

Trabajo de fin de Grado

Famosso Barber

Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web



Presentado por: Víctor Pérez y Alexandre Ribeiro

Fecha de entrega: 03/06/2022

Índice

| | |
|---|----------|
| 1. Abstract | 3 |
| 2. Justificación | 4 |
| 3. Introducción | 5 |
| 4. Objetivos | 6 |
| 5. Temporalización, planificación y costes | 7 |
| 5.1 Costes | 7 |
| 5.2 Temporalización | 8 |
| 6. Desarrollo del proyecto | 9 |
| 6.1 Análisis de requisitos | 10 |
| Requisitos funcionales | 10 |
| Requisitos no funcionales | 11 |
| 6.2 Introducción de herramientas utilizadas | 12 |
| Gestor de bases de datos | 12 |
| Github | 12 |
| Lenguaje de programación | 12 |
| PHP | 12 |
| SQL | 12 |
| 6.3 Diseño | 13 |
| Modelo entidad-relación | 13 |
| Casos de uso | 14 |
| Mapa web | 15 |
| 6.4 Implementación del proyecto | 17 |
| 6.5 Pruebas (plan de pruebas) | 21 |
| 7. Conclusiones y mejoras futuras | 22 |
| 8. Bibliografía | 23 |
| 9. Anexo | 23 |

Tabla de ilustraciones

| | |
|---|----|
| Ilustración 1: Tabla de costes..... | 7 |
| Ilustración 2: Diagrama de Gantt..... | 8 |
| Ilustración 3: Modelo entidad-relación..... | 13 |
| Ilustración 4: Casos de uso..... | 14 |
| Ilustración 5: Mapa web | 15 |
| Ilustración 6: Estructura de la página web Admin..... | 16 |
| Ilustración 7: Plan de pruebas..... | 21 |

1. Abstract

Nowadays there is great competition in local businesses in small and big cities, the constant growth of new technologies and the information age has led many small businesses to bankruptcy.

As a result of this, the Famosso Barber project has taken place, in the search to strengthen and give visibility to a local hairdressing salon for men that not only attracts customers through visibility on the web but also allows workers to make an easy and simple management of the services of their hairdressing salon.

Our application allows customers to view and request the different services provided by the business at that moment through an easy and simple web calendar, as well as allowing the hairdresser's administrator to manage the customers who access their page, the appointments they request and modify the services they provide.

2. Justificación

Llevamos a cabo este proyecto debido a la visión del dueño del establecimiento de dar un paso más y subir el nivel de su negocio. Es un muy buen momento para explotar las herramientas que la propia web nos ofrece y utilizarlas para la captación de clientes y el aumento de ventas.

Dadas las características del proyecto, hemos decidido utilizar PHP y SQL para la conexión con la base de datos y la codificación de la misma. Para el diseño de la página se ha utilizado Bootstrap, ya que se trata de un framework de CSS, altamente usado hoy en día y que permite una gama mucho más amplia de personalización.

Uno de los puntos más relevantes en la web, es un calendario de citas en el cual los clientes pueden ver que horario tiene el establecimiento, y que fechas tienen disponibles para disfrutar del servicio. El cliente interactúa con el calendario, elige el día, la hora y el servicio que necesita.

3. Introducción

El desarrollo de la aplicación está orientado a la creación de una página web para el establecimiento llamado “Famosso Barber”, situado en Colmenar Viejo, Madrid.

La elección de este proyecto surgió por la idea de relanzar el negocio ofreciendo visibilidad online, una vía de promoción en auge actualmente.

El principal objetivo es, ofrecer la información detalladamente de todos los servicios que ofrece el establecimiento y poder así captar más clientes que los habituales.

Para conseguir el objetivo se llevará a cabo el desarrollo de un calendario de citas donde los clientes podrán ver que días y que horas están disponibles y reservar, eligiendo el servicio que necesitan. Cada servicio tendrá un tiempo límite predeterminado para facilitar el trabajo a la encargada de realizarlo.

4. Objetivos

En las reuniones con el dueño de la peluquería hemos sacado diferentes conclusiones y objetivos:

- Registro de usuarios, que servirá para aumentar la cartera de clientes.
- Facilitar la gestión de citas, mediante un crud conectado a la base de datos donde están almacenadas todas las citas.
- Facilitar el control de servicios y precios de la peluquería, al igual que con las citas, se mantienen organizados mediante el crud de servicios.
- Ofrecer una interfaz web con el contacto, siendo esta una interfaz muy intuitiva y clara.
- Dar de alta y eliminar usuarios, gracias al crud que hemos creado puedes tener organizados los usuarios.

5. Temporalización, planificación y costes

En este apartado veremos los costes y la temporalización del proyecto, cómo nos hemos organizado y el tiempo que nos ha llevado.

