemos justificar la elección de una peluquería como caso de uso.

El desarrollo de la aplicación se realiza con la intención de gestionar un sistema de citas para clientes por parte de las diferentes empresas registradas. Con ello, tanto un cliente como la empresa que realiza una serie de servicios podrán establecer contacto directo a través de la aplicación.

Ahora se comenta algunas herramientas similares en el mercado centradas en el mismo tipo de negocio, así como la diferencia que hay de todas ellas con respecto a la aplicación desarrollada en este trabajo.

5.2.1 La Tijera

En primer lugar, hay que comentar la aplicación **La Tijera**¹⁴, la cual tiene un uso exclusivo para móvil, aunque vía web es interesante también debido a que enumera los pasos a seguir para utilizar la aplicación correctamente. Además, adjunta un enlace directo a *Google Play* y *Apple Store* para descargar por medio de dispositivos móviles con sistemas operativos *Android* y *iPhone* respectivamente.

¹⁴ <https://latijera.app/>

Comienza, siempre y cuando se active la ubicación del móvil del cliente, encontrando las peluquerías más cercanas y de mayor a menor valoración por parte de los usuarios registrados. Al elegir una, se puede observar la información básica del local escogido, como es el caso de una imagen del local, la ubicación vía Google Maps, el horario y los servicios que ofrecen, pudiéndose en este último ampliar y escoger el servicio en concreto, así como también realizar una valoración y escribir opiniones. En caso de ser del agrado del usuario, se puede realizar la reserva, haciendo previamente el registro a la aplicación, ya sea por medio de rellenar campos (nombre, email, contraseña y número de teléfono), como continuar con una cuenta de *Facebook*.

Acto seguido, se visualiza un calendario en el que el cliente puede indicar el día y hora a la que quiere pedir la cita. Entre otros aspectos, también puede elegir quién desea el cliente que sea el trabajador que le realice el servicio deseado.

Por otro lado, hay que tener en cuenta otros detalles de la aplicación, como es el caso de que, actualmente, los negocios dependen de un intermediario (administrador) que almacene los datos en la nube.

También comentar que, actualmente, esta aplicación únicamente serviría para aquellos usuarios que residen en la ciudad de Madrid, aunque se pueden adherir más negocios contactando previamente con el desarrollador de la aplicación.

En cuanto a diferencias con nuestra aplicación, la principal que presenta es que no se trata de una aplicación multiplataforma, que es en lo que principalmente se basa la de este trabajo. La forma de utilizarse es únicamente por medio de un dispositivo móvil, ya que por otro te redirige a un enlace por *Play Store* para dispositivos *Android* y por *Apple Store* para *iPhone*.

En la Figura 9 se puede apreciar la pantalla principal de esta aplicación con las funciones ya comentadas desde la perspectiva de un cliente.

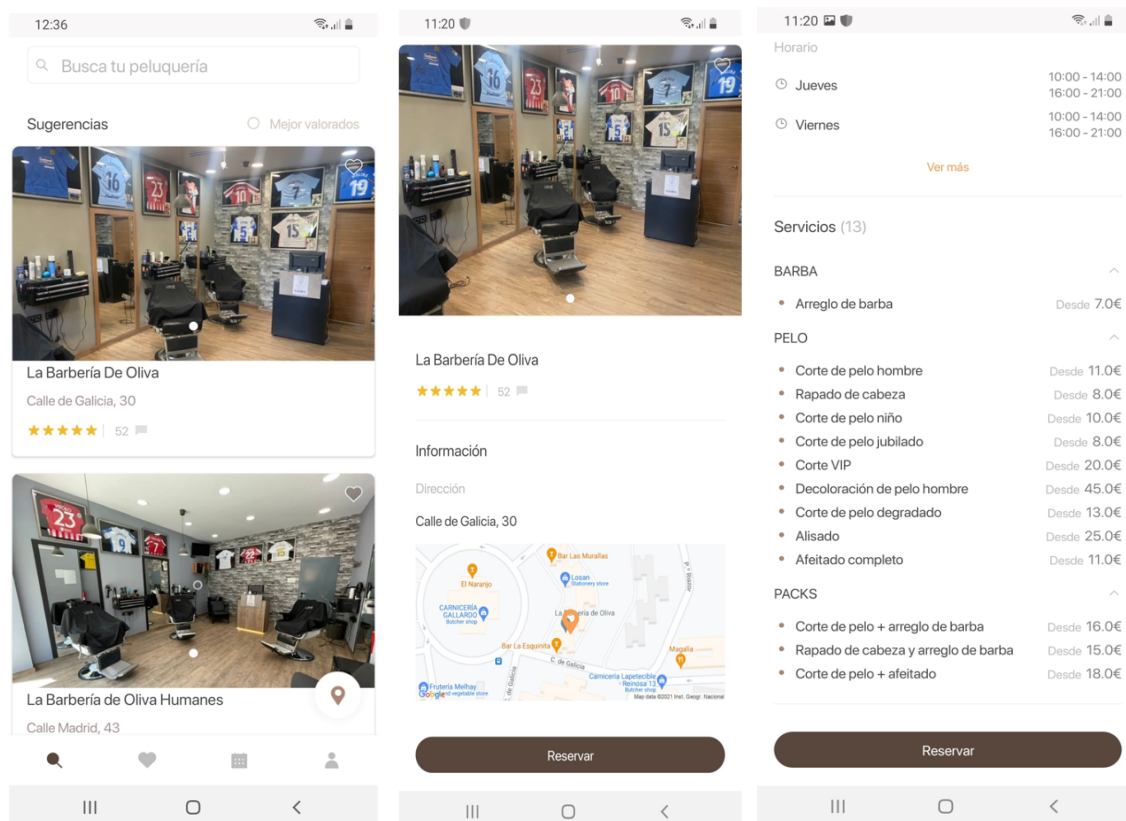


Figura 9: Pantalla principal de La Tijera y sus servicios

5.2.2 SalonBooking

Entre otras herramientas también se puede comentar **SalonBooking**¹⁵, cuya peculiaridad que tiene esta aplicación es que está desarrollada mediante *WordPress*. Esta contiene su cuenta de empresa y otras tantas que tendrán los clientes que se vayan registrando. En la cuenta de empresa de esta aplicación se podrá buscar por nombre o correo un cliente registrado y comprobar, durante el día actual o posteriores, la hora y servicio en los que ha concertado la cita. También se puede filtrar la búsqueda por servicio escogido.

Sin embargo, a diferencia de nuestra aplicación, esta herramienta sólo la podrán emplear completamente aquellos clientes registrados que se suscriban a un plan premium que ofrece, entre otras ventajas, la opción de utilizar la aplicación por medio de un dispositivo móvil. Por ello mismo, no todos los clientes registrados van a tener la posibilidad de sacar todo el potencial que tenga la aplicación. También lo que variará para la cuenta de

¹⁵ <https://www.salonbookingsystem.com/>

la empresa será la vista en la que la misma pueda ver las citas de los clientes que, aunque se emplee la misma herramienta de trabajo, no tiene porqué ser idéntica.

Podemos apreciar en la Figura 10 un ejemplo de cómo se visualizará el horario con las citas concertadas de los clientes de la aplicación desde la cuenta de la empresa.

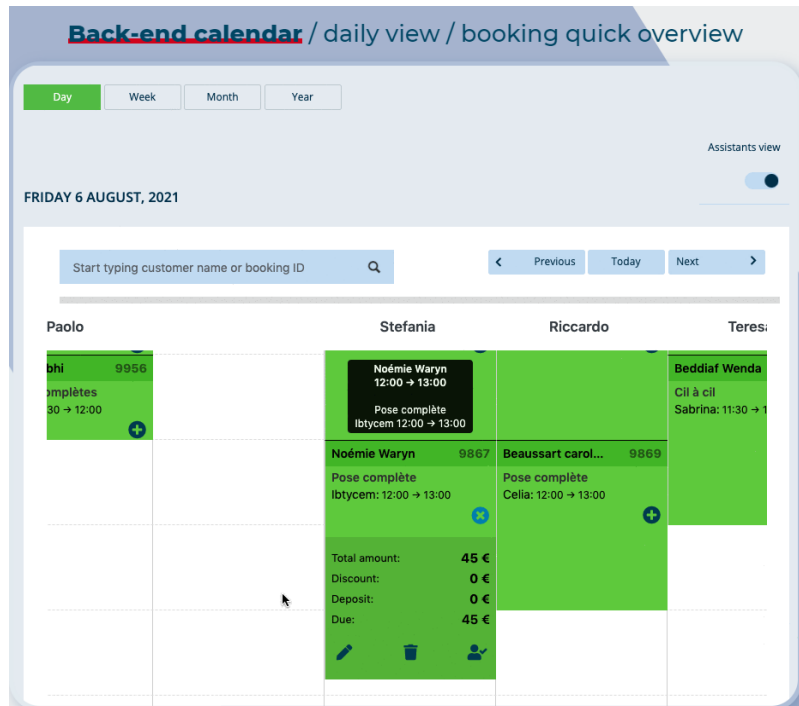


Figura 10: Horario de citas de SalonBooking

5.2.3 ZittApp

Y, por último, hay que comentar la aplicación **ZittApp**¹⁶, la cual es genérica porque trabaja con cualquier tipo de establecimiento como peluquerías, gimnasios, centros médicos, etc. Por otro lado, hay que comentar que se separa en dos partes: una parte web y otra móvil.

La primera consiste en un panel de control apto para los negocios, los cuales pueden gestionar los horarios de sus trabajadores, citas médicas, usuarios de gimnasio y peluquerías por medio del desarrollo de una API, así como también añadir nuevos servicios para cualquiera de estos negocios y, con todo ello, poder llevar a cabo un control dentro de su negocio.

¹⁶ <https://gausswebapp.com/category/ios>

Mientras que la segunda consiste en una aplicación móvil gratuita y disponible para *Android* y *iPhone* en la que los clientes pueden realizar sus respectivas reservas y, de antemano, registrarse en la app por medio de un usuario y una contraseña. Tiene funciones interesantes como filtrar la búsqueda de un comercio por negocio y nombre, añadirlos a la lista particular de favoritos o que las reservas se añaden automáticamente al calendario del dispositivo móvil del cliente, avisando al mismo 30 minutos antes del inicio del servicio.

Además, el negocio puede confirmar las reservas para evitar que estas sean ficticias y, según sea el servicio ofrecido, tienen una duración determinada. También ofrece el horario de cada trabajador y bloquea los días y horas fuera del horario laboral para evitar citas fuera de la franja horaria de trabajo.

En la Figura 11 se puede observar, desde el perfil de un cliente registrado en la aplicación, cómo funciona y qué funciones permite realizar desde un dispositivo móvil. Sin embargo, un inconveniente que puede presentar es que las empresas dependen de un intermediario, que es el que maneja la aplicación en lugar de gestionar ellos mismos sus trabajos y, en caso de caída de la aplicación por ejemplo, podría suponer un grave problema para muchas empresas y sus respectivos clientes.

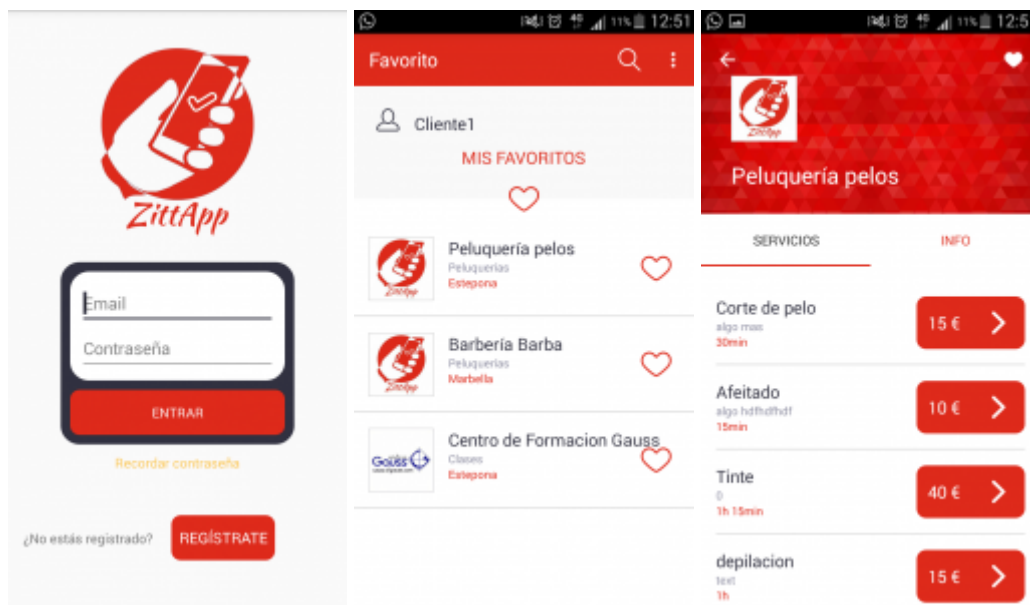


Figura 11: Horario de citas de ZittApp

5.2.4 Diferencias con la aplicación

El proyecto desarrollado como TFG ofrece características tanto novedosas como destacables para la gestión de peluquerías, estéticas, maquillaje, etc. con respecto a las comentadas anteriormente. El sistema es multiplataforma para todo tipo de usuario que se registre en el mismo. Además, destaca por una interfaz intuitiva, sencilla y muy vistosa tanto para todo tipo de usuario cliente que quiera concertar citas como para la empresa a la hora de realizar las gestiones pertinentes dentro de la aplicación. También proporciona un recordatorio vía email para utilizarse de forma que pueda avisar de la cita pedida por cada cliente.

Uno de los aspectos más importantes también es que, de cara al futuro y si la aplicación es un éxito, se pueden adherir a la misma pequeños negocios relacionados con peluquería, estética o incluso otros sectores. Se trataría de un aspecto positivo, ya que por sí mismos muchos comercios no dispondría de recursos suficientes para llevar a cabo un trabajo como este. Gracias al uso de la aplicación, estarían actualizados a los tiempos que corren y se darían a conocer por uno de los métodos más importantes para ello, que es disponer de los servicios de una aplicación multiplataforma, cubriendo de esta forma el problema de la confirmación de citas *online* por parte de los pequeños comercios.

6. Herramientas y tecnologías

En primer lugar, hay que saber cuál es el problema que queremos resolver, la propuesta de solución para abordar el mismo, un análisis de objetivos para cumplir en dicha solución, los pasos a seguir para llevar a cabo nuestro propósito, los requisitos que deben de ser implementados y un boceto o prototipo con el que podríamos apreciar el resultado final de nuestro proyecto. Una vez hecho esto, nos vamos a la pregunta de cómo y de qué manera aplicar todo lo anterior para cumplir con nuestro objetivo principal.

