

Curso 2022/2023

Moustache

Memoria del proyecto

Grupo 7
NOEL SARIÑENA



ÍNDICE

INTRODUCCIÓN	3
OBJETIVO (POR HACER).....	4
VIABILIDAD DEL PROYECTO.....	5
• Estudio de mercado	5
1. Aplicaciones / webs que cumplan una función similar y el valor añadido de la nuestra.	5
2. Encuesta para personas interesadas en el proyecto.....	6
• Project Charter	7
1. Propósito del proyecto.....	7
2. Resumen ejecutivo del proyecto.....	7
3. Resumen del proyecto	7
4. Descripción del proyecto.....	8
5. Condicionantes del proyecto	9
6. Equipo del proyecto	9
DIAGRAMA DE GANTT	10
• Diagrama original	10
• Diagrama modificado	11
• ¿Por qué se ha modificado?	11
DIAGRAMAS DE CLASES	12
• Diagrama de actividad.....	12
• Diagrama de secuencia	12
• Diagrama de casos de uso.....	13
• Diagrama de comunicación.....	13
• Diagrama de navegación	14
DIAGRAMAS DE BASES DE DATOS.....	15
1. Redacta los requisitos que debe tener la BBDD de tu proyecto. En la redacción de los requisitos, deberás detallar y justificar cada uno de los campos de las tablas. Después genera el diagrama E-R tal y como lo hiciste en la UF1 del M02. Si fuera el caso y en el diagrama E-R hubiera alguna casuística que no la contempla el diagrama, coméntalo en un apartado llamado NOTAS.....	15
2. Después de tener el diagrama E-R, pásalo a modelo relacional y confecciona las tablas de integridad y de dominio.....	17
3. Una vez realizado el apartado anterior comenta todas las instrucciones DDL, DML y DCL que vas a utilizar en el proyecto.	18

MOCK-UPS DE LAS INTERFICIES	20
1. Diseña los mock-up de la interfaz de tu aplicación. Puedes utilizar cualquiera de las herramientas que se han visto en M07-Diseño de interfaces.	20
2. Entrega un documento con las capturas y una explicación de la interfaz con la justificación de la usabilidad.	24
DOCUMENTACIÓN DE LA APLICACIÓN	33
• Documentación técnica.....	33
• Documentación de usuario (POR HACER)	43
CONCLUSIONES	44
WEBGRAFÍA.....	45

INTRODUCCIÓN

En el marco de mi formación en el Grado Superior en Desarrollo de Aplicaciones Multiplataforma, me complace presentar este proyecto final titulado 'Moustache'. Este trabajo representa el culmen de mis estudios y me ha brindado la oportunidad de aplicar los conocimientos y habilidades adquiridos a lo largo de mi trayectoria educativa.

El objetivo principal de este proyecto es crear un sistema de gestión de usuarios y reservas. Para lograrlo, he investigado en profundidad el ámbito de las aplicaciones web y sus extensiones. Durante el desarrollo del proyecto, he aplicado diversas técnicas y metodologías, y he utilizado herramientas especializadas como puede ser Bootstrap, jQuery o Ajax.

La relevancia de este proyecto radica en la aplicación de dicho software en el día a día de la empresa, ya que facilita muchas tareas que ralentizan el servicio de dicha barbería. A medida que avance en este informe, le invito a explorar los resultados obtenidos y las conclusiones a las que se han llegado.

Me gustaría agradecer el apoyo de mis familiares y amigos más cercanos por los consejos aportados para el desarrollo del proyecto. También quiero expresar mi gratitud a Moustache, la barbería que confió en mí para llevar a cabo este proyecto de grandes magnitudes para el final de grado. Su apoyo ha sido fundamental para la realización de este trabajo.

Sin más preámbulos, le invito a sumergirse en los detalles de este proyecto y a descubrir las contribuciones y el aprendizaje que ha generado. Espero que este trabajo sea una muestra de mi compromiso, dedicación y pasión por lo estudiado estos dos años, la programación.

OBJETIVO (POR HACER)

VIABILIDAD DEL PROYECTO

- Estudio de mercado

1. Aplicaciones / webs que cumplan una función similar y el valor añadido de la nuestra.

En este apartado voy a citar diferentes aplicaciones que realizan una función similar a alguna de mis aplicaciones desarrolladas en este proyecto. Los softwares que realizan reservas por internet van a ser los que voy a usar como criterio de similitud a mis aplicaciones, ya que Moustache tiene como objetivo final que los clientes puedan hacer una reserva en su barbería de confianza y de esta manera poder aliviar de gestiones de calendario a los empleados de esta misma.

Dicho esto, las aplicaciones que cumplen una función similar a Moustache son:

trivago.es

Trivago es una aplicación que permite comparar y hacer reservas de hoteles online. En relación a mi aplicación, esta web/app permite que cada establecimiento pueda tener y gestionar sus reservas en su calendario disponible.

Este software sería un referente bastante bueno para poder realizar mi proyecto ya que contiene muchos de los apartados que va a tener mi aplicación. Una de ellas es; según la opción que se haya marcado en el menú principal, el calendario dispone de unas opciones de reservas u otras.

Mi aplicación a diferencia de esta gestiona el calendario de reservas de una sola barbería, y volviendo a lo citado anteriormente, esta puede adaptarse al calendario según la opción seleccionada por el usuario. Es decir, siguiendo el ejemplo de a continuación:

- Corte de pelo: 45 min.
- Perfilado de barba: 20 min.
- Corte + perfilado: 1h.

Como valor añadido cabe destacar la diferencia de sectores al que va orientado este proyecto, y más aún, este proyecto puede ser escalable. Con esto quiero decir que dicho proyecto puede mejorarse e implementar diferentes locales dentro del mismo programa.



SegurCaixa Adeslas

A diferencia de la aplicación anterior, SegurCaixa Adeslas es una aplicación/web que gestiona todo tipo de citas al médico o consultas relacionadas con la salud del usuario. Esta aplicación permite según la póliza que tenga el usuario elegir un servicio u otro. Cabe destacar que por tener una póliza que no cubra ese servicio no se le priva de ese mismo, sino que se deberá pagar una cierta cantidad que otras pólizas ya cubren.

Volviendo al software de la aseguradora, esta dispone de un menú de opciones que, según la cita que se quiera pedir, el usuario podrá escoger alguna hora disponible en el calendario de dicha consulta o establecimiento.

Al igual que Tripadvisor, este software puede hacer reservas dependiendo del servicio elegido dentro del sistema, y en relación a mi software, los tres cumplimos con el mismo objetivo común: Poder ofrecer la posibilidad al usuario de generar reservas o citas a los establecimientos ofrecidos por el sistema.