5.1 Costes

Para los costes los hemos dividido en mano de obra, que sería lo que cobraría cada uno de nosotros, siendo uno de los dos desarrolladores y otro como jefe de proyecto; materiales, al haber utilizado aplicaciones de software libre, lo hemos puesto con coste cero y por último costos indirectos que serían los ordenadores y sus periféricos para poder trabajar en ellos.

| Estructura de costos de la página web de Famosso Barber | | | | | | |
|---|----------|----------------|---------------|--|--|--|
| Concepto | Cantidad | Costo Unitario | Costo Mensual | | | |
| MANO DE OBRA | | | | | | |
| Desarrolladores | 1 | 1000 | 1000 | | | |
| Jefe de proyecto | 1 | 2500 | 2500 | | | |
| Total Mano de Obra | | | 3500 | | | |
| MATERIALES | | | | | | |
| Aplicaciones de Software libre | 2 | 0 | 0 | | | |
| Total Materiales | | | 0 | | | |
| COSTOS INDIRECTOS | | | | | | |
| Ordenadores | 2 | 600 | 1200 | | | |
| Monitores | 4 | 150 | 600 | | | |
| Periféricos | 4 | 20 | 80 | | | |
| Total Costos Indirectos | | | 1880 | | | |

| RESUMEN | | Costo por mes |
|-------------------|--|---------------|
| Mano de obra | | 3500 |
| Materiales | | 0 |
| Costos indirectos | | 1880 |
| Total mensual | | 5380 |

Ilustración 1: Tabla de costes

5.2 Temporalización

Para la temporalización lo hemos dividido en 4 bloques: análisis y requisitos, desarrollo del proyecto, desarrollo e implementación y conclusiones y mejoras.

- El bloque de Análisis y requisitos fue lo que hicimos al inicio del todo y lo que trabajamos más en grupo, hablando cuál era la idea del proyecto, yendo a hablar con el dueño del local para que nos comentara que necesita y cómo lo quería...
- En el segundo bloque, Desarrollo del proyecto, empezamos a dividir las tareas del documento y empezamos a modificar las páginas del front.
- En el tercer bloque, Desarrollo e implementación, empezamos a desarrollar lo que es el código PHP y las bases de datos.
- Y por último, en el último bloque, dimos los últimos retoques al documento y a la aplicación web.

A la hora de trabajar utilizamos una aplicación llamada discord que es un chat de voz donde se puede compartir pantalla a la vez que hablamos por voz. Nos reuníamos cada 2-4 días y comentamos los puntos que íbamos a trabajar en estos y quién haría cada tarea.

FAMOSSO BARBER

Famosso barber
Alexandre Ribeiro y Víctor Pérez

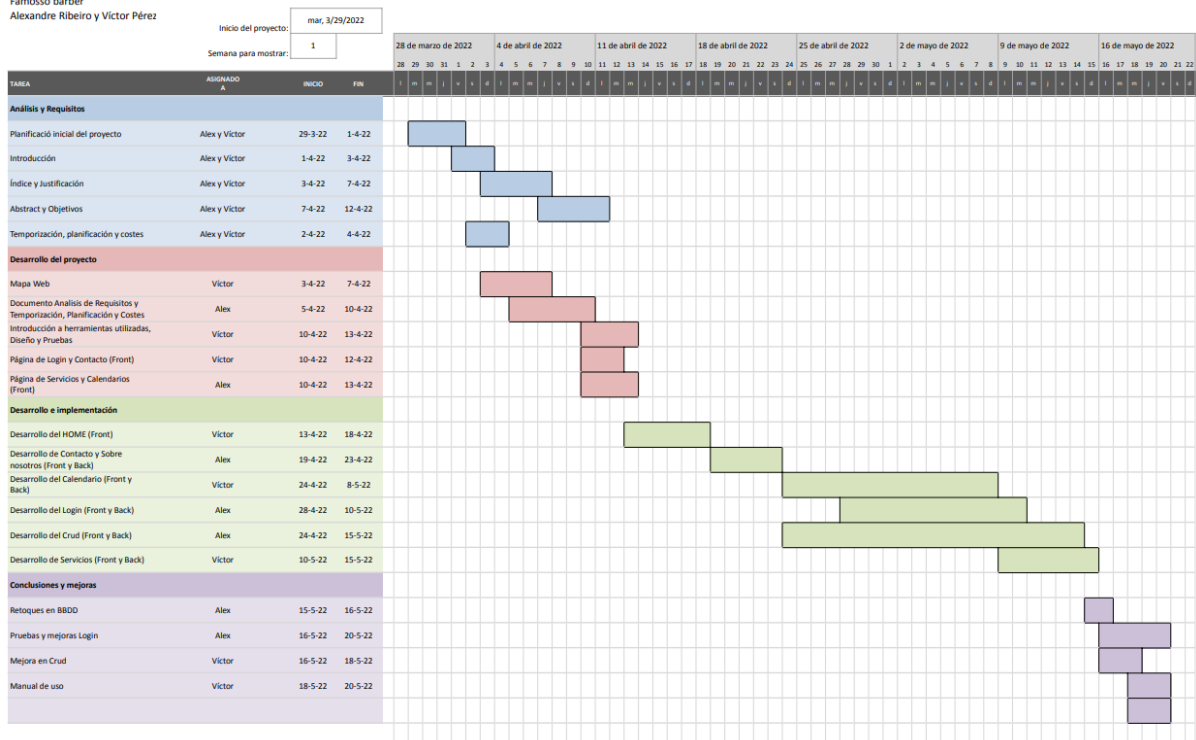


Ilustración 2: Diagrama de Gantt

6. Desarrollo del proyecto

En primer lugar, se llevó a cabo un análisis exhaustivo de las necesidades del cliente, teniendo muy en cuenta los puntos a potenciar y los servicios que ofrecer. Una vez realizado el análisis, comenzó la planificación del diseño de la web y sus funcionalidades. El equipo presentó al cliente la estructura de las landings a desarrollar, las funcionalidades, la usabilidad y accesibilidad para el usuario.