En este apartado se analizará la variedad de herramientas y tecnologías escogidas y utilizadas dentro del abanico que hay de todas ellas para la realización del trabajo. Todo ello engloba programas utilizados, así como también lenguajes de programación, interfaces de desarrollo o librerías entre otras. También se añade una sección de mockups, ya que proporciona, a nivel de boceto, una visión de lo que será la aplicación una vez se desarrolle e implemente.

6.1 *Mockups*

En primer lugar, una vez se sabe con exactitud el problema que se quiere solventar y su correspondiente solución, es necesario establecer un caso gráfico por medio de un boceto antes de comenzar a manejar *softwares*, *frameworks* y programar mediante código de diversos lenguajes de programación. Por ello, se diseñan una serie de bocetos o *mockups* para que, tanto el responsable de diseñar la aplicación como cualquier usuario de fuera pueda visualizar y hacerse una idea de cómo será la aplicación que se realizará posteriormente. Para ello se ha decidido utilizar la herramienta **Moqups** tal y como se comentó en el punto **Metodología**. Estos diseños están representando situaciones tales como el registro de un cliente y una empresa, la reserva de una cita, el calendario de reservas, acceso al catálogo de productos por parte del cliente, añadir y/o modificar información al negocio concreto por parte de la empresa y consultar miembros registrados a la aplicación (esto último por parte del administrador).

Para evitar añadir demasiadas capturas acerca de los bocetos realizados, se ha decidido adjuntarlas en un anexo y referenciarlas en este apartado.

6.2 Funcionamiento de la web en local

El próximo paso, es hacerse la pregunta de qué es lo que se necesita para abordar el problema propuesto con una solución eficaz en nuestra aplicación. Para llevar a cabo el objetivo de una aplicación óptima y multiplataforma que cumpla una serie de funcionalidades con la que se pueda solucionar el problema propuesto, es necesario lo siguiente:

- **Servidor web:** Al tratarse de una aplicación web multiplataforma, es obligatorio alojar nuestro contenido para que se pueda ver por medio de una página web o aplicación móvil.

Para este primer punto, se ha preferido partir de un servidor web local, debido a que no se quiere publicar de primeras al tratarse de una acción poco profesional debido a que se necesita garantizar una página fiable, eficaz y segura ante todo tipo de vulnerabilidades tanto internas como externas. Además, no hay que olvidarse de otros detalles como los errores de código y fallos en las distintas funcionalidades que contenga la web. Una vez ya se hayan hecho todas las pruebas pertinentes a nivel de seguridad, código y funcionalidades, ya podríamos migrar los datos a un servidor FTP o bien a un hosting.

- **Base de datos:** Indispensable si queremos construir una página web en la que se requieran usuarios tanto para clientes como empresas y el propio administrador. Para ello, se ha decidido emplear **MySQL**¹⁷ para la creación de la base de datos y las tablas pertinentes para el uso de *WordPress*.
- **Servidor de correo:** Una herramienta interesante a la hora de confirmar registros de nuevos usuarios por medio de un correo electrónico a aquellos que han rellenado un formulario de registro o bien uno de contacto de un usuario con el administrador de la aplicación. Para ello, es preciso hospedar la aplicación en un dominio web, ya que, según el servidor de correo que se utilice, se pide una serie de datos que desde el propio dominio se pueden obtener gracias a su panel de control. Además, se puede configurar para *WordPress* de dos formas igual de eficaces:

¹⁷ <https://www.mysql.com/>

- Un *plugin* que se instala de *WordPress*:
 - **WP Easy SMTP**¹⁸: *Plugin* sencillo e intuitivo con el que se puede configurar un servidor de correo rellenando los campos que te pide para llevarlo a cabo, como es el caso de: Nombre de dominio, host SMTP, SMTP *Secure*, puerto SMTP, la pregunta de si se requiere autenticación (siendo que sí), un nombre de usuario y contraseña. Además, como el anterior *plugin*, proporciona una opción para enviar correos de prueba con un mensaje predeterminado para comprobar que se envían los correos con la dirección asignada en la configuración del *plugin* que hace de servidor de correo.
 - **WP Mail SMTP**¹⁹: *Plugin* potente que permite configurar un servidor de correo. Es un poco más elaborado que el anterior, pero muy fiable. Se pueden rellenar los campos correspondientes para configurar el servidor de correo de dos maneras: Configuración manual (como el anterior) o mediante un asistente, el cual permite rellenar los campos necesarios mediante 5 pasos y con una interfaz sencilla.
- Un servidor de correo externo, como es el caso de **Mercury**²⁰ (incluido en el panel de control de **XAMPP**²¹).

Para albergar todos estos servicios, se ha decidido utilizar un paquete de software libre que cubre nuestras necesidades como la realización de todas las pruebas pertinentes en cuanto a funcionalidades, uso correcto de la base de datos, vistas web con diseño apropiado, protección de todas las vías de escape de seguridad, flexibilidad con los *plugins* de WordPress y gestión y administración en un panel de control, entre otras.

¹⁸ <https://wordpress.org/plugins/easy-wp-smtp/>

¹⁹ <https://wordpress.org/plugins/wp-mail-smtp/>

²⁰ https://e-mail.fandom.com/es/wiki/Mercury_Mail_Server#

²¹ <https://www.apachefriends.org/download.html>

Es cierto que hay una variedad de paquetes de software libre para elegir y poder configurar un servidor web en local, como es el caso de WAMP²², MAMP²³, LAMP²⁴, AMPPS²⁵ o incluso el uso de simplemente la herramienta Bitnami. Para decidir cuál utilizar finalmente, se ha hecho un análisis para diferenciar las diversas herramientas y poder comprobar cuál es más conveniente para el proyecto y, gracias a ello, se puede definir cada uno de la siguiente forma:

- **XAMPP**: Paquete de software libre muy conocido, multiplataforma y que incluye una serie de servicios eficaces para poder crear un servidor web interesante, como es el caso de un servidor web Apache, una base de datos *MySQL*, un servidor FTP como es FileZilla y un servidor de correo externo (*Mercury*).
- **AMPPS**: Paquete muy parecido a XAMPP aunque, sin embargo, no es el programa idóneo para nuestra aplicación de primeras, puesto que para utilizar WordPress (entre otras herramientas, programas, etc), se requiere de una cuenta premium, lo cual supone un coste que no todo tipo de usuario puede permitirse de primeras (y menos si se trata de una aplicación con el fin de usar para pruebas y no publicarse en el futuro).
- **WAMP**: Programa para utilizar como servidor web en local con uso exclusivo de equipos cuyo sistema operativo sea Windows. Por requerimientos podría ser una buena elección, pero al no manipular el software del todo y encontrar con un panel de control un poco limitado, se decidió cambiar de opción.
- **MAMP** y **LAMP**: Programas de igual uso que el anterior mencionado, aunque estos están limitados a los sistemas operativos Mac y Linux respectivamente.
- **Bitnami**²⁶: Biblioteca de paquetes de software para aplicaciones web que puede utilizarse como complemento adicional sin depender de un paquete de software libre como los anteriores, pero con diversas limitaciones, ya que no posee un panel de control para administrar funcionalidades como por ejemplo el servidor de correo y, además, tiene una proporción escasa de servicios para implementar una aplicación,

²² <http://wampserver.es/>

²³ <https://www.mamp.info/en/windows/>

²⁴ <https://www.onworks.net/es/software/linux/app-lamp-application-server>

²⁵ <https://www.ampps.com/>

²⁶ <https://bitnami.com/>

ya que únicamente cuenta con el servicio web de Apache y una base de datos MariaDB.

Finalmente y una vez concluidas las pruebas sobre los diferentes tipos de software para configurar un servidor web con una base de datos, se ha decidido partir de un paquete de software libre como es **XAMPP [9]**, el cual es muy conocido y proporciona una serie de servicios muy útiles para llevar a cabo una página web y acoplar con un **CMS [10]** (Sistema de Gestión de Contenido o **Content Management System**) como es *WordPress*. En la Figura 12 se puede ver cómo es el panel de control de XAMPP y cómo se activan y desactivan los servicios que tiene.

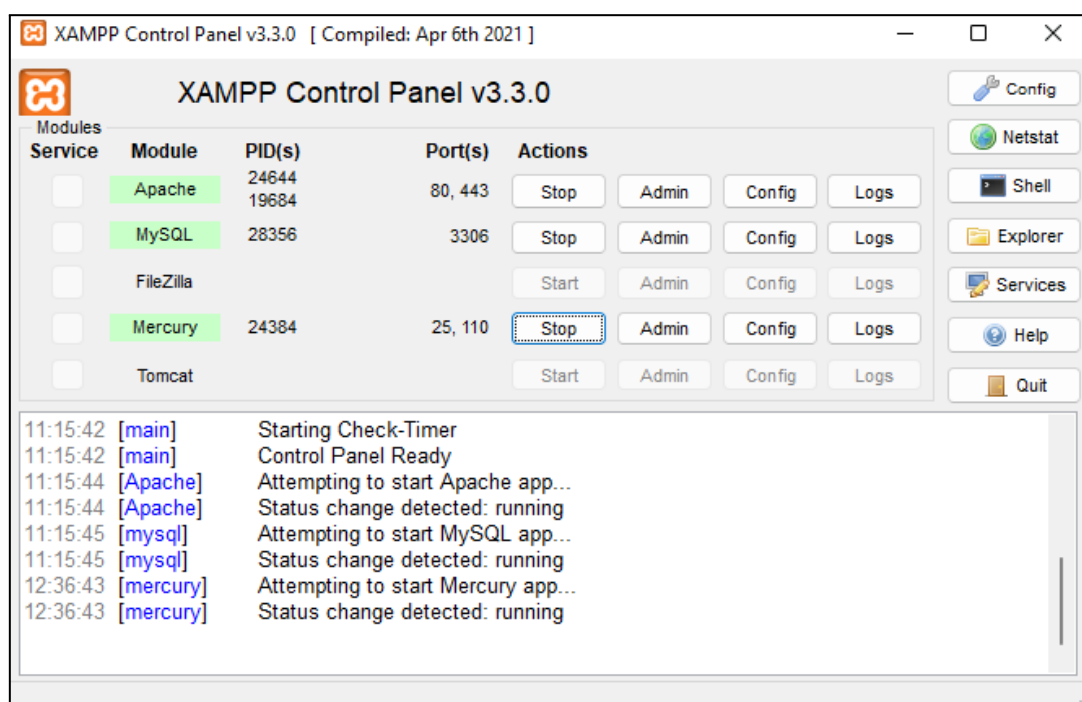


Figura 12: Panel de control de XAMPP con servicios corriendo

6.3 Uso de WordPress para la aplicación

Al final se ha decidido utilizar esta CMS (Sistema de Gestión de Contenido) con el propósito de crear una página web multiplataforma. Después de analizar *frameworks* como Laravel²⁷ u otros CMS como *Joomla* o *PrestaShop*, se ha decidido usar el mencionado

²⁷ <https://laravel.com/>

anteriormente, ya que es utilizado para muchas empresas con labores más allá de la creación de blogs, que era por lo que se conocía *WordPress* desde el principio. Sin embargo, la situación ha cambiado de forma notable gracias a sus nuevas actualizaciones, mayores funcionalidades, usos de diferentes *plugins* para realizar tareas en una web que no impliquen el uso de código, etc. También se ha decidido por el uso de este CMS por diversos motivos:

- Se programa, en su mayor parte, a nivel gráfico sin la necesidad de depender de un editor de texto (*Sublime Text*, *Visual Studio Code*, *Atom*, ...) que, pese a ser muy potente, no deja de ser un programa instalable y que, por consiguiente, ocupa más espacio en memoria. Además, dichos programas siguen sin ser gráficos.
- *WordPress* no es necesario instalarlo, ya que se puede acoplar, sobretodo si lo queremos para realizar pruebas con la aplicación, gracias al paquete de software libre XAMPP, ya que es tan simple como descargarse el CMS en un archivo comprimido y descomprimirlo en la ruta *xampp/htdocs*, tal y como se puede apreciar en la Figura 13. Con ello y habilitando los servicios de Apache y *MySQL* podemos instalar *WordPress* [11] en local, introduciendo un nombre para la base de datos que se va a crear y con su correspondiente usuario (*root* por defecto).

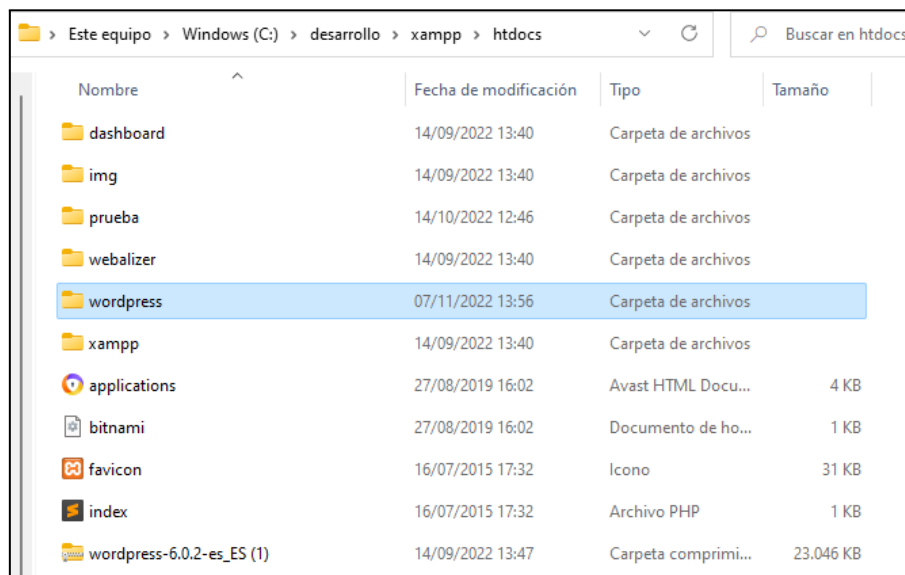


Figura 13: Ruta de archivo WordPress en XAMPP

- Muy utilizado en empresas, actualizado y con la flexibilidad de poder crear cualquier tipo de página web o aplicación multiplataforma.

6.4 Plugins de WordPress

Son extensiones que proporciona el propio CMS de *WordPress*, con las cuales es posible realizar una diversidad de funciones equivalentes a programarlas en un editor de texto. Para muchos casos, es muy cómodo utilizarlas dado que aportan funcionalidades en poco tiempo, ya que el proceso de instalación desde el panel de control de *WordPress* es sencillo y rápido.

A continuación se mostrarán los pasos para instalar los *plugins* en *WordPress* y, además, los que se han elegido para desarrollar la aplicación dentro de la gama proporcionada.

6.4.1 Proceso de instalación de un *plugin*

Para llevar a cabo esta labor, se puede desempeñar de una forma sencilla desde el panel de control de *WordPress*. Una vez se instala este *framework*, proporciona una serie de *plugins* por defecto que se pueden utilizar según el tipo de aplicación que se quiere desarrollar. Se pueden añadir nuevos *plugins* buscando dentro de la gama del propio CMS y, a través de un buscador, escribir para facilitar la búsqueda de los mismos. En la Figura 14 se puede ver tanto el panel de control de *WordPress* como la búsqueda de un *plugin*.