Como punto final, el valor añadido que tiene mi aplicación a diferencia de SegurCaixa Adeslas viene siendo lo mismo que con Trivago, un software escalable y orientado a un sector totalmente diferente.



2. Encuesta para personas interesadas en el proyecto.

El formulario generado para este ejercicio es el siguiente:

<https://forms.gle/TX6qi7CHSUukUBLRA>

https://docs.google.com/forms/d/e/1FAIpQLScorilfbj_TTIWSuNg26dN9clAzA2RKcMyQL9E_Ow8hoVHp2Q/viewform?usp=sf_link

- Project Charter

1. Propósito del proyecto

Moustache gestionará las reservas de una barbería. Para eso se va a crear una web y, si es posible, una aplicación de Android para esta misma.

2. Resumen ejecutivo del proyecto

Se va a desarrollar una aplicación web que gestione el calendario de la barbería y, si el plazo de tiempo lo permite, una aplicación Android para usuarios que cuenten con ese sistema.

El proyecto se va a desarrollar en cinco meses si todos los objetivos temporales se van cumpliendo en sus correspondientes fechas y su costo va a ser de 1.701€ (9€/h * 9h la semana * 21 semanas = 1.701€).

3. Resumen del proyecto

El proyecto fué propuesto por la misma barbería, la cual ya tenía dentro de sus planes digitalizar el proceso de reservas de sus citas. Yo, Noel Sariñena, propuse de intentar realizar este trabajo.

Este software va a estar separado en dos partes;

- El primero y más importante de ambos va a ser una aplicación web que cuente con secciones como por ejemplo “página principal”, “quienes somos”, “servicios” etc. Lo más importante esta web será la sección de “haz tu reserva”. Esta meterá al usuario en un proceso de registro para su reserva:
- Primero de todo se le va a mostrar un menú con los diferentes servicios con los que cuentan (cada uno tiene un precio y un espacio dentro del calendario diferente).
- A continuación, se mostrará un calendario con las diferentes horas disponibles dentro de sus horarios. El usuario escogerá una hora a su gusto donde su tipo de servicio quepa.
- Tras ello se le pedirá la información para el pago (pendiente de revisión). Una vez aportados los campos, se redirigirá a la web oficial de la emisora del pago para que se complete la paga y señal.

Se ha llegado a una decisión con la barbería para que cada reserva que se efectúe se deba realizar una paga y señal, así se evita que una persona pueda rellenar todas las horas dentro de la peluquería y dejar sin horas a los trabajadores.
- Tras el pago, se enviará un correo con la información de la reserva al usuario.

- Por otra parte, pero no menos importante está la aplicación Android. Esta aplicación será la versión portátil de la web. Si esta acaba teniendo un desarrollo correcto, se pondrá a la disposición de todo el mundo en la Play Store.

4. Descripción del proyecto

4.1. Meta y objetivos

Meta	Objetivos
Realizar una aplicación web que gestione las reservas de una barbería, teniendo en cuenta la gestión de usuario, pagos, y reservas asimismo	<ul style="list-style-type: none"> • Realizar una web con una gestión de usuarios. • Implementar las funcionalidades necesarias para que el usuario pueda navegar bien por el sitio (usabilidad). • Crear un calendario que consulte en vivo las reservas disponibles en la base de datos. • Gestionar pagos entre cliente/barbería. • Crear un panel (no visible para los clientes) donde el administrador pueda consultar sus reservas en la barbería. • SI DA TIEMPO: Añadir gestión más profunda a las reservas (cancelar, posponer, ...).

4.2. Requerimientos

Requerimientos obligatorios (<i>must have</i>)	Requerimientos deseables (<i>nice to have</i>)
“CRUD ¹ ” de los usuarios	Paga y señal de las reservas
Correcto funcionamiento del calendario interno en la web.	
Sistema de gestión de reservas: <ul style="list-style-type: none"> - Creación de reservas. - Lectura de reservas. 	

4.3. Presupuesto estimado

Hito	Presupuesto
Mano de obra	9€/h * 9h la semana * 21 semanas = 1.701€
Adquisición de un dominio “.es” por 12 meses.	Gratis primer año + 10.99€ el año.

¹ Create, Read, Update, Delete (Crear, Leer, Modificar y Borrar).

5. Condicionantes del proyecto

5.1. Suposiciones del proyecto

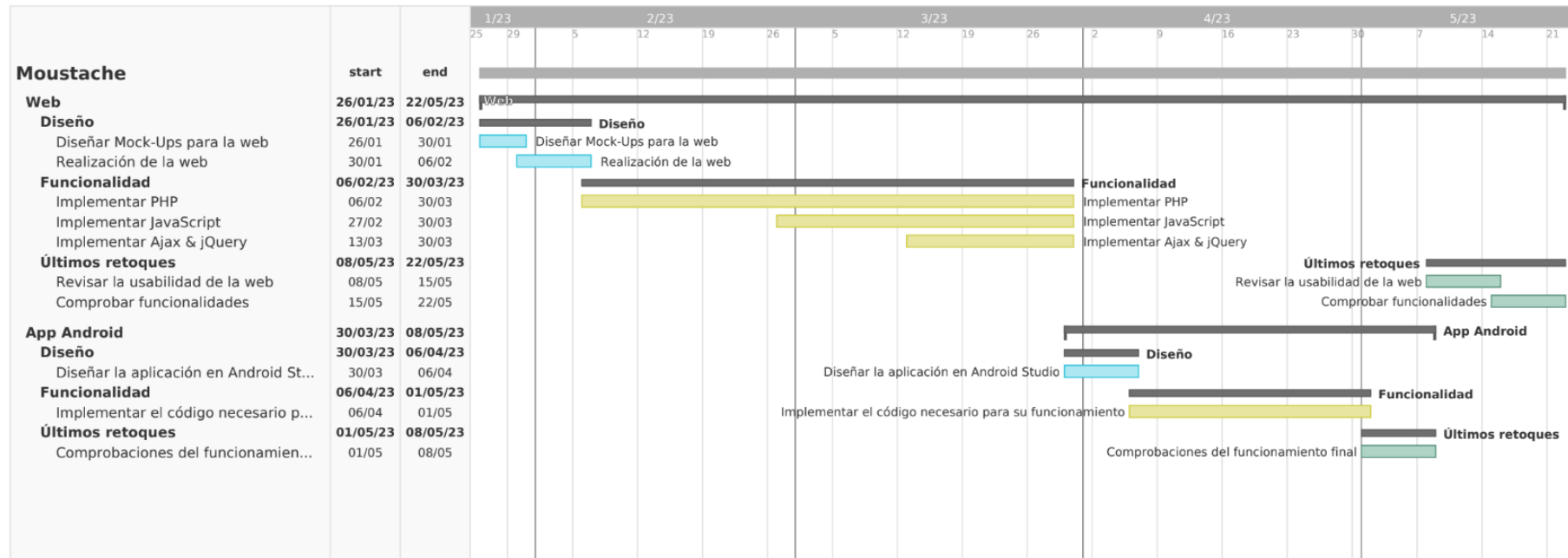
- Librería para crear, de manera visual, un calendario en la web
- Base de datos online para usar tanto en la web como en la aplicación.