Entre estas propuestas se encuentran las diferentes funcionalidades de la aplicación, estas son: el login de usuarios que redirecciona directamente al CRUD en caso de insertar los datos del administrador y al calendario para los usuarios. El propio calendario, que cuenta con un sistema de sesiones y sólo es accesible a través del login de un usuario, también cuenta con una opción de cerrar la sesión para evitar mantener una sesión activa. Este calendario también interactúa con las sesiones recogiendo la información del usuario que se ha logueado para registrar la cita. El CRUD que es una de las funcionalidades más importantes de la aplicación, este permite al gestor de la página administrar los usuarios, servicios y citas de la Base de datos a través de una interfaz simple y sencilla. El CRUD sólo es accesible con credenciales de administrador, las cuales son proporcionadas por nosotros. Una vez dentro del CRUD podemos ver los usuarios, modificarlos y borrarlos, ver las citas y borrarlas y ver los servicios, modificarlos y añadir servicios nuevos, permitiendo una interacción directa con la pestaña de servicios para mantener el catálogo de servicios actualizados.

Todas estas funcionalidades han sido desarrolladas a conciencia de facilitar la gestión de la aplicación para que se mantenga actualizada para los clientes y de fácil manejo para el administrador.

6.1 Análisis de requisitos

Para este apartado, nos fuimos a ver al dueño del local y nos estuvo comentando lo que necesitaba y cómo le gustaría que fuera el diseño y la organización.

- Requisitos funcionales

Estos requisitos definen la funcionalidad de la aplicación, lo que puede hacer. Estos son algunos de los requisitos funcionales que cumple el proyecto:

- **RF 01 Registro usuario**: El usuario puede registrarse en el login introduciendo un email, nombre y contraseña válidos.
- **RF 02 Solicitud de cita**: El usuario puede pedir cita de un servicio seleccionando el que desee en el calendario.
- **RF 03 Contacto**: El usuario puede ponerse en contacto con el administrador mediante el formulario en la página de contacto.
- **RF 04 Login crud**: El crud cuenta con un login para que solo el administrador pueda acceder a él.
- **RF 05 Gestión de usuarios**: Hay una tabla en el crud que hace que el administrador pueda añadir, eliminar o modificar un usuario. La modificación de la contraseña del usuario se mantendrá encriptada.
- **RF 06 Gestión de servicios**: Hay una tabla en el crud que hace que el administrador pueda añadir, eliminar o modificar un servicio indicando el nombre del servicio, el precio del mismo y añadiendo una imagen para visualizarse en el servicio.
- **RF 07 Gestión de citas**: Hay una tabla en el crud que hace que el administrador pueda eliminar una cita.

- Requisitos no funcionales

Estos son requisitos que imponen restricciones en el diseño o la implementación como restricciones en el diseño o estándares de calidad. Son propiedades o cualidades que el producto/aplicación debe tener.

- **RNF 01 Interfaz del sistema:** La aplicación presentará una interfaz de usuario sencilla para que sea de fácil manejo para los usuarios de la aplicación. Prioridad Alta.
- **RNF 02 Desempeño:** La aplicación deberá estar alojada en un servidor eficiente que pueda manejar gran concurrencia de usuarios en ciertos periodos de tiempo. Prioridad Media.
- **RNF 03 Sistema Robusto:** El tratamiento de los datos debe ser de forma segura, cumpliendo los requisitos básicos de seguridad. También introducimos consultas preparadas y para el login contamos con password hash. Prioridad Alta.
- **RNF 04 Documentación:** La aplicación deberá disponer de una buena documentación que permita realizar operaciones de mantenimiento con el menor esfuerzo posible. Prioridad Alta.
- **RNF 05 Diseño para diversas resoluciones de pantalla:** La aplicación debe desarrollarse para adaptarse a diferentes resoluciones de pantalla. Prioridad Alta.
- **RNF 06 Accesibilidad:** Se comprueba la accesibilidad en diferentes páginas web. Prioridad Alta.
- **RNF 07 Arquitectura:** El sitio web será compatible con los principales navegadores de Internet (Mozilla Firefox, Internet Explorer, Opera, Safari, Google Chrome). Los contenidos se almacenarán en un sistema gestor de bases de datos relacional. Prioridad Alta.

6.2 Introducción de herramientas utilizadas

Estas son las principales herramientas utilizadas en la aplicación:

- Gestor de bases de datos

Se usa XAMPP y Apache como servidores locales con sus respectivas bases de datos en PHPMyAdmin para realizar las pruebas necesarias durante todo el proceso de producción. El lenguaje utilizado es SQL, el cual hemos utilizado para la creación de las bases de datos y toda la conexión con la parte frontend, tanto en el calendario, en la sección de servicios y en la sección de productos.