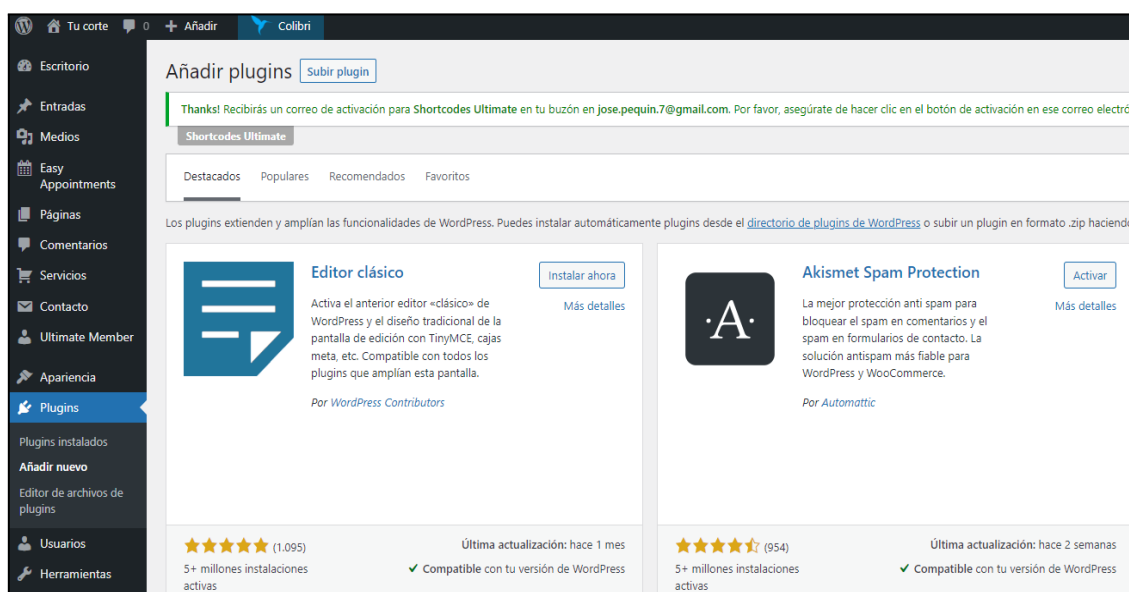


Figura 14: Búsqueda e instalación de un plugin

Una vez se localiza uno, es tan sencillo como darle a **Instalar** y, para que se pueda utilizar, **Activar** el mismo. Una vez se realiza dicha función, aparece una nueva opción en el panel de control de *WordPress* cuyo nombre suele ser el nombre del *plugin* u otro con el que se reconozca la funcionalidad que realiza. En las Figuras 15 se puede comprobar la diferencia entre un *plugin* instalado y uno sin instalar, además de verse reflejado el activado en el panel de control.

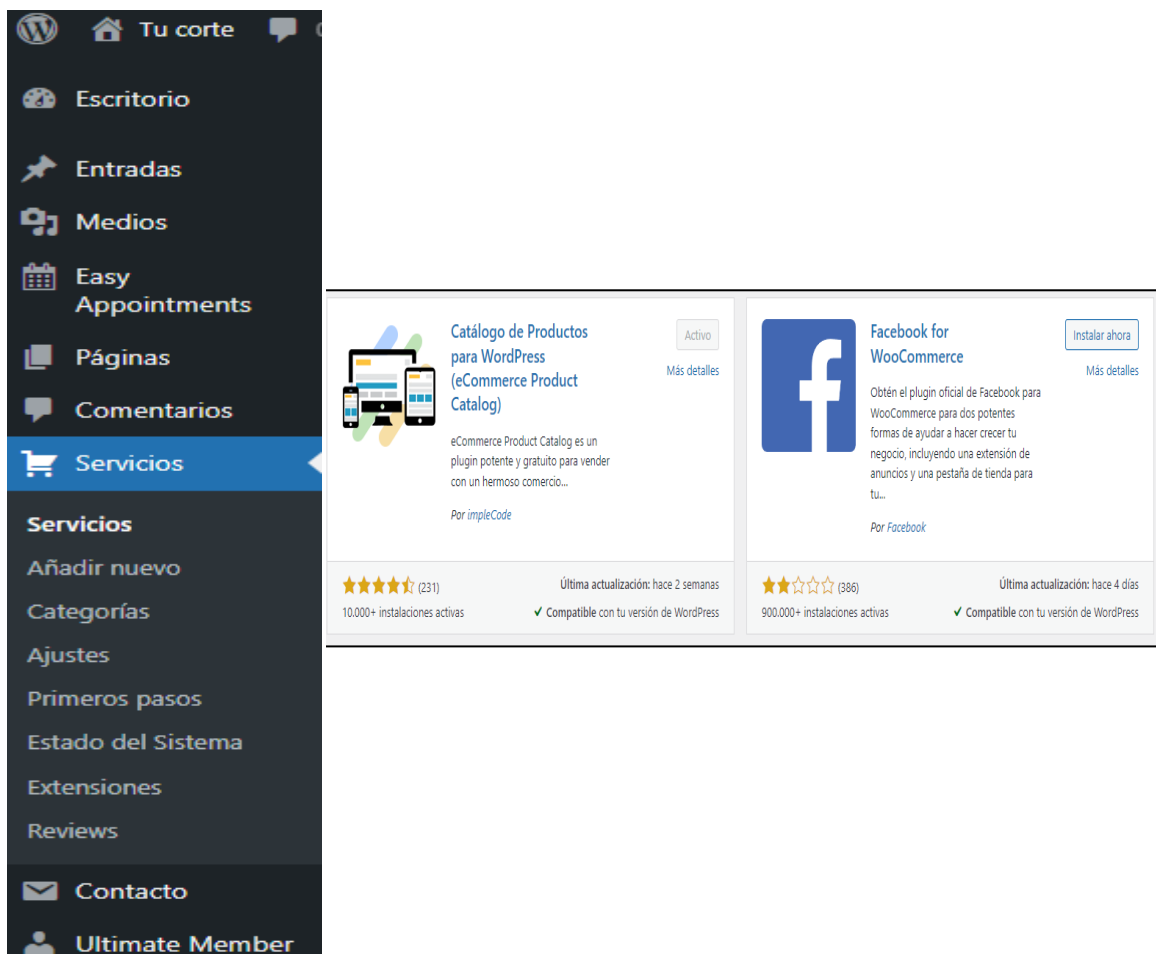


Figura 15: Plugin activado en el panel de control de WordPress

6.4.2 *Plugins* empleados en la aplicación

Después de tener en cuenta estos detalles y emplear un cierto tiempo en obtener información sobre las extensiones acerca del funcionamiento de cada *plugin* que se quiere para la aplicación, se ha decidido, por cada nivel de funcionalidad en la aplicación, emplear las siguientes extensiones para las determinadas actividades:

- Registro de usuarios e inicio de sesión
 - *Theme My Login*²⁸: Extensión que permite registrar a un usuario en la aplicación. Además, se genera una serie de URL o *slugs* con las acciones necesarias para realizar un registro de usuario, las cuales son: Inicio, *login*, salir de la sesión, registro, contraseña perdida y restablecer la contraseña.
 - *Ultimate Member*²⁹: Plantilla por defecto para administrar el perfil de un usuario, así como la vista correspondiente para iniciar sesión en un cuadro centrado que permite acceder por medio de nombre de usuario o correo con su correspondiente contraseña.

- Uso de código
 - *Simple CSS*³⁰: Extensión que se caracteriza por un marco en blanco debajo de cada página en el que se puede escribir código independiente para cada una en extensión CSS.
 - *Insert PHP Code Snippet*³¹: Permite crear nuevos archivos dentro de *WordPress* en el que se le puede aplicar código PHP sin necesidad de utilizar las etiquetas correspondientes para abrir y cerrar el formato. Estos archivos se pueden insertar en las páginas de la aplicación de *WordPress* que sean necesarias.

- Front Page
 - *Front Page Builder*³²: *Plugin* útil con el que se puede crear de forma sencilla una bonita e intuitiva página de inicio. Es flexible, ya que se puede usar de forma fácil para cualquier tema que contenga la web.

²⁸ <https://wordpress.org/plugins/theme-my-login/>

²⁹ <https://wordpress.org/plugins/ultimate-member/>

³⁰ <https://wordpress.org/plugins/simple-css/>

³¹ <https://wordpress.org/plugins/insert-php-code-snippet/>

³² <https://es.wordpress.org/plugins/front-page-builder/>

- *Child Theme Configurator*³³: Instalador que proporciona temas hijo a la plantilla de una página web cuando el uso del personalizador no es suficiente. Es interesante puesto que personaliza estilos, plantillas y funciones entre otras cosas. Además, a nivel de código en el archivo ***funcions.php***, si se quiere añadir y/o modificar funciones dentro del mismo y, por consiguiente, se quiere cambiar de plantilla en el futuro, esos cambios se van a aplicar siempre y cuando se apliquen las funciones sobre el tema hijo. En caso de no ser así, esos cambios en el archivo se habrán borrado.
- Roles de usuarios
 - *Nav Menu Roles*³⁴ [12]: Permite asignar, para cada tipo o rol de usuario, quiénes pueden ejercer diversas funciones dentro de la aplicación, así como crear nuevos roles y renombrar los anteriores.
- Shortcodes
 - *Shortcodes Ultimate*³⁵ [13]: Colección de elementos funcionales y visuales, de forma que hace más sencillo y vistoso al usuario que lo va a ejecutar. Contiene una gama de más de 50 *shortcodes*. Es adaptable a cualquier tema, con un diseño moderno y adaptable y completamente amigable al usuario.
- Copias de seguridad o *backup* en base de datos
 - *Duplicator*³⁶ [14]: Extensión que puede copiar el contenido de la base de datos y la web en sí de *WordPress* con la intención de volcarlo en una web distinta (función que se realizará más adelante).
 - *WP-DBManager*³⁷: Alternativa a utilizar *phpMyAdmin*, ya que se puede realizar consultas en SQL, así como observar las bases de datos y tablas que se encuentran creadas y, además, realizar copias de seguridad.
 - *All-in-One WP Migration*³⁸: Extensión cuya funcionalidad es exclusivamente realizar copias de seguridad de la base de datos. Estas mismas se pueden

³³ <https://wordpress.org/plugins/child-theme-configurator/>

³⁴ <https://wordpress.org/plugins/nav-menu-roles/>

³⁵ <https://wordpress.org/plugins/shortcodes-ultimate/>

³⁶ <https://wordpress.org/plugins/duplicator/>

³⁷ <https://wordpress.org/plugins/wp-dbmanager/>

³⁸ <https://wordpress.org/plugins/all-in-one-wp-migration/>

descargar en el escritorio y quedan reflejadas en una lista por defecto en *WordPress*.

- Catálogo de productos
 - *eCommerce Product Catalog*³⁹ [15] para *WordPress*: *Plugin* elegante y gratuito para el *ecommerce* de *WordPress* que permite crear y personalizar un catálogo de productos simple y, a su vez, efectivo. Posee distintas funcionalidades como:
 - Habilitar un carrito de compra para cada cliente, aunque en este caso se va a desactivar, ya que el propósito de esta aplicación es un catálogo simple de productos para que cada cliente pueda ver los servicios incluidos en la web.
 - Opción para filtrar productos por nombre de producto/negocio, categoría o subcategoría entre otras opciones.
 - Crear y modificar tantos artículos como hagan falta.
 - Importar/Exportar negocios por medio de una plantilla CSV rellenando los cinco campos por defecto que se requieren. Son 5 en lugar de 6 ya que no se va a visualizar el precio al ser negocios y no productos “suelos”.
 - Añadir, por medio de la extensión ***Reviews Plus - Google compatible WordPress Reviews for any content***⁴⁰, un apartado para publicar reseñas, con sus respectivos comentarios y valoraciones de cada cliente a los diferentes negocios publicados en el catálogo.

- Calendario de reservas
 - *Easy Appointments*⁴¹ [16] : Sistema de gestión simple con el que se pueden realizar de forma rápida y sencilla tanto citas como reservas. Contiene varias particularidades al respecto de otros calendarios de reservas, como son los siguientes casos:
 - Calendarios de reservas particulares para cada negocio.
 - Se puede establecer una conexión negocio-servicio-trabajador.

³⁹ <https://wordpress.org/plugins/ecommerce-product-catalog/>

⁴⁰ <https://wordpress.org/plugins/reviews-plus/>

⁴¹ <https://easy-appointments.net/>

- Se pueden añadir más trabajadores para un determinado servicio y/o negocio.
 - Permite enviar correos de reserva, a modo de recordatorio, tanto al cliente que la solicita como al trabajador del negocio que la recibe. Dentro del propio correo, también es posible tanto confirmar la reserva como cancelarla a través del enlace que se proporciona.
 - Se puede personalizar una ventana emergente con los contenidos de cada cita para obtener una información más detallada de la cita de cada cliente.
- Servidor de correo
 - *Easy WP SMTP*
 - *WP Mail SMTP*
 - Otras funcionalidades
 - *Increase Maximum Upload File Size*⁴²: Extensión de *WordPress* que permite modificar el tamaño de subida de todo tipo de archivos a la web. Es muy interesante, puesto que se puede utilizar para importar copias de seguridad o bien subir archivos comprimidos de mucha capacidad.

En este punto se ha decidido mencionar las funcionalidades, ya que su desarrollo como tal se explicará en el siguiente punto, es decir, en la **Arquitectura del sistema**, ya que se realiza una explicación profunda de la aplicación en general y todas sus funcionalidades.

6.5 Tecnologías web

En este apartado se comentarán las tecnologías web empleadas para desarrollar la aplicación por medio de *WordPress*. Se utilizan, según para que funcione, los lenguajes de **HTML** (*HyperText Markup Language*), **CSS** (*Cascading Style Sheets*) y **PHP**. Estos lenguajes se emplean con los distintos fines:

- **HTML**

⁴² <https://wordpress.org/plugins/wp-maximum-upload-file-size/>

- Crear páginas a las que podrá acceder cualquier usuario (ya esté logueado o no).
 - Diseñar formularios tanto de registro por parte de los usuarios clientes y administradores como de contacto.
 - Creación de tablas para almacenar los datos personales de cada usuario en sus respectivos perfiles.
 - Tablas en las que se podrá visualizar los servicios disponibles en la aplicación en los que el usuario cliente quiera pedir una cita.
 - Uso de distintas capas por medio de la etiqueta `<div id>` con el fin de darle un toque estético y estructurado para organizar las diferentes páginas.
- **CSS**
 - Aplicar estilo a las diferentes etiquetas que se encuentran en las páginas creadas en *WordPress* con formato HTML, utilizando las diferentes formas para ello:
 - Código CSS en el archivo ***style.css***, que es un fichero creado por defecto en *WordPress*.
 - Uso de un *plugin* de *WordPress* llamado ***Simple CSS*** que permite personalizar el código de estilo para cada página de la aplicación.
 - Uso de determinadas etiquetas de código html con estilo css de plantillas *Bootstrap* (al igual que el *header*, *footer* y algunas vistas).
 - **PHP**
 - Aplicar funciones para el correcto funcionamiento de la aplicación en *WordPress*. La gran mayoría vienen ya implementadas, pero se ha decidido hacer uso de:
 - El fichero ***functions.php*** para verificar las funcionalidades de la aplicación y, entre otras, añadir un tema hijo para añadir una plantilla

de *WordPress* en lugar de usar el tema padre. Es importante usar hijos, ya que si se implementan nuevas funciones y se cambia de tema padre, estas funciones creadas manualmente se eliminan, cosa que no pasa con los temas hijos.