6. Equipo del proyecto

Nombre	Cargo	Responsabilidad
Noel Sariñena Varela	Jefe de proyecto, diseñador y programador	Llevar a cabo todo el proyecto.

DIAGRAMA DE GANTT

- Diagrama original



- Diagrama modificado²



- ¿Por qué se ha modificado?

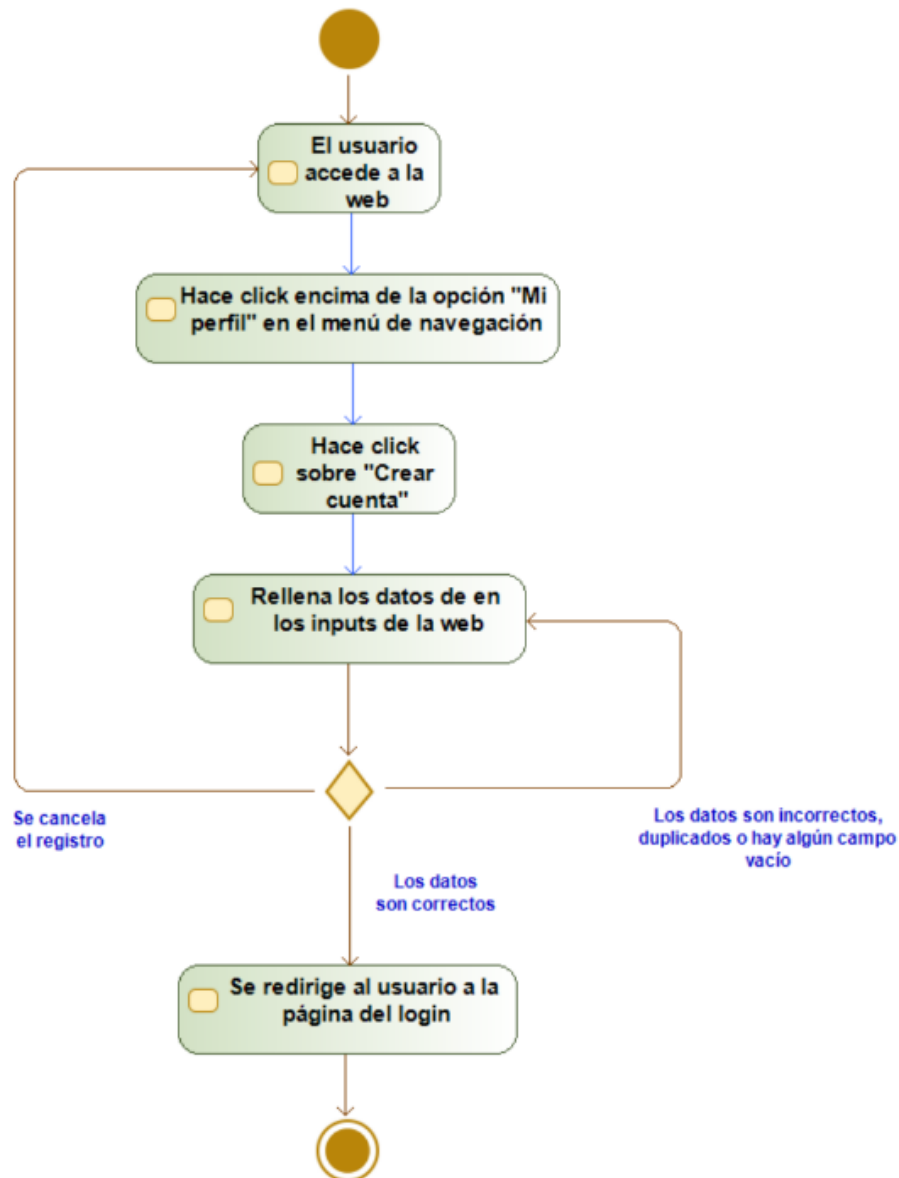
El documento original ha sido modificado ya que, como he explicado en el apartado de “Objetivo”, inicialmente se tenía en mente crear una versión Android del proyecto, pero por dichos motivos, no se ha podido realizar la aplicación Android y se ha mejorado lo máximo posible la web.

El tiempo que iba a ocupar dicho desarrollo de aplicación ha acabado en mejora de funcionalidad del sitio web.

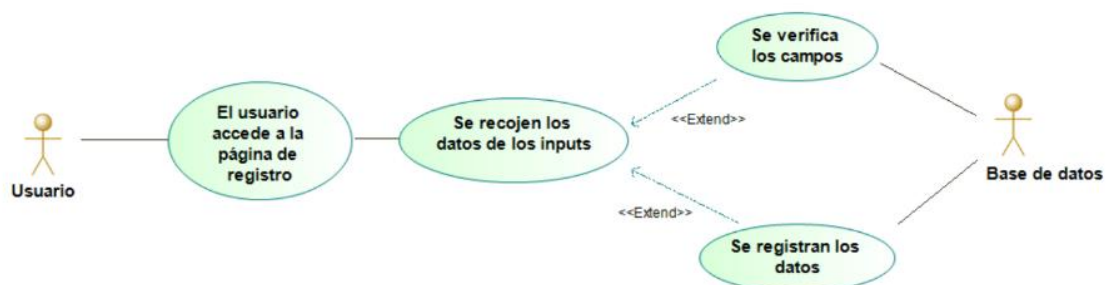
² Si se hace clic sobre el título, se puede descargar la versión PDF del diagrama.