- Github

Utilizamos la aplicación de Github Desktop y la web de github para almacenar nuestro proyecto en internet y así poder sincronizar, compartir y que no tengamos ningún problema a la hora de modificar código.

- Lenguaje de programación

Se propone como lenguaje de programación del proyecto a desarrollar PHP. Es un lenguaje de programación interpretado, de código abierto, muy popular, especialmente adecuado para desarrollo web.

- PHP

Este lenguaje nos permite desarrollar la página favoreciendo la conexión entre los servidores y la interfaz de usuario. Junto con HTML, forman la estructura completa con la cual se llevará a cabo el desarrollo del calendario.

- SQL

Se crearán las bases de datos con este lenguaje y se llevará a cabo la conexión con PHP. Se usará una base de datos de cliente para la concertación de citas, para el manejo de los datos, bajo su consentimiento, para campañas de marketing especializadas y el posterior registro de las compras que realicen. También existirá una base de reservas de citas y por supuesto de productos.

6.3 Diseño

En el apartado de diseño vamos a ver y hablar sobre cómo interactúan las bases de datos y las páginas web entre sí.

- Modelo entidad-relación

Es un tipo de diagrama de flujo que ilustra cómo personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Se usan a menudo para diseñar o depurar bases de datos relacionales

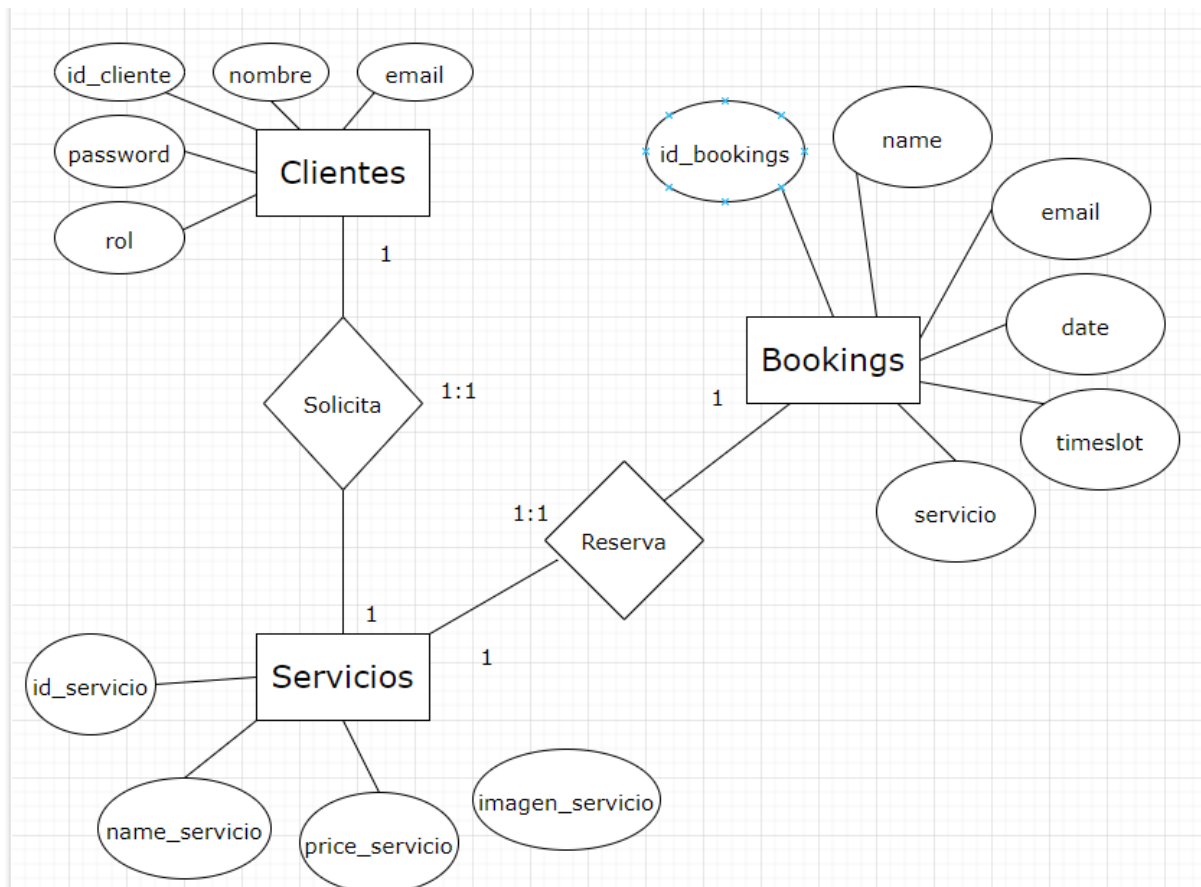


Ilustración 3: Modelo entidad-relación

- Casos de uso

Es un diagrama que define una secuencia de acciones que da lugar a un resultado de valor observable. Los casos de uso proporcionan una estructura para expresar requisitos funcionales.