- Uso del *plugin XYZ PHP Code* que sirve para “crear” *snippets* (fragmentos de código) con extensión PHP para aplicarlos en las diversas páginas de la aplicación. Es interesante utilizarlo, ya que emplea código PHP sin necesidad de abrir y cerrar con las etiquetas `<php?` y `?>` respectivamente.
- Código suelto en diferentes vistas (páginas) para aportar datos en tablas dinámicas con las que se pretende que cada usuario, independientemente del rol que asuma, sea capaz de ver su propia información personal, la cual es la que se otorga una vez se registra en el formulario.

7. Arquitectura del sistema

En este punto se va a ver un diagrama con la arquitectura del programa, en el cual se podrá ver cuáles son los recursos utilizados para desarrollar la aplicación multiplataforma en *WordPress*, teniendo en cuenta los siguientes aspectos:

- Se podrá ver los roles que tendrá el usuario según las acciones que desempeñe una vez acceda a la aplicación.
- Se comprueba la conexión establecida del usuario con el servidor web a través de un dispositivo en concreto y un navegador web.
- Muestra los servicios, tanto web, como de base de datos y correo, que contiene el servidor al que quiere conectarse el usuario.

La arquitectura del sistema se muestra en la Figura 16 de esta forma.

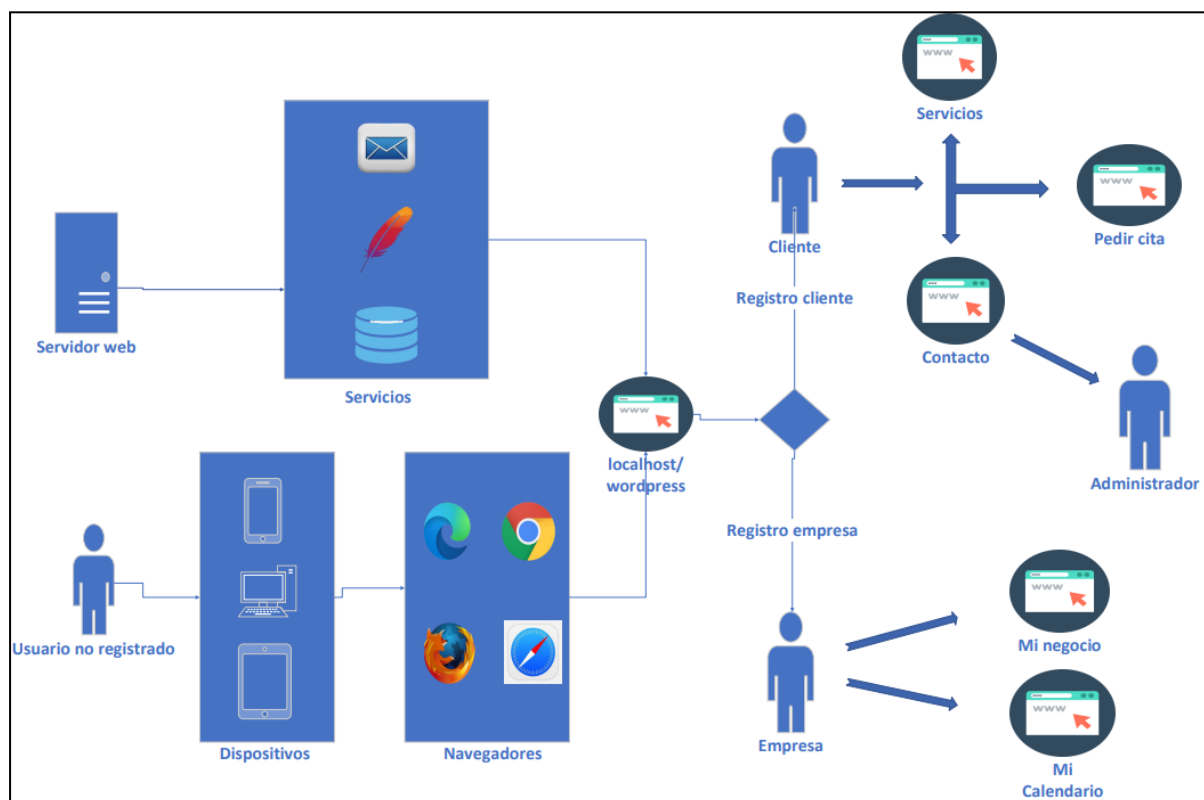


Figura 16: Arquitectura del sistema

Podemos ver cómo el usuario, sin estar registrado, accede por medio de un dispositivo y con un navegador cualquiera a una página web para conectarse a un servidor web, el cual proporciona una serie de servicios (web apache, base de datos *MySQL* y de correo). Se puede comprobar que, tanto los navegadores como los servicios utilizados, aparecen dentro de cajas independientes indicando que para las labores que toca, se pueden utilizar. Acto seguido, si todo ha sido un éxito, accede a la página de inicio de la aplicación. Para el caso de dispositivos móviles y tabletas, también es posible acceder escaneando un código QR. Ya dentro de la página, este usuario se registra como cliente o empresa rellenando su correspondiente formulario y, según el rol asignado, tendrá la capacidad de desempeñar diversas funciones tales como, siendo cliente, la de pedir citas y consultar negocios. Mientras que si se trata de una empresa es capaz de consultar citas de clientes y añadir información sobre el producto en el catálogo web. Dichas funcionalidades son tratadas en la arquitectura del sistema con una imagen en forma de página web, indicando de esta manera que cada acción se desempeña en una vista diferente dentro de la aplicación.

8. Funcionalidades del sistema

En este punto se va a explicar el funcionamiento de la aplicación realizada por medio de *WordPress* y las herramientas y tecnologías comentadas en el punto anterior, la arquitectura del programa y, además, se realizarán algunas pruebas para corroborar el correcto funcionamiento de la aplicación, viendo así de qué manera cubre las necesidades básicas para abordar el problema propuesto al principio de la documentación.

8.1 Acceso a la aplicación multiplataforma

Para conocer su funcionamiento de principio a fin, se va a explicar de primeras cómo se accede a la web por medio de la plataforma que el cliente desee:

- Si se quiere acceder vía ordenador de escritorio o portátil, independientemente del sistema operativo existente, simplemente se tiene que teclear la dirección web en el navegador que el usuario desee para acceder a la página de inicio de la aplicación.
- Si el dispositivo empleado se trata de un móvil o una tableta, es posible realizar la primera acción y, además, escaneando el código QR⁴³ correspondiente de la web. Este se encuentra tanto en papel físico como ahora en la documentación.

Una vez se acceda a la página de inicio desde cualquier dispositivo, por defecto será desde un usuario no registrado. En función del rol que se quiera desempeñar, se realizará un registro como **cliente** o bien como **empresa** y, a continuación, realizar sus funcionalidades.

8.2 Roles de usuario

Aquí se explica con más detalle el funcionamiento completo de la aplicación desde todos los roles existentes.

⁴³ <https://login.qr-code-generator.com/>

Usuario no registrado

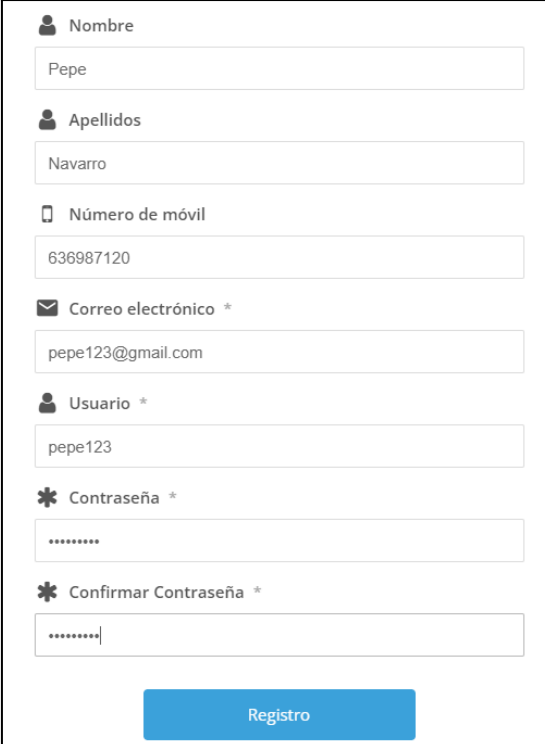
Simplemente puede acceder a la página de inicio, ya que, tanto para pedir cita a uno de los establecimientos como para ver los mismos en el catálogo de productos, tendrá que registrarse, ya sea como cliente o como empresa, en función de las necesidades que tenga ese usuario en concreto.

Cliente

Una vez se cumplimenta el formulario de registro de cliente, se redirige de forma automática a la página de inicio con el nuevo rol asignado. De esta forma, puede realizar las siguientes acciones:

- Rellena datos básicos en el formulario y le permite navegar por toda la aplicación con permisos de lectura.
- Es capaz de consultar todos los negocios adheridos a la aplicación, apoyándose de un buscador por nombre de negocio o bien filtrando por servicios que ofrece cada establecimiento. Dentro de cada negocio, también se puede enviar una valoración con una breve descripción sobre su experiencia.
- Puede enviar un mensaje al administrador de la aplicación gracias a una página de contacto, de forma que recibirá una respuesta en su dirección de correo.
- Puede pedir cita en alguno de los establecimientos de la aplicación, especificando el nombre del negocio, servicio, fecha y hora para la misma.

La Figura 17 muestra el formulario que tiene que rellenar un cliente para registrarse.



Formulario de registro para clientes. El formulario contiene los siguientes campos:

- Nombre: Pepe
- Apellidos: Navarro
- Número de móvil: 636987120
- Correo electrónico *: pepe123@gmail.com
- Usuario *: pepe123
- Contraseña *: [oculto]
- Confirmar Contraseña *: [oculto]

Botón de Registro

Figura 17: Formulario de registro para clientes

Empresa

Su forma de registro es idéntica al del rol de cliente, pero rellenando un formulario diferente. En cambio, las acciones que puede desempeñar son las siguientes:

- Incluye el nombre de la empresa en la información de su perfil.
- Ajeno al formulario de registro de empresa pero dentro de la misma vista, puede acceder a un enlace que se encuentra debajo del formulario para contactar con el administrador de la aplicación por correo electrónico.
- Puede añadir su negocio en el catálogo de productos de *Ecommerce de WordPress* de tres formas:
 - Manualmente, rellenando el nombre de los campos que se requiere: URL, nombre, categorías, descripción corta y descripción larga.
 - Importando el contenido por medio de un fichero CSV.
 - Pidiéndolo al administrador por medio de un correo electrónico.
- Tiene un acceso directo a su producto para editarlo de una forma ágil o bien consultar las valoraciones que realizan los clientes.

- En su cuenta, puede consultar su respectivo calendario de reservas en una página, visualizando en franjas de colores las celdas en caso de que un cliente haya pedido cita para ese establecimiento. Cada color representa un servicio llevado por un trabajador en concreto dentro del mismo negocio.

A diferencia de los clientes, las empresas rellenan el formulario que puede verse en la Figura 18 para registrarse en la aplicación.



Formulario de registro para empresas. El formulario contiene los siguientes campos:

- Nombre: Julián
- Apellidos: Ruiz
- Número de móvil: 685478532
- Dirección de correo electrónico *: julian123@gmail.com
- Empresa * ? : Peluquería Julián
- Nombre de usuario *: julian123
- Contraseña *: [oculto]
- Confirmar la contraseña *: [oculto]

Botón de Registro (azul) y botón de Enviar correo (naranja). Texto de enlace: Si deseas más información, contacta vía email con nosotros.

Figura 18: Formulario de registro para empresas

Administrador

Es el encargado del mantenimiento de la aplicación y eliminación de usuarios (si lo requiere). Tanto usuarios con rol de cliente como el de empresa pueden contactar con él por

medio de una página de contacto. Además, desde el rol de empresa, se puede enviar un correo electrónico con el *email* y el asunto establecidos por defecto.

8.3 Funcionamiento después de registro

Una vez se ha realizado el correspondiente registro de cliente o de empresa, se redirige al mismo a la página de inicio y se da por logueado automáticamente. Para llevar un control de los usuarios registrados, se puede comprobar en la Figura 19 el formulario de miembros que proporciona la *plugin Ultimate Member* explicado anteriormente.

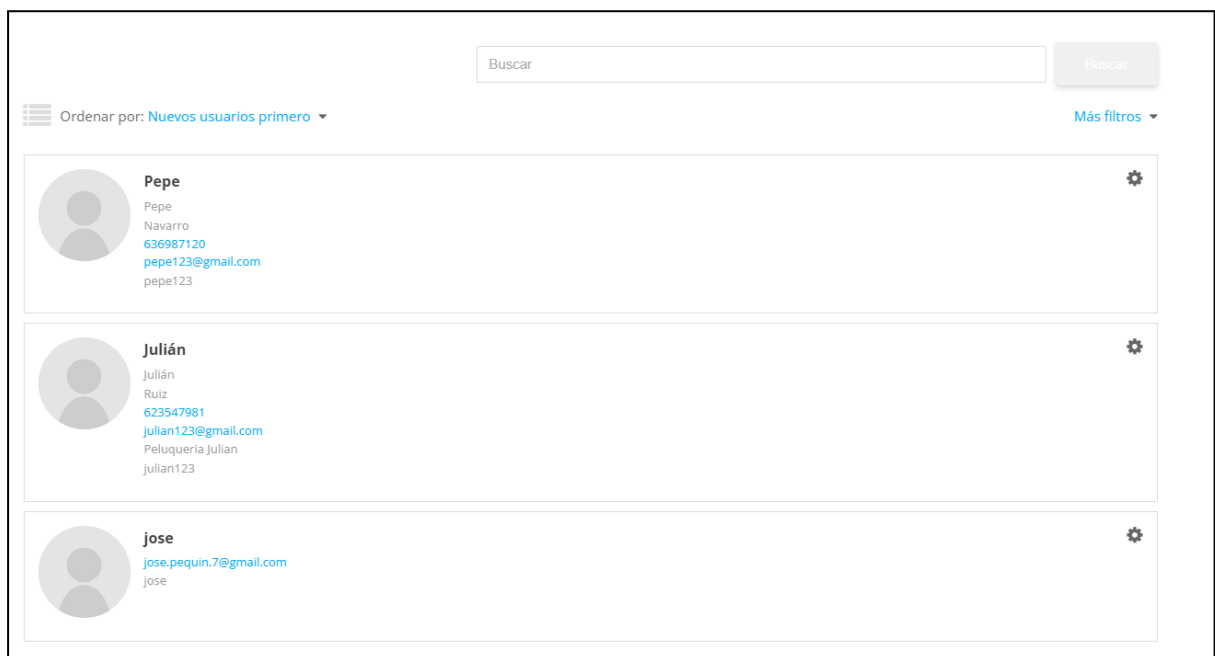


Figura 19: Listado de miembros registrados en la aplicación

Se puede ver los usuarios registrados en la aplicación, siendo Pepe un cliente, Julián una empresa y Jose el único administrador.