DIAGRAMAS DE CLASES

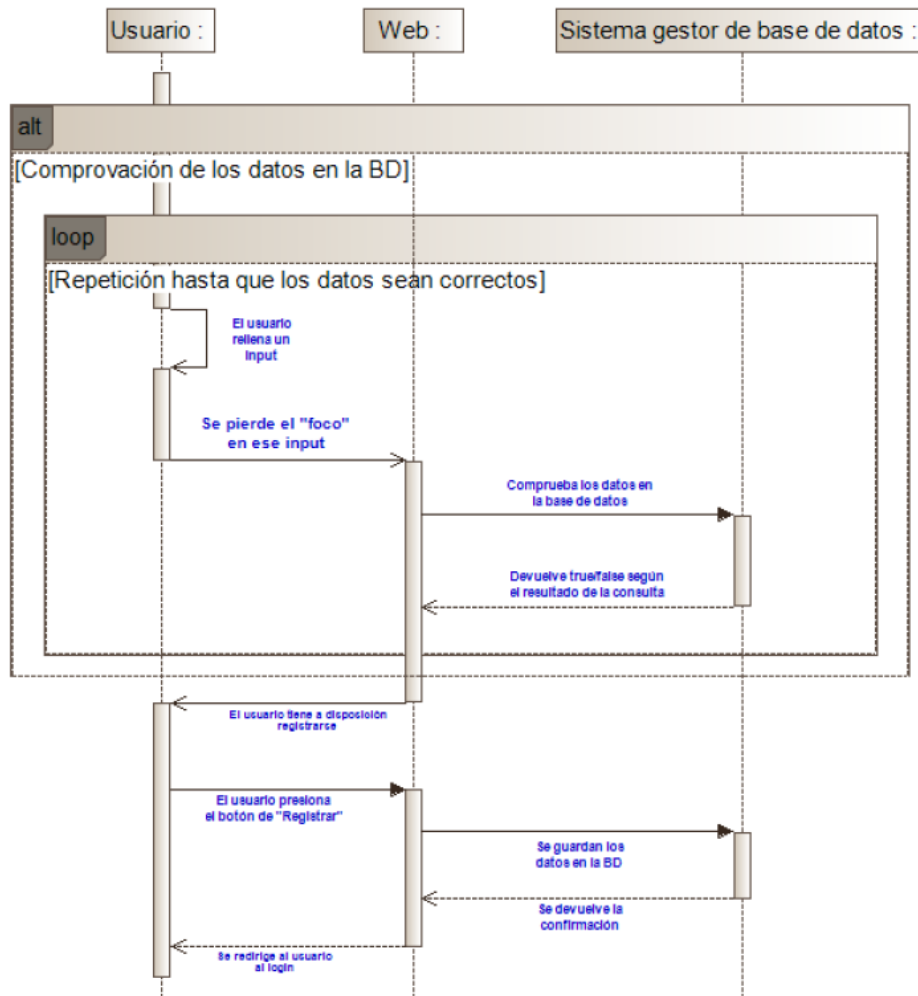
- Diagrama de actividad



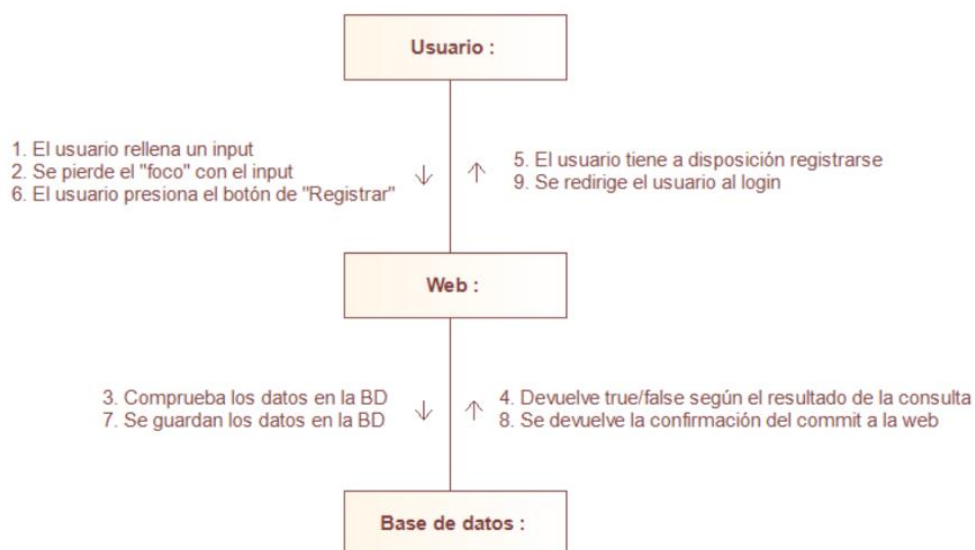
- Diagrama de secuencia



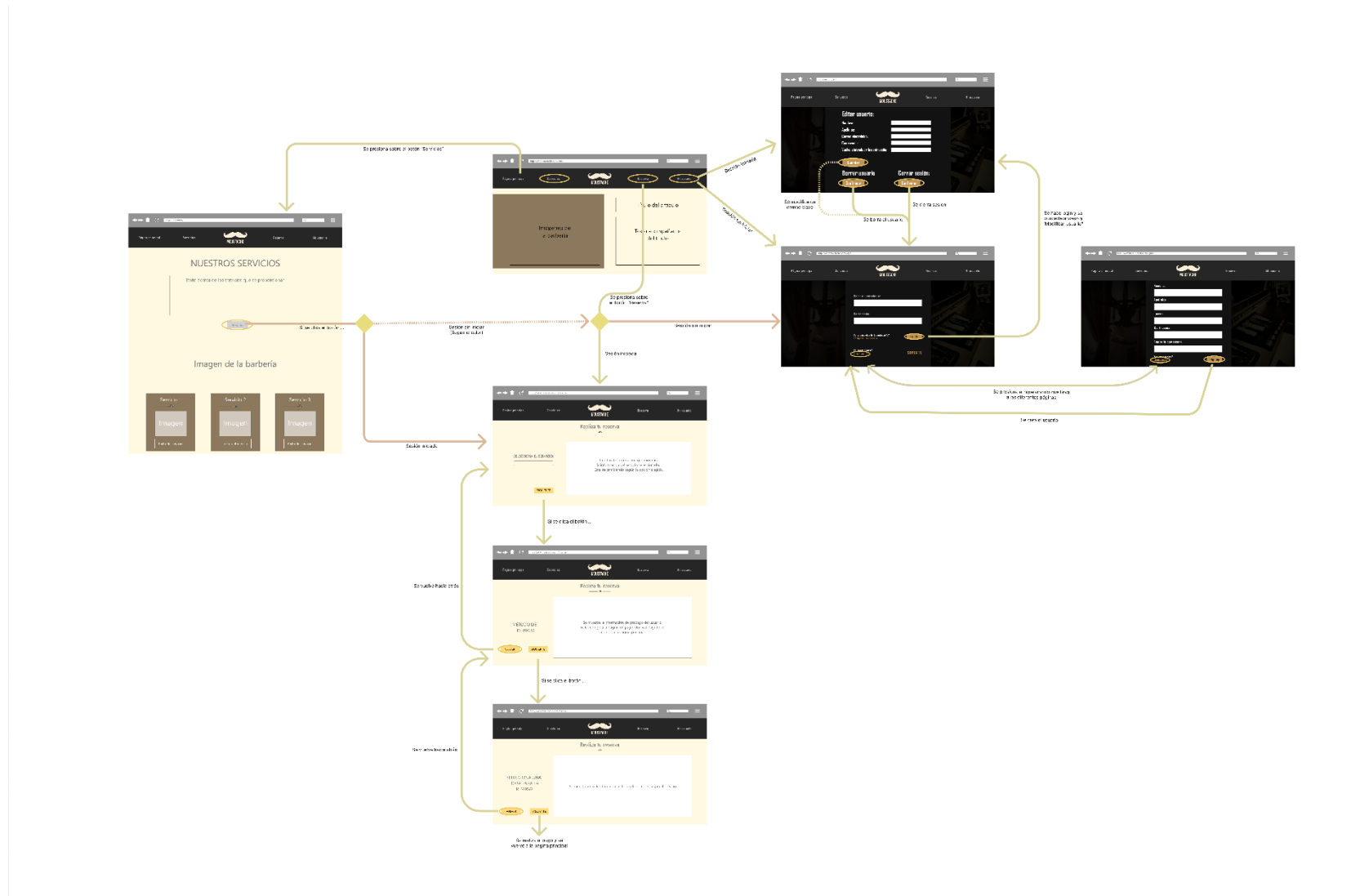
- Diagrama de casos de uso



- Diagrama de comunicación



- Diagrama de navegación



Link para ver la imagen a mayor resolución: <https://i.imgur.com/eUXrlDr.png>

DIAGRAMAS DE BASES DE DATOS

1. Redacta los requisitos que debe tener la BBDD de tu proyecto. En la redacción de los requisitos, deberás detallar y justificar cada uno de los campos de las tablas. Después genera el diagrama E-R tal y como lo hiciste en la UF1 del M02. Si fuera el caso y en el diagrama E-R hubiera alguna casuística que no la contempla el diagrama, coméntalo en un apartado llamado NOTAS.

La base de datos para el proyecto se llama “moustache” y cuenta con tres tablas para el almacenaje de sus datos, dos de ellas tienen las mismas columnas, pero separan informaciones diferentes:

❖ Primera tabla; “Users”:

Esta tabla guarda todos los datos del usuario a la hora de crear su cuenta. Todos los campos de esta son “Not null”, por lo que siempre se deberán rellenar para la inserción de los datos en esta. La tabla cuenta con los siguientes campos:

- *ID*:
Este campo asigna un número al usuario de manera automática a la hora de añadir el registro a la base de datos, por lo que este campo es Integer(255), Autoincremental y clave primaria. Este campo sirve para que el backend de la web/aplicación pueda hacer operaciones sin acudir a ningún otro campo del usuario.
- *Name & Surname*:
Estos dos campos guardan el nombre y el apellido respectivamente del usuario. Estos campos son Varchar2(50-100 respectivamente). El campo *Name* se usa para mostrar el nombre del usuario en el frontend.
- *Email*:
Este campo guarda el email del usuario, por lo que este será un Varchar2(200). Este campo se quiere usar para la confirmación de modificación de datos de la BD. También se quiere usar este para enviar promociones e información acerca de la barbería.
- *Password*:
Este campo guarda la contraseña del usuario en formato hash tipo “SHA-512”. Este campo es Varchar2(200).

❖ Segunda y tercera tabla; “reservasPelo” y “reservasBarba”:

Estas tablas van a guardar toda la información acerca de las reservas que se realicen a los usuarios. Igual que la tabla anterior, todos los campos en esta son “Not null”. Los campos son los siguientes:

○ *IDReserva*:

Este campo asigna un número a la reserva de manera automática a la hora de añadir el registro a la base de datos, por lo que este campo es Varchar(255) y clave primaria. Este campo nos servirá para distinguir las reservas entre ellas, si queremos acceder a alguna de estas será tan fácil como acudir al ID en concreto.

○ *UserID*:

Este campo se relaciona la tabla de “users” como clave foránea, por lo que este campo adapta los valores del campo “ID” de la otra tabla; Integer(255). Este campo se usará para que los empleados de la barbería puedan obtener la información completa del usuario que ha realizado la reserva. También servirá para realizar operaciones en el backend del software.

○ *FechaReserva*:

Este campo guarda la fecha en la que se ha realizado la reserva. El formato de este campo será Datetime.

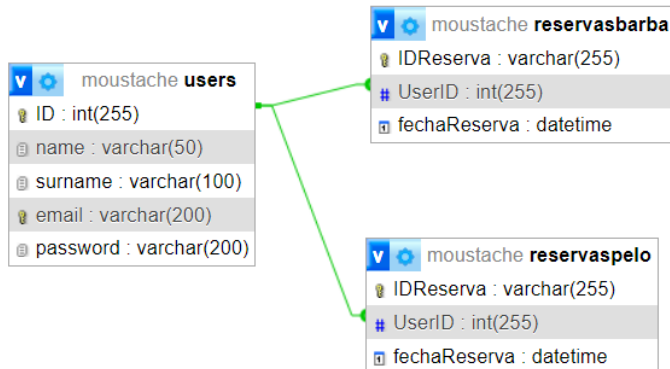
El diagrama entidad relación (E-R) será similar al siguiente:

USUARIO: ID*, Name, Surname, Email, Password
RESERVAS: IDReserva*, UserID, FechaReserva



2. Después de tener el diagrama E-R, pásalo a modelo relacional y confecciona las tablas de integridad y de dominio.

❖ Modelo relacional:



USERS: ID, name, surname, email, password

RESERVASPELO: IDReserva, ID.USERS, fechaReserva

RESERVASBARBA: IDReserva, ID.USERS, fechaReserva

❖ Tabla de integridad

USERS						
Nombre	Descripción	Dominio	P	F	Referencia	R
ID	ID del usuario	DID	S	N	-	S
name	Nombre del usuario	DNOM	N	N	-	S
suraname	Apellidos del usuario	DNOM	N	N	-	S
email	Email del usuario	DEMA	N	N	-	S
password	Contraseña del usuario	DNOM	N	N	-	S

RESERVASPELO						
Nombre	Descripción	Dominio	P	F	Referencia	R
IDReserva	ID de la reserva	DID	S	N	-	S
UserID	ID del usuario que ha generado la reserva	DID	N	S	ID.USERS	S
fechaReserva	Día y hora de la reserva	DFEC	N	N	-	S

RESERVASBARBA						
Nombre	Descripción	Dominio	P	F	Referencia	R
IDReserva	ID de la reserva	DID	S	N	-	S
UserID	ID del usuario que ha generado la reserva	DID	N	S	ID.USERS	S
fechaReserva	Día y hora de la reserva	DFEC	N	N	-	S

❖ Tabla de dominio

TABLA DE DOMINIO				
Nombre	Base	N.Caracteres	N.Decimales	Valores
DNOM	Caracteres	255		Letras
DID	Enteros		0	Números
DEMA	Caracteres	255		Alfanumérico
DFEC	Enteros		0	Alfanumérico

3. Una vez realizado el apartado anterior comenta todas las instrucciones DDL, DML y DCL que vas a utilizar en el proyecto.