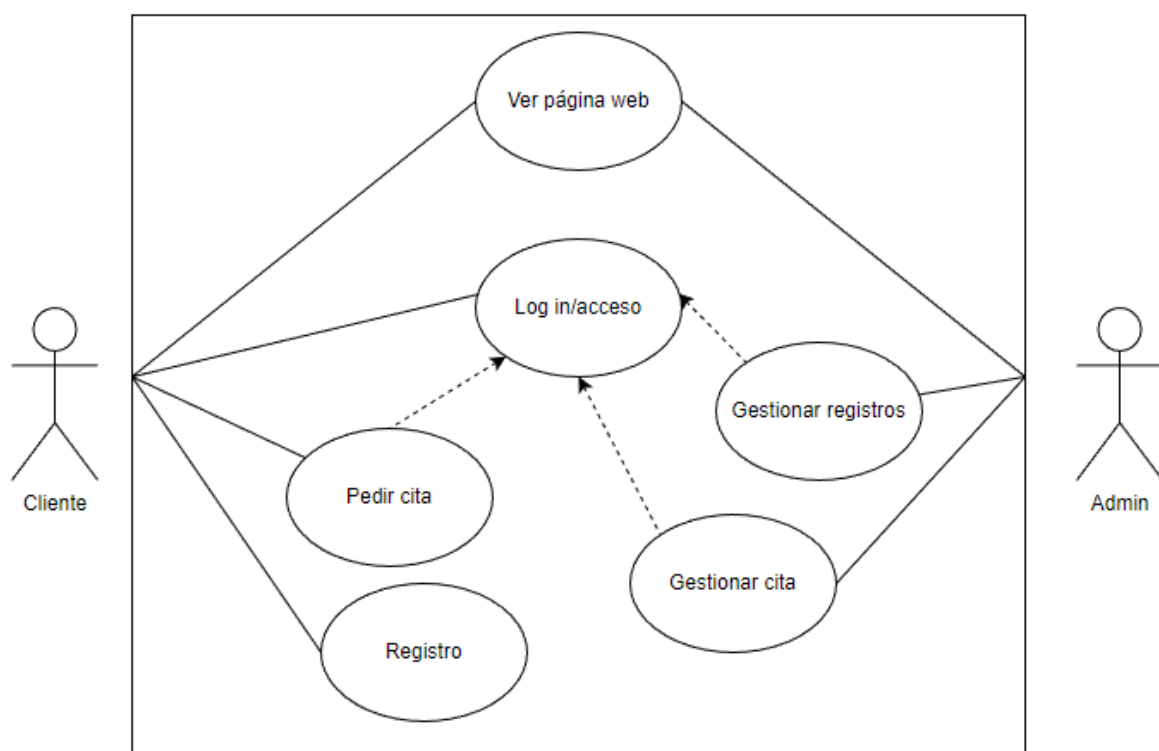


Ilustración 4: Casos de uso

En esta ilustración podemos observar las partes que van a hacer uso de la aplicación. Por un lado, tenemos al cliente, el cual accede a la página principal y al login, en el login tiene la opción de iniciar sesión o registrarse. Una vez iniciada la sesión accede al calendario de citas y solicita una cita indicando la fecha y la hora deseada.

Por otro lado tenemos el administrador, este también accede a la página principal para ver que los servicios existentes aparecen correctamente y accede al login donde inicia sesión con credenciales de administrador. Al acceder como administrador es redirigido directamente a la aplicación de gestión de datos, donde puede crear, editar y eliminar los diferentes registros de las tablas clientes, citas y servicios.

- Mapa web

Este es un diseño de la estructura de nuestra web y cómo se comunican las páginas entre sí a nivel usuario.

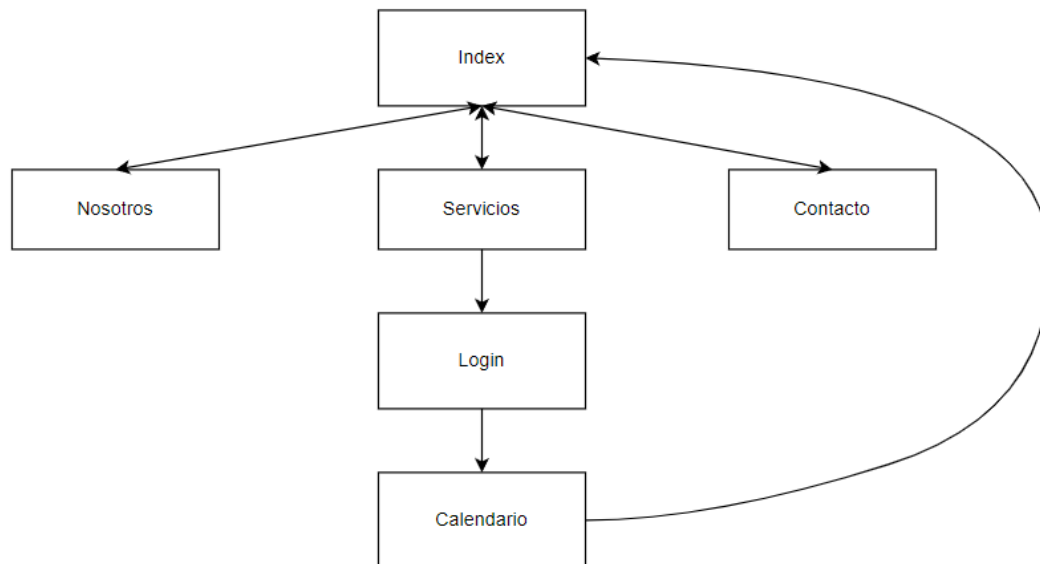


Ilustración 5: Mapa web

Este es un diseño de la estructura de la web a nivel interno para el administrador.