Según el rol asignado, las opciones del menú variarán debido a la visibilidad de campos, ya que un usuario cliente puede realizar acciones como el acceso al catálogo de productos o solicitar una cita entre otras. Mientras que el que se registre como empresa podrá hacer tareas como editar el negocio subido al catálogo de productos o visualizar el calendario de reservas para comprobar las citas que han sido registradas, con su fecha, hora y servicio solicitado por cada cliente.

En la Figura 20 y Figura 21 podemos notar la diferencia que hay del menú de un cliente con respecto al de una empresa.



Figura 20: Menú de un usuario registrado con rol de cliente



Figura 21: Menú de un usuario registrado con rol de empresa

Si un usuario (ya sea cliente como empresa) quiere cambiar la contraseña o se le ha olvidado, puede cambiarla clicando en el enlace que tiene debajo del formulario para iniciar sesión. Además, también es capaz de rellenar el formulario de contacto con el administrador, consultar la pestaña Nosotros, ver las políticas de privacidad y editar su perfil, independientemente del rol asignado.

8.4 Funcionalidades en el usuario cliente

Vamos a comenzar por las funcionalidades que puede desempeñar un usuario cliente.

Catálogo de productos Ecommerce con rol de cliente

Un usuario cliente puede ver el contenido de un catálogo de productos almacenado en la aplicación. En ella están ubicados los negocios adheridos por la aplicación, perteneciendo cada uno de ellos a un usuario empresa que se haya registrado y haya subido su producto (negocio) a la misma. El cliente, para mayor comodidad, puede buscar el negocio que desee escribiendo el nombre del mismo o bien filtrando por los servicios que realice cada uno. En la Figura 22 se puede apreciar el acceso de un cliente al catálogo de productos y su forma de buscar el deseado.



Figura 22: Catálogo de productos

Una vez selecciona el negocio apropiado, puede ver una serie de campos que proporciona información del servicio en sí, como es el caso de la ubicación, teléfono, correo electrónico, horario y servicios. Además, proporciona una descripción sobre el negocio donde se puede apreciar, de forma más detallada, los servicios con los que trabaja dicho establecimiento. Dentro de esa misma página, hay un apartado para que el usuario cliente registrado deje una valoración con un comentario sobre ese negocio o alguno de sus servicios, tal y como se muestra en la Figura 23.

Review Peluquería Julián.

Conectado como julian123. [Edita tu perfil.](#) ¿Salir? Los campos obligatorios están marcados con *

Your Rating ☆☆☆☆☆

Título de tu opinión

Tu Opinión *

[ENVIAR OPINIÓN](#)

Figura 23: Valoración de un usuario a un establecimiento del catálogo

Calendario de reservas con el *plugin Easy Appointment* con rol de cliente

El calendario de reservas está realizado por medio de un *plugin* de *WordPress* que hace muchísimo más sencillo y ameno el funcionamiento de un sistema de citas. Por parte del cliente, simplemente tiene que acceder al apartado **Citas** que se encuentra en su menú principal y le redirigirá a la página. Una vez dentro de ella, le aparece un formulario para solicitar información sobre la cita, como es el caso de:

- **Ubicación:** El nombre del establecimiento.
- **Servicio:** Dependiendo del negocio, se elige el servicio que quiere el cliente (peluquería, estética, maquillaje, mechas, tatuajes, ...).
- **Trabajador:** En función del negocio y servicio escogidos, se podrá elegir al trabajador que se encargue de dicha labor.

Los tres campos contienen una lista desplegable en la que salen los respectivos nombres asociados, es decir, un trabajador que pertenezca a ese negocio en concreto y se encargue de un servicio de esa empresa.

Una vez rellenados los campos, aparece el calendario cuyas franjas horarias por día pueden ser libres (color azul de fondo) u ocupadas (color blanco de fondo). Después de elegir la hora que mejor convenga, aparece un recuadro con la información de la cita y los datos básicos del cliente y, a continuación, un botón para confirmar la cita. Una vez se pulse el botón, se dará por confirmada la cita, llegando un correo electrónico tanto al cliente que ha pedido la cita como al trabajador de la empresa que va a realizar el servicio elegido.

Se puede comprobar en la Figura 24 cómo un cliente rellena los campos necesarios para completar una reserva.

Ubicación

Servicio

Trabajador

enero 2023

L	M	X	J	V	S	D
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

09:00	09:30	10:00	10:30
11:00	11:30	12:00	12:30
13:00	13:30	14:00	14:30
15:00	15:30	16:00	16:30
17:00	17:30	18:00	18:30
19:00	19:30	20:00	20:30

Información personal

Los campos con * son obligatorios

Nombre *

Teléfono *

Correo electrónico *

Descripción

Resumen de la reserva

Por favor revisa los detalles de tu cita a continuación y confirma:

Location **Peluquería Julián**

Service **Peluquería**

Worker **Julián**

Fecha y hora **17 de enero de 2023 11:30**

Figura 24: Reserva de cliente a establecimiento

8.5 Funcionalidades en el usuario empresa

Una vez terminadas las funcionalidades del usuario cliente, se va a comenzar por explicar las funcionalidades que puede desempeñar un usuario empresa.

Catálogo de productos Ecommerce con rol de empresa

Al igual que un cliente puede ver los establecimientos adheridos en la aplicación, los usuarios con rol de empresa pueden ver sus respectivos productos (negocios) desde la pestaña **Mi servicio** en el menú principal, con el fin de añadir y/o modificar alguno de los apartados una vez está publicada la empresa en la aplicación. También, desde esa misma vista, podrá comprobar las reseñas publicadas por los usuarios cliente sobre su producto. Si resulta que el usuario de empresa no ha introducido aún su producto en la aplicación, podrá realizarlo de tres formas:

- Crearlo desde la aplicación manualmente.
- Solicitar al administrador por medio de la página de contacto o vía *email* que le añada el producto, especificando la información que necesita.
- Importando un archivo CSV la información que solicita relleno los 5 campos por defecto que necesita.

Lógicamente, también podrá eliminar su producto del catálogo cuando lo vea necesario.

En la Figura 25 se podrá apreciar mejor cómo un usuario de empresa añade su establecimiento al catálogo.

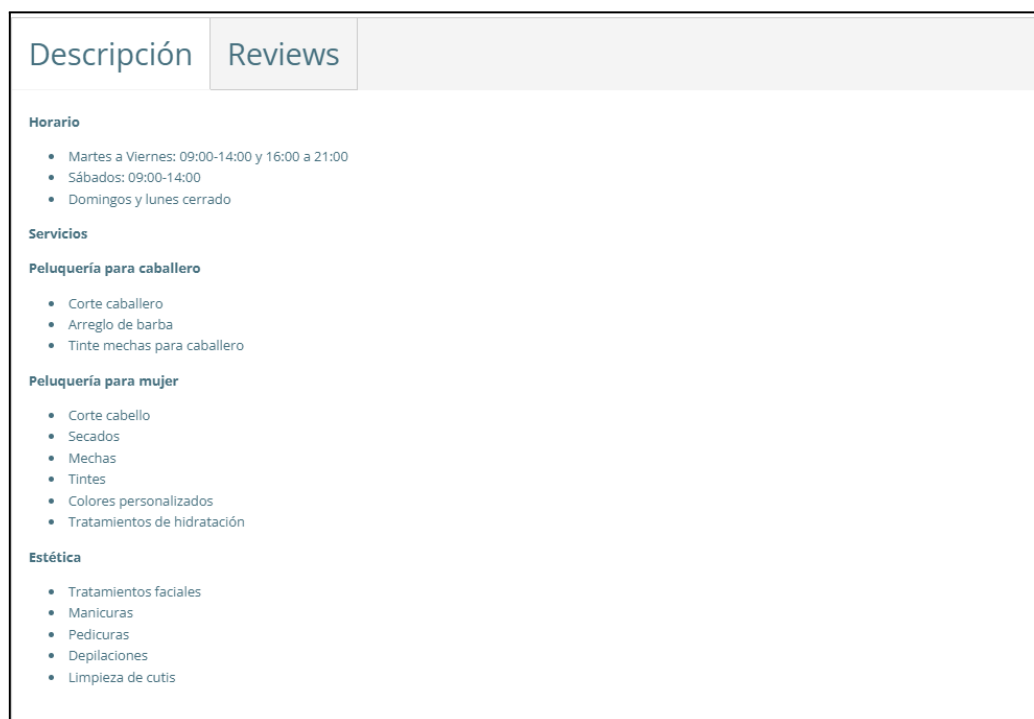


Figura 25: Descripción de producto de empresa

Calendario de reservas con *plugin Easy Appointment* con rol de empresa

Comentando el *plugin* necesario para la reserva de citas, si el cliente únicamente tiene que realizar la cita, el usuario empresa tiene más labores con este mecanismo. Para que el usuario cliente pueda realizar la reserva, la empresa necesita tener registrado los campos siguientes: **Ubicación**, **Servicios** y **Trabajadores**. En cada campo se requiere la siguiente información:

- Ubicación
 - Nombre: Nombre de la empresa.
 - Dirección: Ubicación del establecimiento.
 - Localidad: Ciudad o pueblo en el que se encuentra el local donde se ejerce.
- Servicios
 - Nombre: Nombre del servicio.
 - Duración: Tiempo en minutos en el que se ejerce el servicio/cliente.
 - Paso de ranura: Horario entre celda y celda para pedir la cita.
 - Bloquear antes de: Se puede bloquear el servicio x minutos antes de la cita.
 - Bloquear después de: Se puede bloquear el servicio x minutos después de la cita.
 - Límite de reservas/día: Permite establecer un límite de reservas de un servicio por día. Al estar a 0 quiere decir que no hay ningún límite.
 - Precio: Importe establecido al servicio. En la aplicación está deshabilitado, con la intención de que los clientes lo puedan ver ya sea en el catálogo de productos o bien en los establecimientos.
 - Color servicio: Permite asignar el color de una franja en concreto por servicio, de forma que el usuario empresa sepa distinguir bien los servicios en el calendario de citas.
- Trabajadores
 - Nombre: Nombre del trabajador perteneciente al establecimiento.
 - Descripción: Funciones o servicios que desempeña en el negocio.
 - Dirección de correo
 - Teléfono

Una vez se consigue los datos correspondientes por cada usuario empresa, se pueden conectar, de forma que un cliente pueda reconocer los servicios y trabajadores

pertenecientes a un negocio en concreto. Además, permite rellenar más campos como son los siguientes:

- Número de ranuras: Indica el número de reservas/cliente que permite cada celda que se vaya a ocupar.
- Días de trabajo: Indica los días en los que ese establecimiento está abierto.
- Rango de días de trabajo: Indica hasta qué día está abierto el establecimiento. Normalmente no se sabe una fecha exacta de cierre, pero dichos datos se pueden actualizar.
- Tiempo de trabajo por día: Define el horario de apertura y cierre de un negocio en concreto.
- Opción de si actualmente ese negocio está activo.

Después de configurar el calendario de reservas para cada empresa, en la Figura 26 se puede ver cómo gestiona las citas que le llegan de los clientes registrados, poniéndolas en un calendario.

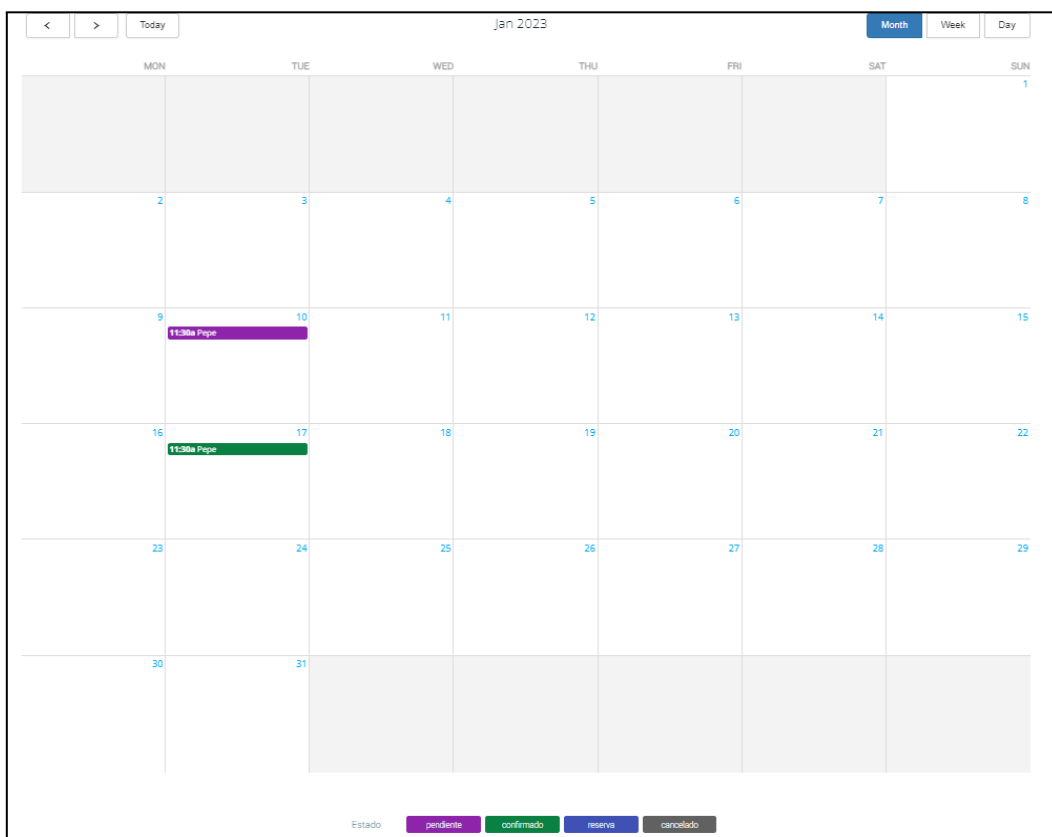


Figura 26: Calendario de reservas de establecimiento

8.6 Ajustes importantes sobre el Calendario de citas con el *plugin Easy Appointment*

En este punto se van a comentar algunos ajustes que son interesantes de añadir al calendario de reservas por parte de los usuarios empresa.