❖ Sentencias DDL

- Creación de la base de datos “Moustache”:

```
CREATE DATABASE IF NOT EXISTS `moustache` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

```
USE `moustache`;
```

- Creación de la tabla “reservasBarba”:

```
CREATE TABLE `reservasbarba` (
  `IDReserva` varchar(255) NOT NULL,
  `UserID` int(255) NOT NULL,
  `fechaReserva` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- Creación de la tabla “reservasPelo”:

```
CREATE TABLE `reservaspelo` (
  `IDReserva` varchar(255) NOT NULL,
  `UserID` int(255) NOT NULL,
  `fechaReserva` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- Creación de la tabla “users”:

```
CREATE TABLE `users` (
  `ID` int(255) NOT NULL,
  `name` varchar(50) NOT NULL,
  `surname` varchar(100) NOT NULL,
  `email` varchar(200) NOT NULL,
  `password` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- Añadir la clave primaria y una clave única a “users”:

```
ALTER TABLE `users`  
  ADD PRIMARY KEY (`ID`),  
  ADD UNIQUE KEY `email` (`email`);
```

- Añadir clave primaria(1) y clave foránea(2) a “reservasPelo”:

```
(1)  
ALTER TABLE `reservaspelo`  
  ADD PRIMARY KEY (`IDReserva`),  
  ADD KEY `reservaspelo_ibfk_1` (`UserID`);
```

```
(2)  
ALTER TABLE `reservaspelo`  
  ADD CONSTRAINT `reservaspelo_ibfk_1` FOREIGN KEY (`UserID`) REFERENCES `users` (`ID`)  
  ON DELETE CASCADE ON UPDATE CASCADE;
```

- Añadir clave primaria(1) y clave foránea(2) a “reservasBarba”:

```
(1)  
ALTER TABLE `reservasbarba`  
  ADD PRIMARY KEY (`IDReserva`),  
  ADD KEY `reservasbarba_ibfk_1` (`UserID`);
```

```
(2)  
ALTER TABLE `reservasbarba`  
  ADD CONSTRAINT `reservasbarba_ibfk_1` FOREIGN KEY (`UserID`) REFERENCES `users` (`ID`)  
  ON DELETE CASCADE ON UPDATE CASCADE;
```

❖ Sentencias DML

```
INSERT INTO `users` (`ID`, `name`, `surname`, `email`, `password`) VALUES  
(0, 'Noel', 'Sariñena Varela', 'noel@gmail.com',  
'c1f3270666a17ea7ac53fca4170b55181081db37fdd7b30350138b34c36b59a8'),  
(12, 'Delia', 'Soria', 'delia@gmail.com',  
'c1f3270666a17ea7ac53fca4170b55181081db37fdd7b30350138b34c36b59a8');
```

MOCK-UPS DE LAS INTERFICIES

1. Diseña los mock-up de la interfaz de tu aplicación. Puedes utilizar cualquiera de las herramientas que se han visto en M07-Diseño de interfaces.

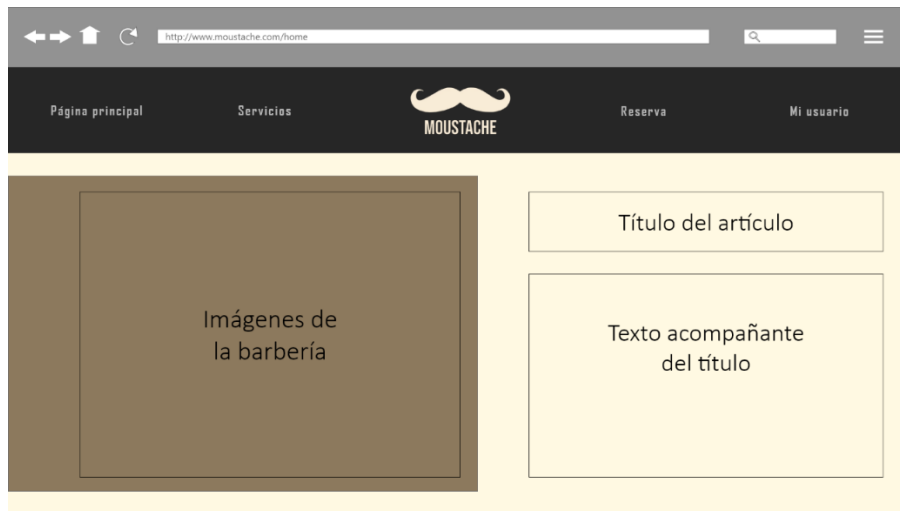
❖ Alta

The mockup shows a web browser window with the URL <http://www.betterpresets.com/signup>. The page has a dark theme with a navigation bar at the top containing links: [Página principal](#), [Servicios](#), [Reserva](#), and [Mi usuario](#). The Moustache logo, featuring a yellow mustache, is centered in the navigation bar. The main content area has a dark background with a central white form for registration. The form fields are labeled: **Nombre:**, **Apellidos:**, **Correo:**, **Contraseña:**, and **Repita la contraseña:**. Below the form, there is a link [Ya tienes cuenta? Inicia sesión](#) and a **Sign up** button. The background of the page features a dark, artistic image of a mustache and a keyboard.

❖ Inicio de sesión

The mockup shows a web browser window with the URL <http://www.moustache.com/login>. The page has a dark theme with a navigation bar at the top containing links: [Página principal](#), [Servicios](#), [Reserva](#), and [Mi usuario](#). The Moustache logo, featuring a yellow mustache, is centered in the navigation bar. The main content area has a dark background with a central white form for login. The form fields are labeled: **Correo electrónico:** and **Contraseña:**. Below the form, there is a link [No te acuerdas de la contraseña? Recuperar contraseña](#) and a **Log in** button. At the bottom, there is a link [No tienes cuenta? Crear cuenta](#) and a **SOPORTE** button. The background of the page features a dark, artistic image of a mustache and a keyboard.