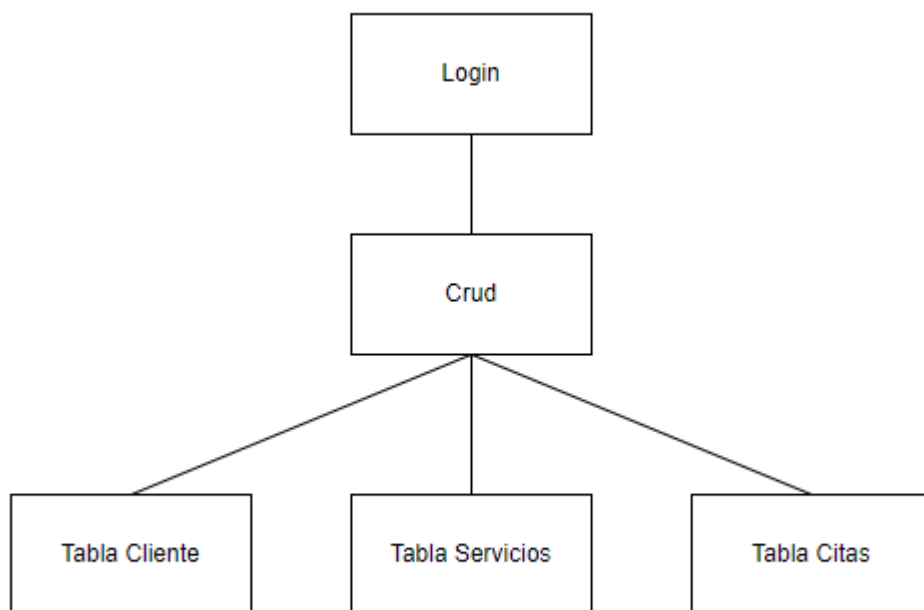


Ilustración 6: Estructura de la página web Admin.

6.4 Implementación del proyecto

El proyecto tiene 3 funcionalidades principales, el login, el calendario de citas y un crud para el administrador, vamos a tratar de explicar cómo funciona el código de cada una de estas funcionalidades.

Además de lo indicado, como cualquier página web, el proyecto necesita una interfaz para interactuar con la aplicación. En nuestro caso, para la parte del Front hemos utilizado una plantilla de bootstrap la cual hemos modificado algunas partes para ponerlo a nuestro gusto, como la parte de servicios para que salgan los tres servicios en horizontal y con las fotos que queremos nosotros de nuestra base de datos.

- Proceso de Login:

El login es un formulario creado en HTML, PHP y SQL.

Lo primero que hace nuestra aplicación es crear la conexión con la base de datos a través del archivo `conexion.php`, este archivo como indica su nombre realiza la conexión con la base de datos, se trata de un código PHP muy sencillo, el cual usando la función `mysql_connect()` crea la variable `$con`, donde almacena los datos de conexión, además mediante un `if()` recoge la posibilidad de un fallo al tratar de conectarse a la base de datos especificada. Una vez tenemos nuestra conexión realizada lo siguiente que procede a hacer el código del login es verificar si los datos del formulario han sido registrados, para ello usa un `if(isset($_POST[]))` de dicho formulario. En caso de que haya sido enviado procede a crear las variables correspondientes con su correspondiente encriptado (nombre, password y email). Una vez tenemos esto pasamos a verificar si los datos introducidos son correctos, para ello hace un `if()` para cada variable donde verifica si ha sido ingresado algún valor, tras lo cual limpia el valor y verifica que es un texto mediante el método `filter_var()`, en caso de no haberse introducido ningún valor devuelve un error para cada valor (nombre y email).

Una vez hecho esto pasa a comprobar que la conexión está correcta, de caso contrario arroja un error. Si la conexión es correcta procede a verificar si los datos introducidos coinciden con los datos existentes en la base de datos, para el caso del login simplemente verifica que el nombre de usuario existe y mediante el método `password_verify()` confirma que la contraseña introducida coincide con la contraseña encriptada en la base de datos.

Para el registro verifica que el nombre y el correo electrónico no coinciden con ninguno en la base de datos, en caso de no existir procede a almacenar los datos en la base de datos encriptando la contraseña con el método `password_hash()`. Una vez realizado estos procesos nos redirecciona a la página del Calendario.

En caso que los datos introducidos coincidan con los datos de un administrador, nos redirecciona al CRUD en su lugar.