En aspectos generales, es posible realizar una reserva automática, de forma que se autorrellena los campos restantes una vez que el usuario cliente elija fecha y hora para la cita, además del estado predeterminado que permite saber si esa cita está pendiente o se confirma. En este caso se dejará directamente como **confirmada**. También es posible limitar el número de citas por cliente, teniendo hasta 3 citas para realizar. Estos detalles se pueden apreciar en la Figura 27.

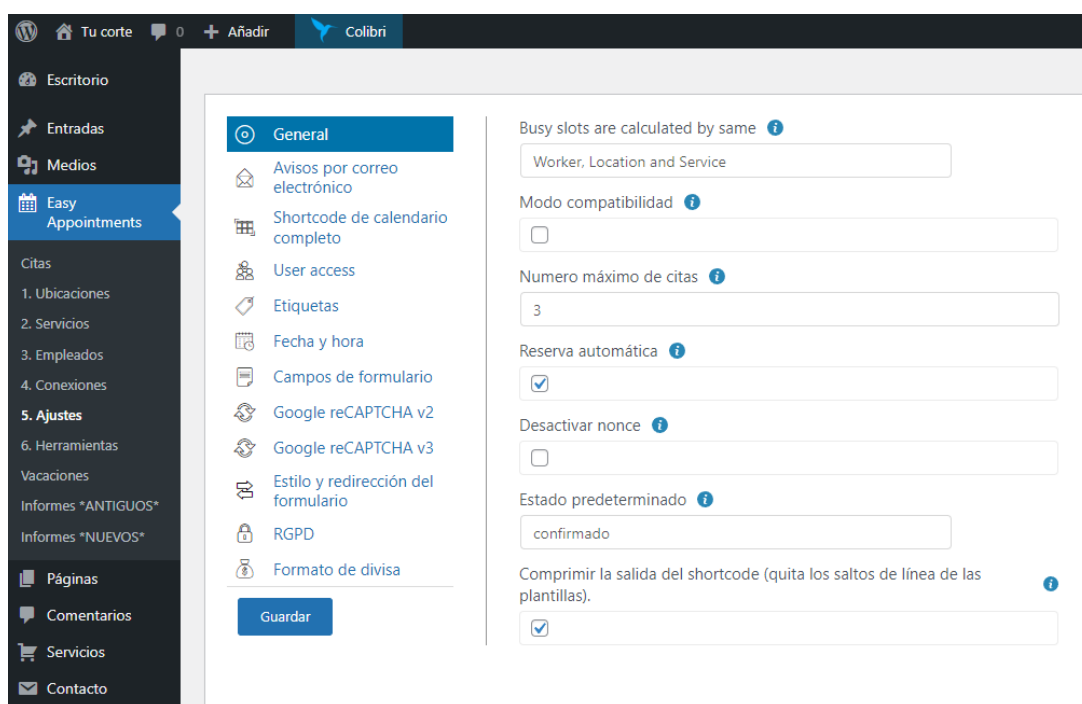


Figura 27: Ajustes generales del sistema de reservas de WordPress

En cuanto a los avisos por correo electrónico, se puede añadir el cuerpo de un mensaje y su asunto de forma personalizada, así como establecer la opción de que dicho aviso lo reciba por correo tanto el cliente que realiza la reserva como el trabajador que cumple con el servicio escogido.

Se puede personalizar el contenido del evento o reserva elegidos dentro del calendario. Se ha escogido que salgan los campos que muestra la Figura 28, obteniendo contenido de forma dinámica según los datos que contenga la cita establecida.

Los datos de la cita son los siguientes:

- **Fecha:** {= event.date}
- **Hora inicio cita:**{= event.start}
- **Hora fin cita:**{= event.end}
- **Dirección:**{= event.location_name}
- **Nombre del trabajador:**{= event.worker_name}
- **Confirmar reserva:** {= link_confirm}
- **Cancelar reserva:** {= link_cancel}

Figura 28: Datos de la cita en ventana emergente del calendario de reservas

Otras funcionalidades que se pueden editar son:

- Personalizar los formatos de fecha y hora.
- Dar acceso a los usuarios empresa para editar cambios.
- Añadir y/o modificar los campos del formulario que rellena un cliente para pedir cita.

9. Dificultades afrontadas y experiencias durante el desarrollo

Durante el transcurso del trabajo, se han encontrado una serie de dificultades que han supuesto más o menos tiempo en solventar y que es ideal comentarlas para que, tanto en las ampliaciones futuras sobre este trabajo como para otros usuarios que quieran desarrollar un trabajo similar, los tengan en cuenta.

También se explica la experiencia a nivel personal sobre el uso de *WordPress*, así como también el proceso de elección de los *plugins* instalados para la aplicación, la valoración de este *framework* y si el resultado final se corresponde con lo que se planteaba al principio del proyecto. Con todo ello, se cubrirá el objetivo 5 (Evaluar *WordPress* como *framework* de diseño de aplicaciones multiplataforma) del proyecto.

9.1 Problemas encontrados

En el siguiente punto se explicarán de manera detallada los problemas que han ocasionado demoras a lo largo del trabajo tales como: Error de conexión con la base de datos de *WordPress*, archivos corruptos o congelados de la base de datos, cambio de paquete de software libre WAMPP a XAMPP debido a que la instalación de *WordPress* requería el plan *premium*, fallo en la caché de la aplicación, etc.

Para trabajar en una aplicación web, es necesario tener el control de los servicios para habilitar, así como un acceso directo a los ficheros de configuración por si hay que realizar una serie de modificaciones en lugar de realizar una búsqueda del mismo en la ruta donde se encuentren. Para ello, es preciso tener un panel de control, el cual se otorga de forma automática si se migra el contenido web en un dominio o bien, si se decide trabajar en local como en este caso, partir de un paquete de software libre que albergue dichas funcionalidades. Por tanto, era el momento oportuno de realizar la búsqueda e instalación de un programa que proporcione su propio panel de control.

Por otro lado, hay que tener en cuenta que el usuario y contraseña proporcionados tiene que ser el mismo tanto para la instalación de la base de datos de *WordPress* como para el acceso a *phpMyAdmin*.

Una vez se ha decidido utilizar XAMPP como paquete de software libre, por el momento iba todo en orden, con *WordPress* instalado en local por medio de una carpeta comprimida introducida en la ruta correspondiente para su correcta instalación y su adecuado funcionamiento vía web. Sin embargo, una vez transcurrido un cierto tiempo, surgió un error al activar el servicio *MySQL* desde el panel de control de XAMPP.

A medida que se iba avanzando en la aplicación, se encontró otro problema, tratándose de un error en la función `wp_cache_get`, el cual no permite iniciar ni la página web ni el panel de control de *WordPress*.

9.2 Soluciones planteadas

Para los diferentes problemas encontrados a lo largo del trabajo, se ha decidido llevar a cabo soluciones distintas para seguir hacia adelante con el trabajo y, posteriormente, anotar los problemas y soluciones planteadas para que en el futuro si vuelve a darse, volver a emplear las mismas soluciones. Además, también puede servir de ayuda a otros usuarios que pretendan llevar a cabo un trabajo similar a éste o en el que se empleen las mismas herramientas. El autor del TFG ha decidido resumir en una tabla las relaciones **Problema encontrado - Solución planteada** en el orden en el que han ido surgiendo los mismos a lo largo del desarrollo del trabajo. Posteriormente, se explican con mayor nivel de detalle. Los problemas encontrados y soluciones planteadas son las siguientes:

ID	Problema encontrado	Solución planteada
1	Falta de un software con panel de control para instalar un servidor web local con una base de datos y utilizar para pruebas, además de incluir <i>WordPress</i> de forma gratuita	<ul style="list-style-type: none"> - Instalar Bitnami. - Instalar WAMPP con los servicios de Apache y <i>MySQL</i>. - Instalar XAMPP con los servicios de Apache, <i>MySQL</i> y Mercury.
2	Acceso denegado a <i>phpMyAdmin</i> desde el usuario <i>root</i> .	<ul style="list-style-type: none"> - Crear antes la base de datos <i>WordPress</i> y, desde el fichero

		<i>config.ini.php</i> , añadir la contraseña que se quiere poner al usuario.
3	Archivos corruptos en la base de datos de <i>WordPress</i> , impidiendo el funcionamiento de la aplicación en el servidor local.	<ul style="list-style-type: none"> - Cambiar el número de puerto del servicio <i>MySQL</i>. - Copia del contenido de la carpeta <i>data</i>, a excepción de las carpetas <i>information_schema</i>, <i>mysql</i> y <i>phpMyAdmin</i>, además del fichero <i>ibdata1</i>
4	Error en la función <i>wp_cache_get</i> que se encuentra en la carpeta <i>wp_includes</i> y en el fichero <i>option.php</i> .	<ul style="list-style-type: none"> - Utilizar el <i>plugin All-in-One WP Migration</i>. - Borrar contenido de las carpetas <i>wp_content</i>, <i>wp_admin</i> y <i>wp_includes</i>. - Buscar copia de seguridad que se realizó unos días antes y volcarla en un nuevo servidor web utilizando el <i>plugin Duplicator</i> de <i>WordPress</i>.

Tabla 9: Problemas encontrados y soluciones planteadas a lo largo del desarrollo

Software con panel de control

Al principio se decidió probar Bitnami, ya que ocupaba poco espacio en memoria y tiene el material suficiente para construir una aplicación sencilla con una interfaz web y su correspondiente base de datos. Sin embargo, en vista de no poseer un panel de control para administrar los usuarios y las bases de datos, la hacía vulnerable y poco flexible a nivel administrativo. Los problemas que acarreaban eran tales como la inaccesibilidad a la base

de datos por medio de **phpMyAdmin** por problemas con el usuario por defecto (**root**), así como el fallo al utilizar un servidor de correo al requerir un host SMTP.

Por otro lado, se probó con el paquete de software libre WAMPP, el cual es gratuito y posee su propio panel de control y, desde el mismo, es sencillo habilitar y deshabilitar los servicios web de Apache y de base de datos de *MySQL*. Sin embargo, para instalar *WordPress* desde el mismo, se necesitaba instalar la versión *premium* y, al no tratarse de una aplicación con fines comerciales, no era necesario utilizar esta solución.

Finalmente, se decidió instalar otro paquete de software libre que es, sin duda, el más conocido y utilizado, es decir, XAMPP. Una vez instalado, se ha decidido descargar *WordPress* para usar en local por medio de una carpeta comprimida, introduciendo la misma en la ruta **xampp/htdocs** para su correcta instalación y su adecuado funcionamiento vía web.

Sin embargo, una vez transcurrido un cierto tiempo, surgió un error al activar el servicio *MySQL* desde el panel de control de XAMPP.

Acceso denegado a phpMyAdmin

Cuando se instala *WordPress* en local y se escribe el nombre de la base de datos que va a albergar todo el contenido del *framework*, se necesita unas credenciales, las cuales deberían de ser como usuario *root* y una contraseña para que se complete el proceso sin errores. Sin embargo, es interesante poder acceder a *phpMyAdmin* aunque haya una serie de *plugins* de *WordPress* que permitan realizar consultas. Una vez completada la instalación, no permite acceder a *phpMyAdmin* debido a que, por defecto, se accede con el usuario *root* y sin contraseña. Por tanto, habría que escribir exactamente la misma que se puso para instalar la base de datos de *WordPress*. Para ello, es tan sencillo como acceder al fichero **config.inc.php** que se encuentra en la ruta **xampp/phpMyAdmin** y modificar las siguientes líneas tal y como aparece en la Figura 29:

```
$cfg['Servers'][$i]['user'] = 'root';  
$cfg['Servers'][$i]['password'] = '';
```

Figura 29: Código para cambiar credenciales de usuario en *phpMyAdmin*

Finalmente, una vez se reinicie el servicio *MySQL* desde el panel de control de XAMPP, permite acceder a *phpMyAdmin* correctamente.

Archivos corruptos en la base de datos

Una vez que se solucionaron los dos errores anteriores, tanto los servicios de XAMPP como el acceso a la web de *WordPress* funcionaban correctamente. Sin embargo, una vez transcurridas unas semanas, surgió un error al activar el servicio *MySQL* desde el panel de control de XAMPP.

En primer lugar se decidió reiniciar el servicio *MySQL* y, además, reiniciar el paquete de software libre XAMPP para cargarlo posteriormente. En segundo lugar, viendo que el problema permanecía, se cambió el puerto de ese servicio por si el mismo estaba obstruido u ocupado por otro, pasando del 3306 por defecto a probar con el 3307 o incluso el 3308. Sin embargo, el problema seguía siendo el mismo, sin posibilidad alguna de ejecutar el servicio y, por consiguiente, no poder ejecutar la aplicación en *WordPress*.

En cambio, después de probar las soluciones anteriores para solventar el error, se consiguió localizar el origen del error y no era otro que archivos corruptos en la base de datos. En otras palabras, quiere decir que cierta parte de la base de datos está inoperativa [17], congelada, fuera de uso, etc. Por tanto, no se puede acceder a la web y, por consiguiente, supone un serio problema que hay que solventar. El recurso para solucionar este contratiempo se realiza mediante los siguientes pasos:

- Crear una carpeta *data_old* que contendrá todo el contenido de la carpeta *data* a modo de copia.
- Sustituir todo el contenido de la carpeta *data* por el que se encuentra en la de *backup*.
- Volcar de la carpeta *data_old* todo el contenido a *data* a excepción de las carpetas:
 - *mysql*
 - *performance_schema*
 - *phpmyadmin*
- Sustituir el fichero *ibdata1* ubicado en *data* por el que se encuentra en *data_old*.

En la Figura 30 se puede ver dónde se ubican los archivos de la base de datos cuando se usa XAMPP, marcando las carpetas **mysql**, **performance_schema** y **phpmyadmin** y el archivo **ibdata1** como aquellos que se deben de cambiar de carpeta **data** para solventar el error que se acaba de explicar.