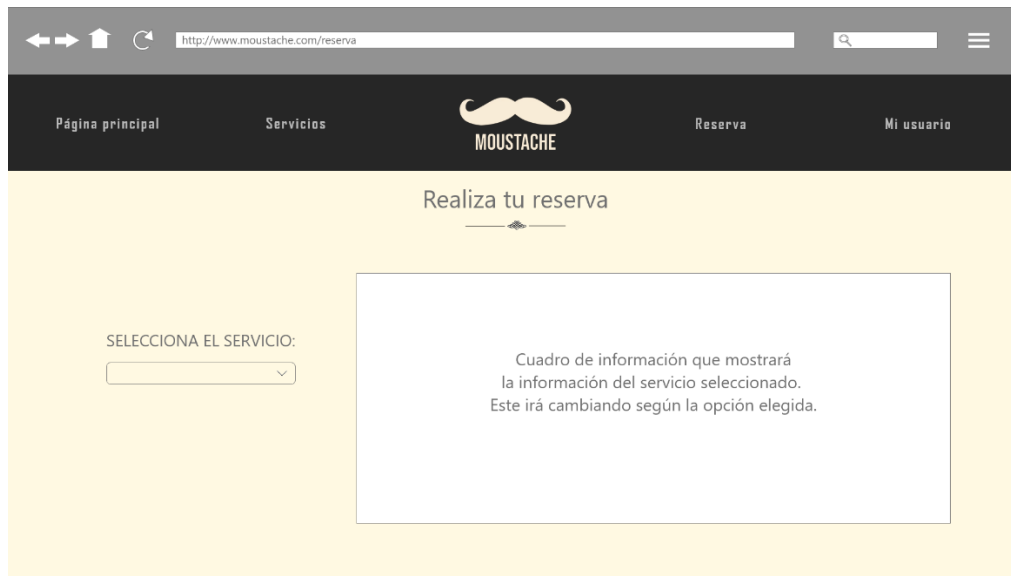
❖ Página principal



❖ Servicios



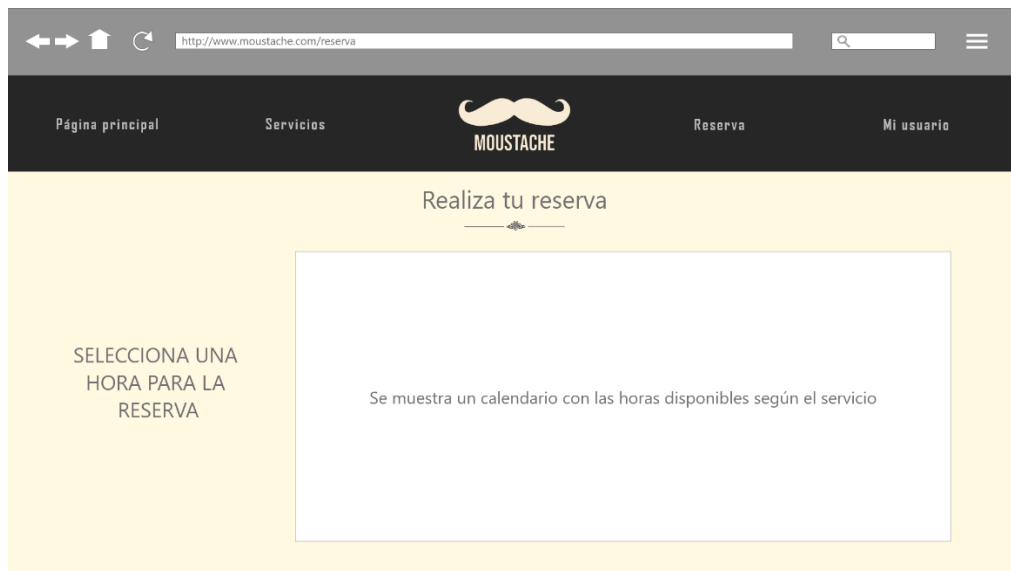
❖ Reserva



Realiza tu reserva

SELECCIONA EL SERVICIO:

Cuadro de información que mostrará la información del servicio seleccionado. Este irá cambiando según la opción elegida.



Realiza tu reserva

SELECCIONA UNA HORA PARA LA RESERVA

Se muestra un calendario con las horas disponibles según el servicio

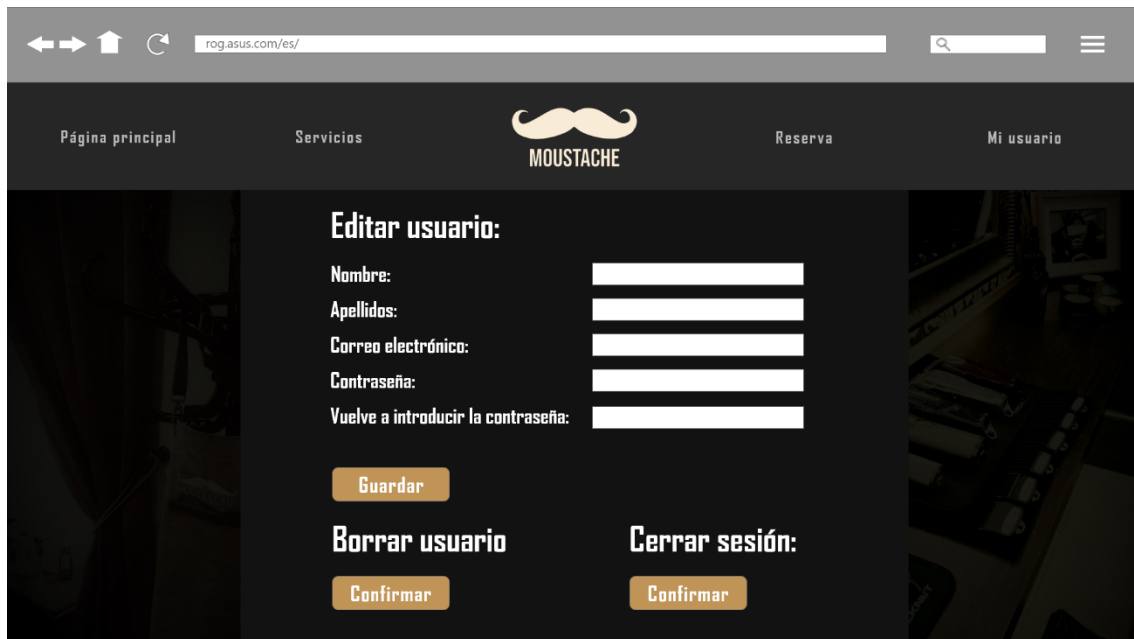


Realiza tu reserva

MÉTODO DE PRE-PAGO

Se muestra la información de prepago del usuario. Se le redirige a la página de pago. Una vez pagado se redirige a la página principal.