- Gestión de acceso a datos:

El CRUD es un formulario creado en HTML, PHP y SQL al igual que el login, lo primero que hace es traer el archivo conexión, donde se realiza la conexión a la base de datos.

Seguidamente crea las query que se va a usar para mostrar los registros de las diferentes tablas. Estas queries las guarda en una variable cada una para posteriormente mostrarlas en el HTML. Este HTML crea 3 DIV principales donde se van a mostrar las tablas, estos son botones que al hacer clic en ellos despliega la sección que está escondida. Para cada sección se trae la tabla en un bucle while, donde hace un `mysql_fetch_assoc()` de nuestra variable donde está almacenada la consulta y muestra dentro de una etiqueta `<tr>` cada fila de la tabla de manera que recoge los datos directamente de la consulta almacenada en la variable. Hace esto para las tres secciones, una para cada tabla. Una vez hecho esto crea los botones de Editar y Borrar, los cuales están enlazados al id del registro de la tabla. Al hacer clic en crear registro nos redirige a la página de `crear.php`, la cuál está maquetada en HTML e incluye un formulario donde se introducen los datos del registro que queremos crear. Una vez introducidos los datos hace un `isset()` del formulario, verifica que los datos introducidos son correctos y prepara la sentencia insert. Una vez realizadas las validaciones lanza la query y devuelve un mensaje de confirmación y nos devuelve a la página principal del crud. Hace lo mismo para la opción de editar pero en lugar de una query insert hace un update.

Es importante destacar que cada tabla atacada por el CRUD tiene su peculiaridad: La tabla clientes permite la modificación de todos los datos del cliente, inclusive permite asignar privilegios de administrador a un usuario existente. Además las contraseñas modificadas también serán cifradas.

La tabla servicios permite crear servicios que se mostrarán en nuestra pantalla de servicios por ello nos permite no solo indicar el nombre y el precio del servicio, sino también añadir una imagen a mostrar en la pantalla.

La tabla citas nos permite crear, editar y eliminar las citas, para que en caso de que un cliente nos llame solicitando modificar una cita podamos hacer el cambio desde nuestra aplicación.

- Gestión de citas:

Para el calendario ha sido utilizado el lenguaje de programación PHP y HTML. Se ha creado una función en PHP para estructurar mediante una tabla un calendario donde aparezca el mes en el que estamos. Hay tres botones en los cuales puedes ir al mes anterior, al mes actual y al mes siguiente en el que se ha utilizado la función mktime para obtener el mes y el año. En cada celda hay un botón verde que te lleva a la página para ver las franjas horarias las cuales salen en verde si está disponible pedir una cita o en rojo si no está disponible esa hora. Una vez clicas en la hora que deseas pedir la cita, te salta una ventanita para que rellenes tu nombre, tu email y selecciones el servicio que deseas contratar (el cual está implementado desde la BBDD) y una vez lo envías, se añade el registro en la tabla "bookings" que es donde se guarda el nombre, email, día de la cita, hora de la cita y el número del servicio. Se han añadido dos botones de volver al calendario por si el cliente no encuentra cita el día que quería a la hora que quería y otro botón para volver al inicio de la página por si quiere ir al contacto o seguir visitando la web.

Por último, comentar que toda la parte de la documentación del código ha sido creada durante el desarrollo del proyecto, dentro del código se ha añadido diferentes comentarios explicando lo que hace la función o la etiqueta para facilitar la rápida localización de los errores que haya podido haber durante el proceso o que pueda haber en el futuro.

6.5 Pruebas (plan de pruebas)

Las pruebas las hemos ido haciendo según íbamos terminando de configurar cada parte del código. Probamos cada cosa dos o tres veces para asegurarnos de que funcionaban correctamente y de todas las formas posibles. Una vez terminado el proyecto, volvimos a probar todo para que no hubiera ningún error en el momento de la entrega.

Hicimos dos tipos de pruebas, caja negra y caja blanca. Las pruebas de caja negra ejercitan los requisitos funcionales desde el exterior del módulo, las de caja blanca están dirigidas a las funciones internas.

Para meter una imagen en una base de datos y que desde el código nos mostrase dicha imagen hicimos un par de pruebas de caja blanca. También para la verificación del login que nos daba problemas por la longitud del campo.

A su vez, las pruebas de caja negra que hemos realizado han sido sobre el crud, el login y el calendario sobre todo, solucionando los problemas de pedir citas de más de 30 minutos, las sesiones en el crud y destruir las sesiones en el crud.