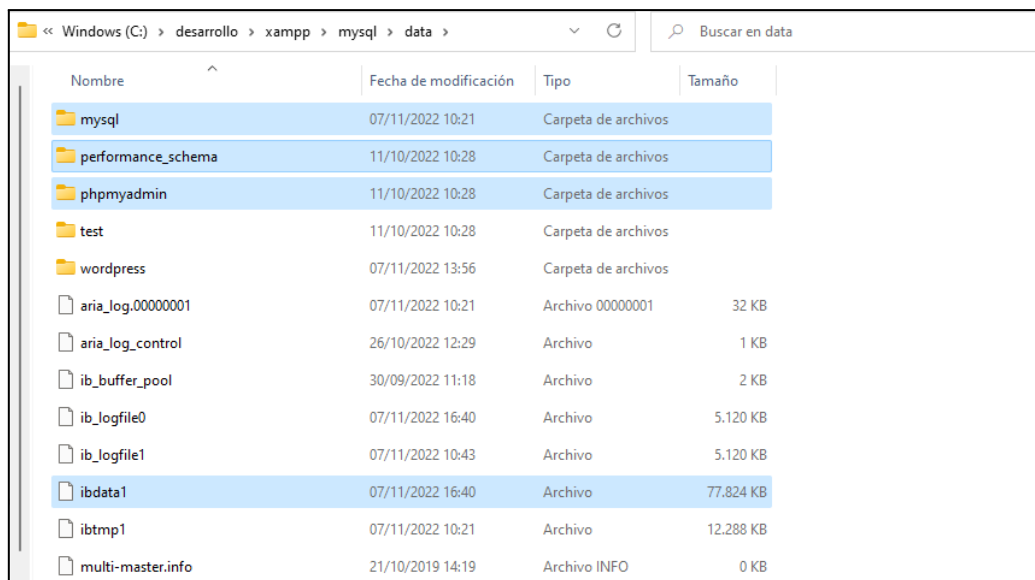


Figura 30: Ruta de archivos de la base de datos en XAMPP

Finalmente, se reinicia el servicio *MySQL* y acaba funcionando correctamente.

Error en la función `wp_cache_get`

Se trata de un error poco frecuente en aplicaciones simples de *WordPress*, pero que se ha dado y deja inaccesible la página aunque carguen perfectamente los servicios de Apache y *MySQL* desde el paquete de software libre (XAMPP).

En primer lugar, después de buscar donde se encontraba esa función en `xampp/htdocs/wordpress/wp_includes/option.php`, el autor del TFG probó volcar el contenido de una de las copias de seguridad que hacía cada semana, de forma que en ese momento la aplicación funcionaba correctamente. Se decidió crear otro directorio donde depositar la página web con *WordPress*, descargar el *plugin All-in-one WP Migration* y volcar la copia de seguridad que tenía en el Escritorio de su equipo. Sin embargo, no era posible importarlo debido a un error y tampoco era posible restaurar un *backup*, ya que para este método había que instalar versión *Premium* de la extensión utilizada.

En segundo lugar, al creer que el problema era un fichero de un directorio de *WordPress*, se decidió, primero, deshabilitar todos los *plugins* y temas utilizados en la aplicación y, posteriormente, eliminar los directorios completos ***wp_content*** y ***wp_admin***, pasando a continuación por cerrar y reiniciar los servicios utilizados. Sin embargo, el problema persistía y no se encontraba una solución para recuperar la web desde ese directorio.

Al tratarse de un error más complicado de resolver que el anterior, se decidió solventar migrando una copia de seguridad (tanto del contenido de *WordPress* como de la base de datos del mismo) por medio del *plugin Duplicator*, escogiendo una ruta alternativa donde almacenar el fichero comprimido en ZIP y el instalador con extensión PHP, aplicando los pasos necesarios para llevar a cabo el volcado del *backup* en un directorio. Cuando se realiza un *backup*, se parte de un fichero comprimido en *zip* y un fichero con extensión *.php*. Ambos se guardan en un directorio dentro de XAMPP, accediendo a esa ruta desde el navegador y utilizándose en la URL la ruta donde están dichos archivos, saliendo una página como la que se ve en la Figura 31.

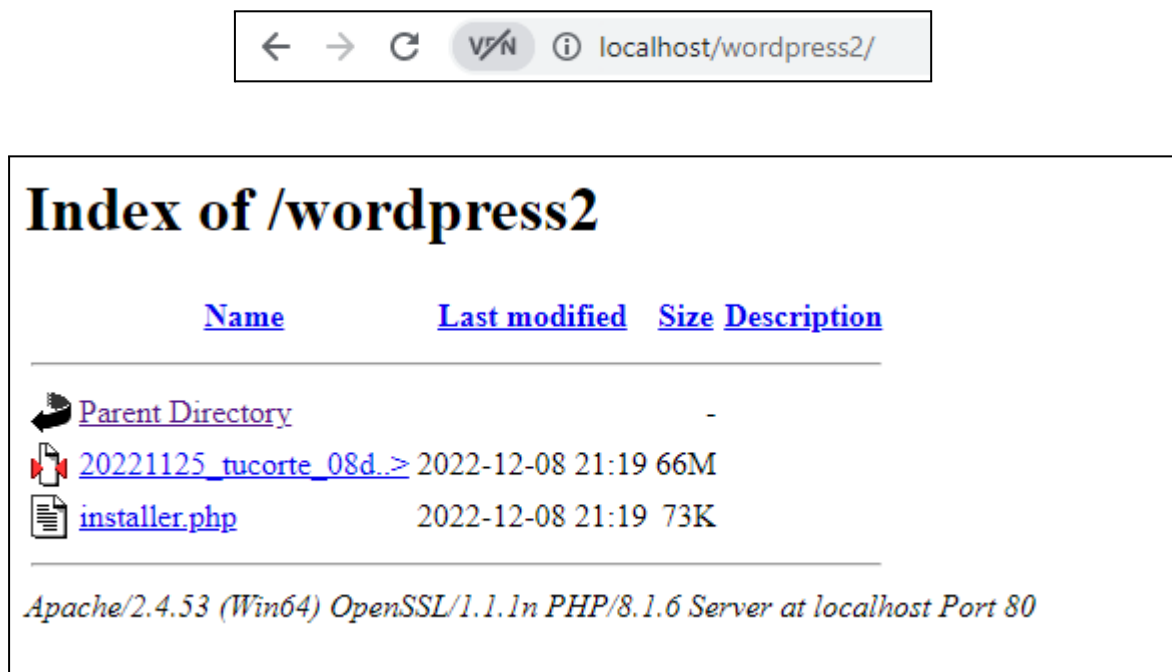


Figura 31: Archivos para volcar copia de seguridad

Una vez hecho esto, se completan los datos necesarios para realizar el volcado, como es el caso de introducir la base de datos (nombre de la misma que se desee recuperar, así

como el usuario y contraseña). En la Figura 32 se puede ver cómo seguir los pasos para realizar el volcado.

Step 1 of 2: Deployment Basic Advanced
This step will extract the archive file, install & update the database.

Overview

Installation Archive

View: Try [Basic](#) new or [Advanced](#) views ⓘ
Status: Standard Single Site Setup
Mode: Standard Install

Install Type: Full install single site

Setup

Database Settings

Database Connection Default cPanel

Action: ▾
Host:
Database:
Warning: The selected 'Action' above will remove **all data** from this database!
User:
Password:

Validation

ⓘ

Figura 32: Pasos para volcar backup

Si no se sabe con exactitud cuáles son las credenciales solicitadas, se pueden conseguir por medio de *phpMyAdmin* o accediendo al fichero *my.ini*.

Una vez se completan los pasos para realizar el volcado, se muestran todos los datos de la base de datos, contenido de las páginas, panel de control, temas y *plugins* correctamente.

9.3 Aclaraciones sobre cuándo usar *WordPress*

Aquí se observa una serie de aspectos sobre cuándo es conveniente utilizar *WordPress* y cuándo es preferible otra plataforma:

- La gestión de contenido es fácil de entender y funcional, de forma que el administrador es capaz de añadir y editar el contenido de forma sencilla sin utilizar diseñadores web. Además, permite crear menús personalizados y elementos de navegación de forma sencilla sin utilizar código.
- La gestión de usuarios es fácil y segura gracias al control de acceso a su contenido, el sistema de roles establecido y las capacidades de cada usuario, de forma que, según su rol, podrá ejercer más o menos tareas dentro de la aplicación.
- Permite añadir complementos, tanto gratuitos como *premium*, que se encuentran dentro del repositorio de *WordPress* y con los que se puede ahorrar tiempo y dinero.
- Contiene actualizaciones de seguridad frecuentes de forma que cada vez se vuelve menos vulnerable a posibles ataques informáticos.
- Tanto *WordPress* como algunos de sus servicios (PHP y MySQL) y la mayoría de sus complementos son gratuitos.

En cambio, como todo, también presenta sus motivos para utilizar otras aplicaciones:

- No se debe de migrar una aplicación si ésta se ha desarrollado en buena parte con otra plataforma, debido a que *WordPress* no tiene porqué tener las funcionalidades realizadas desde esa aplicación.
- Si es más necesario el rendimiento y optimización de la aplicación que el poder actualizarla rápidamente.
- Si se planea vender o aplicar licencia al sitio web ya que, debido a su licencia general de GNU [18], no puede restringir al usuario final ninguna acción al poseer libertad para utilizar, modificar y distribuir el *software*.
- Si la aplicación debe ser altamente en tiempo real como por ejemplo una sala de chat o un juego multijugador, ya que se sale de la arquitectura típica de *WordPress*, que consiste en que un usuario visita una URL, se comunica con un servidor web a través de HTTP, inicia un script PHP para generar la página (por ejemplo con la intención de ver los datos de un usuario) y devuelve una página completa al usuario.

9.4 Aspectos importantes de los *plugins* a utilizar

Antes de instalar los *plugins* se ha comprobado, dependiendo de la funcionalidad que queremos añadir, la información de cada *plugin* disponible, es decir, su descripción, registro de cambios, capturas de pantalla y valoraciones. Para seleccionar los *plugins* instalados y desechar el resto, a nivel general, se ha seguido el siguiente proceso:

- Comprobar que el *plugin* seleccionado se ha probado con la versión de *WordPress* instalada. Esta información sale reflejada para cada extensión.
- Ver las valoraciones y comentarios de otros usuarios que han utilizado el *plugin* en cuestión, ya que es importante asesorarse de la experiencia de diferentes usuarios que han desarrollado una aplicación con este CMS.
- Observar las capturas de pantalla dentro de la información de cada *plugin*, ya que puede resultar más sencilla la configuración del mismo y con un diseño más elegante e intuitivo que en otras extensiones.

Mientras que si tenemos en cuenta el tipo de funcionalidad, hay que tener en cuenta las siguientes consideraciones además de las ya mencionadas:

- Para habilitar un catálogo de productos, la intención era obtener una *plugin* en el que los usuarios cliente sólo puedan visualizar el contenido de cada establecimiento adherido, así como realizar valoraciones sobre los productos. Mientras que otros *plugin* proporcionan aspectos como un carrito para añadir productos, la opción de compra *online* y revelar el precio de los productos.
- Para realizar el sistema de citas, es imprescindible que cada negocio adherido tenga su propio calendario en el que gestionar las citas de los clientes a su establecimiento. Además, se puede añadir más trabajadores dentro de un mismo negocio y establecer los servicios realizados por cada negocio.
- En el caso de los *backup*, me he decantado por un *plugin* cuyos pasos a seguir tanto para crear como para migrar una copia de seguridad sean sencillos y, sobre todo, fiables.
- En los registros de usuarios, se utiliza un *plugin* en el que se pueden crear diversos formularios (cada uno para un rol) y estos se elaboran de una forma sencilla y, a su vez, lo más elegante posible.

- Un acceso directo en cada vista que permite escribir en formato HTML y CSS de una forma sencilla y sin tener que editar los ficheros creados en *WordPress* por defecto.

9.5 Curva de aprendizaje sobre *WordPress*

En este punto se explica la curva de aprendizaje acerca de la plataforma utilizada para el desarrollo de la aplicación.

A diferencia de otras herramientas comentadas anteriormente, *WordPress* tiene una curva de aprendizaje menor, es decir, es la que se requiere menor esfuerzo del usuario, dado que su funcionamiento, complementos y temas se enfocan en una mayor ayuda al usuario más que a los programadores. Sin embargo, no significa que sea una herramienta menos potente para el desarrollador que otras, más bien todo lo contrario, debido a que proporciona:

- Una buena estructura organizada.
- Facilidad de programación mediante eventos [19] que permiten manipular el procedimiento, conectando funciones que se encuentren en los archivos dentro del núcleo de *WordPress*, es decir, las carpetas *wp_admin*, *wp_includes* y *wp_content*. Esto sirve para ampliar o modificar el contenido que muestra una función por defecto en *WordPress*.
- Funciones por medio de *plugins* o programando código.

Todo ello otorga libertad, flexibilidad y escalabilidad a la hora de realizar funcionalidades en la aplicación al desarrollador.

Para poder comprobar de una manera visual la curva de aprendizaje de los diferentes CMS más utilizados (*WordPress*, *Drupal* y *Joomla*), se puede apreciar en la Figura 33 una gráfica (en forma de caricatura y muy exagerada) recogida de la fuente ***WordPressDirecto*** [20] del año 2013.

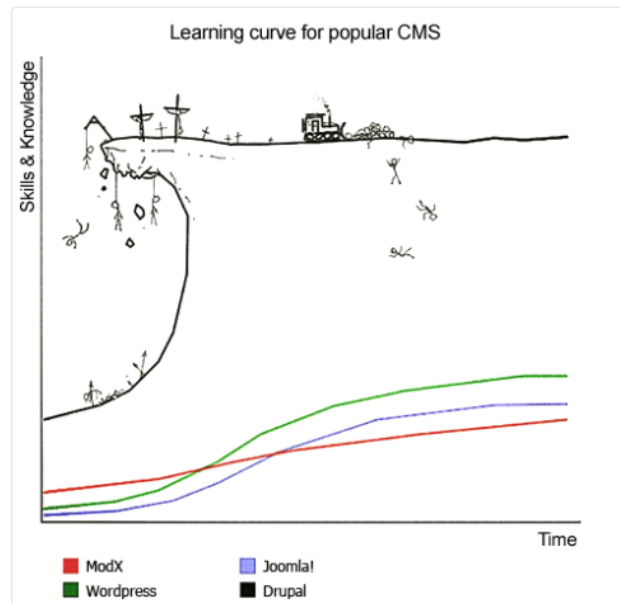


Figura 33: Curva de aprendizaje de los CMS

Con ello sólo se quiere transmitir al lector que la percepción que existe entre la comunidad de desarrolladores es que el uso del CMS *WordPress* es el que mejor curva de aprendizaje presenta, obteniendo un equilibrio entre el conocimiento que se requiere para implementar una aplicación y el tiempo que se emplea para el mismo. En cambio, *Drupal* exige de primeras un nivel avanzado de conocimientos para llevar a cabo su implementación y luego sube bastante el nivel en un período más corto de tiempo.

10. Conclusiones y trabajo futuro

Se ha desarrollado una aplicación multiplataforma en *WordPress* que permite realizar citas de usuarios a los servicios de un negocio de empresas registradas. El nicho en el que me he centrado para la realización de esta aplicación es el de las peluquerías. También se ha asignado dos roles que realizan funcionalidades diferentes, como son el de cliente y el de empresa. Por parte de los clientes, son capaces de, además de realizar funcionalidades básicas en toda aplicación como el registro, inicio de sesión y contactar con el administrador, buscar y opinar sobre los negocios adheridos a la aplicación y, lo más importante, realizar citas *online* a un determinado negocio sobre un servicio que realizan a una fecha y hora concretas. Mientras que por parte de la empresa, son capaces de gestionar las citas de los clientes en un calendario con información sobre la misma en ventanas emergentes y, además, pueden actualizar el contenido de su producto en concreto que se almacena en el catálogo de la aplicación.

Sobre este trabajo se ha aprendido durante su transcurso lo interesante que es realizar una aplicación web multiplataforma con el *framework WordPress*. Hace tiempo se le conocía más como una herramienta para blogs, pero ahora es un programa con un gran potencial y apto para cualquier tipo de página web. Es capaz de facilitar mucho el desarrollo de una aplicación gracias al potencial que ejerce, a nivel funcional, gracias a la instalación de *plugins* y, a nivel gráfico, por la gran diversidad de temas personalizables que pueden acoplarse perfectamente a cualquier tipo de página web. Además, si dichos temas no son del agrado del desarrollador, admite el uso de una plantilla *Bootstrap*.

Por otro lado, instalar *WordPress* no es nada difícil ni tiene porqué suponer un coste, ya que se puede utilizar a nivel local partiendo de un paquete de software libre como es XAMPP. En este caso, es importante haber realizado la instalación y trabajo de esta forma, ya que se puede controlar y utilizar para pruebas para que, finalmente, se pueda lanzar desde un dominio web. A nivel de seguridad puede que no sea el mejor, pero para una pequeña aplicación flexible y sencilla en el que no se requiere muchos datos personales a los usuarios en los formularios es idónea. Sin embargo, la realización de copias de seguridad y migración de las mismas es sencilla y muy segura gracias al uso de *plugins* como es el caso de *Duplicator*, consiguiendo importar en otro equipo tanto el contenido de la página de *WordPress* como el de su base de datos. No obstante, aunque para lanzar una

página web sencilla cubra las necesidades básicas del sistema, siempre hay que tener en cuenta que, a lo largo del tiempo de vida de la aplicación, se podrá ver detalles que habrá que modificar y/o actualizar de vez en cuando.

En cuanto a mi experiencia personal utilizando este *framework* puedo decir que se trata de una herramienta muy potente para desarrollar cualquier tipo de aplicación. Se le puede asignar, a través de *plugins* que proporciona *WordPress*, una serie de funcionalidades que pueden hacer muy interesante una página web para cualquier usuario. Además, para aplicar un diseño vistoso y sencillo se puede aplicar una serie de temas que contiene esta herramienta o bien importar plantillas *Bootstrap* que pueden encajar muy bien con el contenido de la página creada. A lo largo del proyecto, se han propuesto otras herramientas como *Joomla*, *PrestaShop* y *Laravel* para realizar un sistema de citas en forma de aplicación. Sin embargo, se ha optado finalmente por *WordPress* al hacer de forma más sencilla y gráfica que una página tenga funcionalidades distintas adaptadas para diversos roles.

A continuación se mencionan a las ampliaciones futuras de la aplicación, que podrían ser tales como:

- Mostrar precios de los servicios en cada establecimiento (en peluquería los cortes de pelo, barba; estética para la pedicura, manicura, ...).
- Modificar la franja horaria de las citas por servicios solicitados y/o género del usuario que la vaya a solicitar, ya que no es lo mismo el corte de pelo de un hombre que el de una mujer y, además, pueden tener servicios únicamente disponible para hombres como por ejemplo hacerse la barba o para mujeres un alisado de keratina.
- Añadir el pago del servicio citado desde la aplicación por distintas vías, como puede ser el caso de tarjeta de crédito, paypal o bizum.
- Sincronizar el calendario de reservas con los calendarios de los dispositivos.
- Exportar detalles de la cita a formato PDF.
- Añadir ubicación vía Google Maps.

Referencias

[1] COVID-19:

<https://es.wikipedia.org/wiki/COVID-19>

[2] *WordPress* como tendencia en muchas empresas:

<https://www.webempresa.com/blog/estadisticas-WordPress.html>

[3] Pirámide de requisitos:

[IDR Tema 03b Modelado de Casos de Uso.pdf](#)

[4] Las mejores herramientas y aplicaciones Software para peluquería:

<https://www.getapp.es/directory/212/salon/pricing/free/software>

[5] *WordPress* vs *PrestaShop*:

<https://marketingblog.es/prestashop-vs-wordpress-escoger-crear-tienda-online/>

[6] *WordPress* vs *Joomla*:

<https://diferenciario.com/wordpress-y-joomla/>

[7] *WordPress* vs *Drupal*:

<https://www.redeszone.net/2019/02/24/wordpress-vs-drupal-diferencias/>

[8] Creación de aplicaciones web con *WordPress*:

https://www.oreilly.com/library/view/building-web-apps/9781491990070/ch01.html#chapter_1

[9] Paquete de software libre XAMPP:

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

[10] Sistema de gestión de contenidos o *Content Management System*:

https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos

[11] Instalación de XAMPP y *WordPress* en servidor local:

<https://www.ciudadano2cero.com/instalar-wordpress-xampp/>

[12] Asignar privilegios y roles a usuarios con *Nav Menu Roles* de *WordPress*:

<https://www.francogiardina.com/menu-wordpress-segun-roles-usuario/>

[13] Guía para el uso del *plugin Shortcodes Ultimate* de *WordPress*:

<https://getshortcodes.com/docs/user-data/>

[14] Guía para migrar *backups* con la extensión *Duplicator* de *WordPress*:

<https://ebweb.es/tutorial-duplicator-plugin-para-migrar-wordpress/>

[15] Catálogo de productos para *WordPress*:

<https://es.wordpress.org/plugins/ecommerce-product-catalog/>

[16] Tutorial acerca del calendario de reservas *Easy Appointment*:

<https://code.tutsplus.com/es/tutorials/get-started-with-a-free-booking-calendar-plugin-in-wordpress--cms-32908>

[17] Solución del error de base de datos inoperativa:

<https://www.devopsschool.com/blog/error-mysql-shutdown-unexpectedly/>

[18] Licencia general GNU de *WordPress*:

<https://kinsta.com/es/aprender/wordpress-gpl/>

[19] Uso de eventos en *WordPress*:

<https://ayudawp.com/ganchos-filtros-acciones-wordpress/>

[20] Curva de aprendizaje sobre los CMS más conocidos:

<https://wpdirecto.com/la-curva-de-aprendizaje-segun-el-cms/>

Anexos

A continuación, se muestran los *mockups* realizados para la aplicación, de forma que se muestran la vista para iniciar sesión por parte de un usuario en concreto y otra con la página de inicio de un usuario con rol de empresa en la Figura 34. En cuanto a la Figura 35 podemos observar, de izquierda a derecha, los formularios de registro que se tienen que completar para el usuario cliente y empresa respectivamente. Mientras que en la Figura 36 se ve cómo desde un usuario cliente se puede comprobar los detalles de una cita (si se realizó anteriormente), pudiendo anularla a través del botón que se encuentra y, además, se puede ver cómo un usuario empresa puede ver una tabla con los usuarios que han confirmado una cita en su establecimiento en concreto.

Mockups

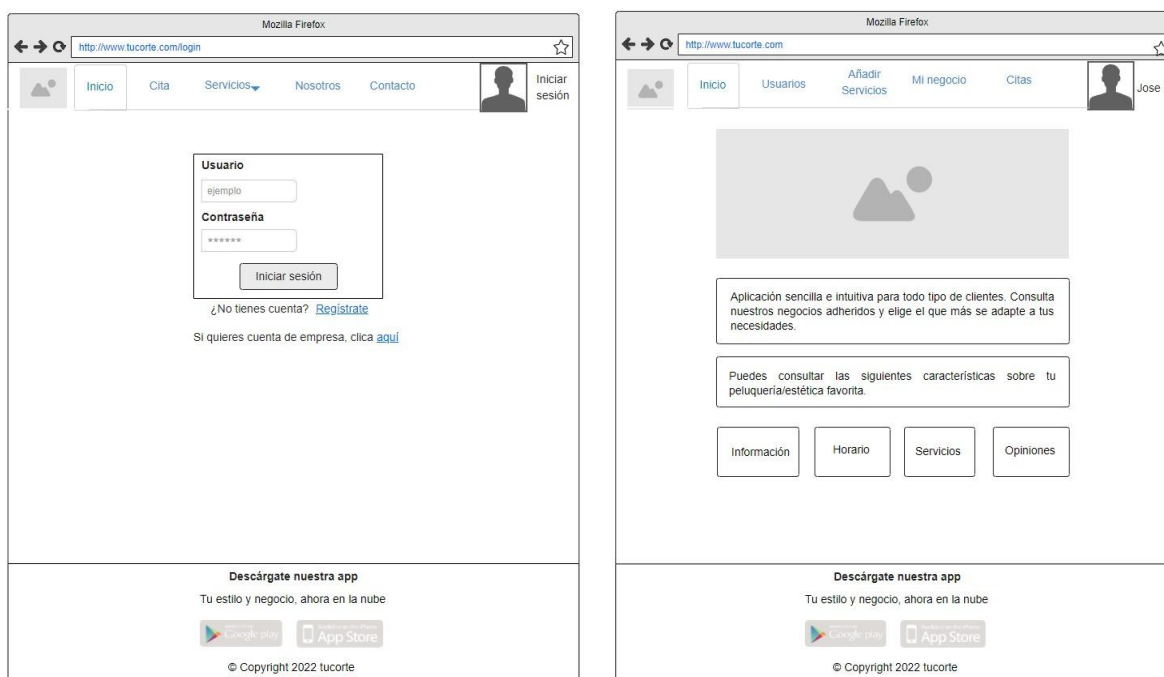


Figura 34: Mockups 1

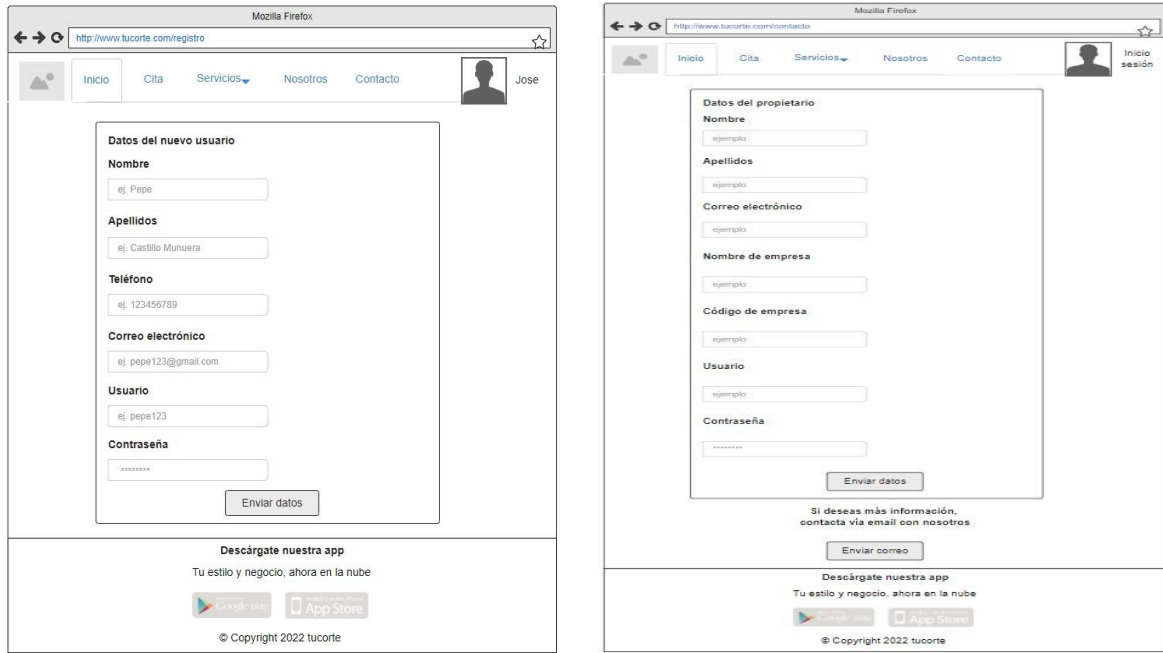


Figura 35: Mockups 2

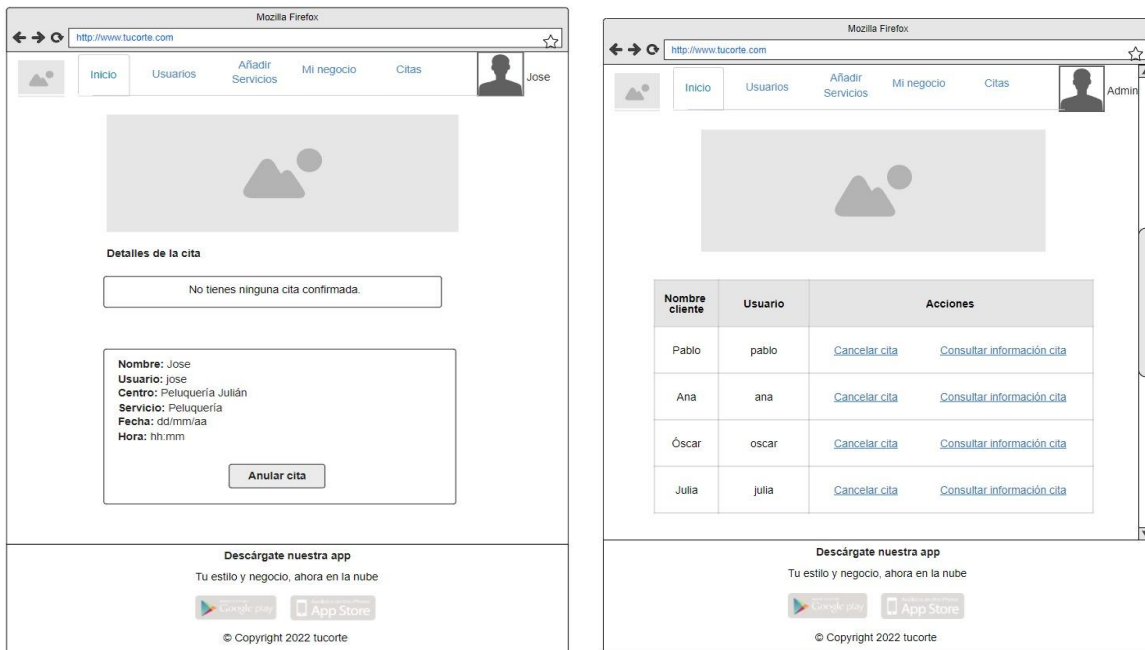


Figura 36: Mockups 3