| ID | Descripción prueba | Tipo | Subtipo | Clasificación | Prioridad | Input | Output | Estado | Fecha | Comentarios |
|----|------------------------------------|-------------|---------------------------|---------------|-----------|--|---|--------|------------|---|
| 1 | Dato de logueo válido | Caja Negra | Validación | Login | 1 | Nombre: alex Contraseña: asda | Acceso al calendario | | 02/05/2022 | Pendiente de verificar error |
| 2 | Dato de logueo inválido | Caja Negra | Validación | Login | 1 | Nombre: alex Contraseña: aaaa | Acceso al calendario | | 02/05/2022 | Necesita más pruebas |
| 3 | Solicitar cita en calendario | Caja Negra | Verificación | Calendario | 1 | Victor,victor@mail.com,Cortar pelo, 17:30-18:00 | Citado correctamente | | 26/04/2022 | |
| 4 | Solicitar cita en calendario | Caja Negra | Verificación | Calendario | 1 | Victor,victor@mail.com,Cortar pelo y Barba,17:00-18:00 | Citado correctamente | | 10/05/2022 | Aparentemente te lo guarda en la bbdd pero al acceder de |
| 5 | Imágenes sacadas de BBDD | Caja Blanca | Pruebas de funcionamiento | Front | 1 | <?php echo base64_encode(\$fila['imagen_servicio']); | No muestra la imagen | | 10/05/2022 | Falta conocimiento sobre el funcionamiento de los archivos BLOP |
| 6 | Imágenes sacadas de BBDD | Caja Blanca | Pruebas de funcionamiento | Front | 1 | src="data:image/jpeg;base64,<?php echo base64_encode(\$fila['imagen_servicio']);"> | Muestra la imagen | | 10/05/22 | |
| 7 | Sesiones en el crud | Caja negra | Pruebas de funcionamiento | Back | 1 | Datos de logueo como admin | No accede al crud.php | | 21/05/22 | La sesión no se guarda correctamente |
| 8 | Sesiones en el crud | Caja negra | Pruebas de funcionamiento | Back | 1 | Datos de logueo como admin | Accede al crud | | 25/05/22 | Añadimos el session_start() a todas |
| 9 | Sesiones en las funciones del crud | Caja negra | Pruebas de funcionamiento | Back | 1 | Datos de logueo como admin | La sesión no se queda guardada al salir de la página. | | 25/05/22 | Mal funcionamiento de las sesiones, falta |
| 10 | Sesiones en las funciones del crud | Caja negra | Pruebas de funcionamiento | Back | 1 | Datos de logueo como admin | La sesión se queda guardada al salir de la página. | | 25/05/22 | Añadimos la condición de la sesión en todas |
| 11 | Botón cerrar sesión crud | Caja negra | Pruebas de funcionamiento | Back | 1 | Datos de logueo como admin | Recibe por consulta la función session_destroy() | | 25/05/22 | Queda funcionando correctamente |
| 12 | Inicio de sesión | Caja Blanca | Verificación | Back | 1 | Datos de logueo como admin | No loguea | | 26/05/22 | No loguea |
| 13 | Inicio de sesión | Caja Blanca | Verificación | Back | 1 | Datos de logueo como admin | Loguea | | 26/05/22 | Tras cambiar la longitud del campo Password a varchar 255 funciona. |

Ilustración 7: Plan de pruebas

7. Conclusiones y mejoras futuras

Durante el proceso de desarrollo de la aplicación han ido apareciendo problemas, sobre todo en las principales funcionalidades, como que no aparecía bien la tabla del calendario, no funcionaba correctamente el login para el administrador en el crud... que en medida de lo posible está todo solucionado. Se han pensado en mejoras para la aplicación en el futuro, cómo:

- Una gestión de las bases de datos especializada y más exhaustiva para poder potenciar las campañas de marketing y el proceso de compra de los propios clientes.
- Adición de una sección de blog para que los clientes puedan darnos feedback sobre su experiencia y/o hacer reseñas.
- Desplegar en un servidor. Esto supondría el funcionamiento correcto del contacto, ya que al desplegar en servidor, se podrían mandar los mails.

8. Bibliografía

Estas son algunas de las fuentes de las que hemos obtenido información y ayuda para crear nuestro proyecto:

- tutorialrepublic.com/php-tutorial/php-mysql-login-system.php - [LOGIN PHP](#)
- obedalvarado.pw/blog/crud-datos-empleados-php-mysql-bootstrap/ - [CRUD en PHP](#)
- www.daniweb.com - [Imágenes en formato BLOB](#)
- www.youtube.com/watch?v=Y0cz_SV0X3Y - [PhpWebsTuts](#)
- getbootstrap.com/ - [Plantilla bootstrap](#)
- www.w3schools.com/ - [W3Schools](#)
- meeraacademy.com/php-sessions-example/ - [Sesiones Meeraacademy](#)
- https://validator.w3.org/#validate_by_input - [Accesibilidad](#)
- https://www.php.net/manual/en/function.password-verify.php - [Hash de password](#)
- https://www.php.net - [Otras consultas varias](#)
- Almacenamiento de imágenes en BLOB - [IES San Clemente](#)

9. Anexo

Adicionalmente a esta memoria, se adjuntan en formato digital todos los materiales utilizados para la elaboración del proyecto. Se pueden encontrar todos estos documentos dentro de la carpeta Anexos facilitada con el proyecto:

- Documento de plan de pruebas y costes → Plan de pruebas+Costes.xlsx
- Documento del Diagrama de GANTT → Diagrama de GANTT.xlsx
- Documento de Manual de uso → Manual de uso.pdf
- Documento de la memoria del TFG → TFG Peluqueria.pdf