❖ Modificar datos



The screenshot shows a web browser at the URL `rog.asus.com/es/`. The website has a dark theme with a navigation bar containing links: [Página principal](#), [Servicios](#), [Reserva](#), and [Mi usuario](#). The central logo is a yellow mustache above the word **MOUSTACHE**. The main content area is titled **Editar usuario:** and contains the following form fields:

- Nombre:
- Apellidos:
- Correo electrónico:
- Contraseña:
- Vuelve a introducir la contraseña:

Below the form fields are three buttons: **Guardar**, **Borrar usuario**, and **Cerrar sesión:**. The **Borrar usuario** and **Cerrar sesión:** buttons have a **Confirmar** sub-button below them.

2. Entrega un documento con las capturas y una explicación de la interfaz con la justificación de la usabilidad.

- Patrones generales.



Figura 1.1

Para mi página he optado por una tipología de letras diferente a Sans-Serif, pero “amistosa” para la vista. He distribuido el menú de la que se encuentra en la parte superior ya que este no contiene muchas redirecciones a diferentes páginas y el logo podía caber en el medio.



Figura 1.2

En primer lugar, el logo se encuentra centrado en medio de todo el menú de redirecciones. Es por eso que he optado por un diseño sencillo pero minimalista, que sea claro y accesible desde todas las partes de la página.

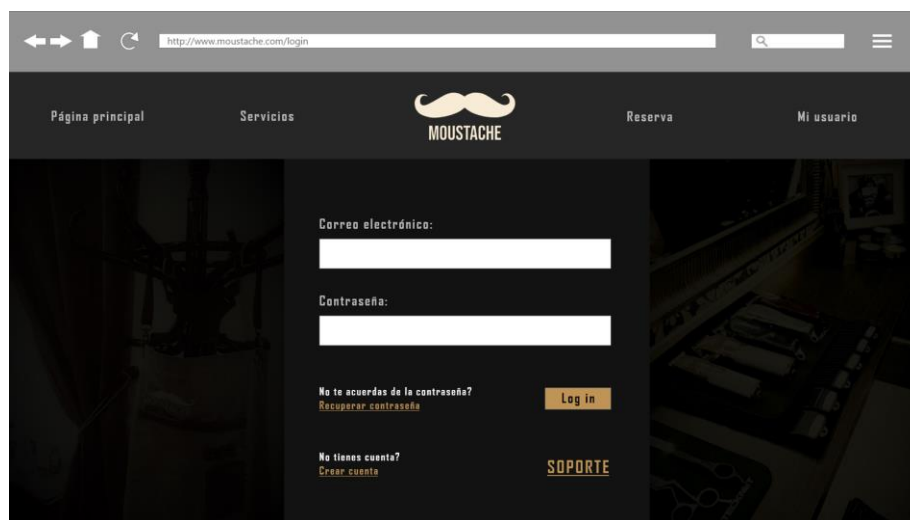


Figura 1.4

El color usado para esta web es un marrón oscuro y un color carne. Con esto busco que la web se vea más elegante y atractiva a la vista del usuario, ya que contrastar con los colores de esta paleta, desde mi punto de vista, da más tranquilidad. A esta paleta de colores la he nombrado “paleta otoñal”; contiene todo tipo de colores ocre que hacen que la página se vea “clásica” y formal.

- Alta a la web

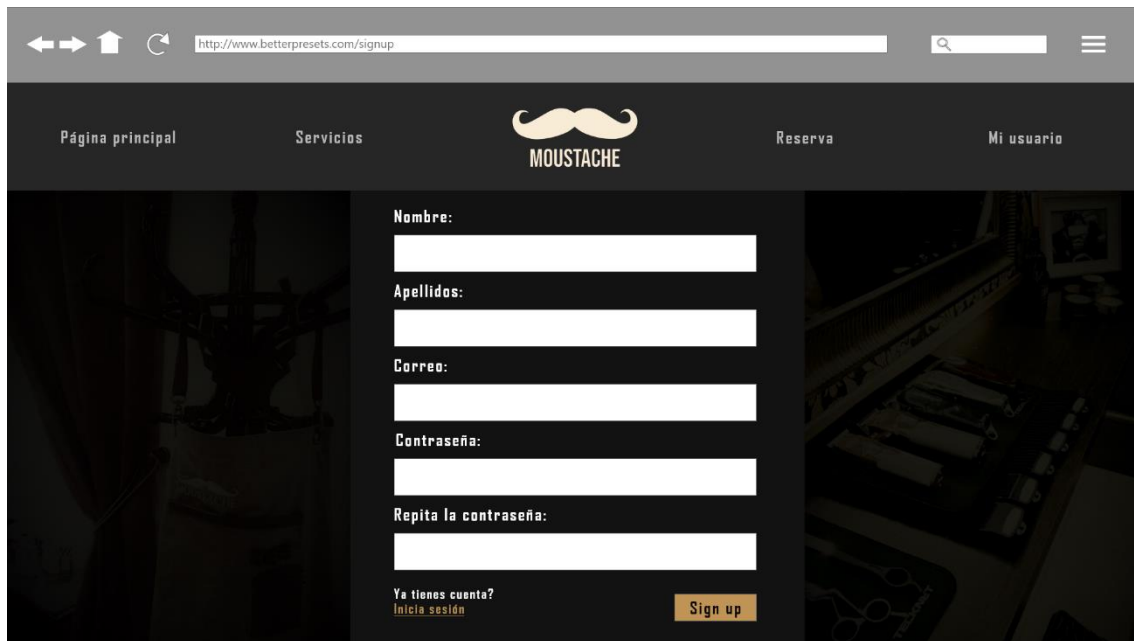
The image shows a web browser window with the URL 'http://www.betterpresets.com/signup'. The website has a dark theme with a navigation bar at the top containing links for 'Página principal', 'Servicios', 'Reserva', and 'Mi usuario'. The central logo is a yellow mustache above the word 'MOUSTACHE'. The main content area features a sign-up form with five white input fields labeled 'Nombre:', 'Apellidos:', 'Correo:', 'Contraseña:', and 'Repita la contraseña:'. Below the form, there is a link 'Ya tienes cuenta? Inicia sesión' and a yellow 'Sign up' button. The background of the form area is a dark, artistic image of a mustache.

Figura 2.1

El usuario deberá rellenar los campos que se muestran en la figura 2.1 para darse de alta en la web. Una vez rellenados los campos, dispone del botón “Sign up” que da de alta al usuario en caso de que toda la información introducida sea correcta y concuerde con los tipos de datos de la base de datos.

En caso de que ya tenga una cuenta dispone de un botón que redirige a la interfaz de inicio de sesión.

- Log In

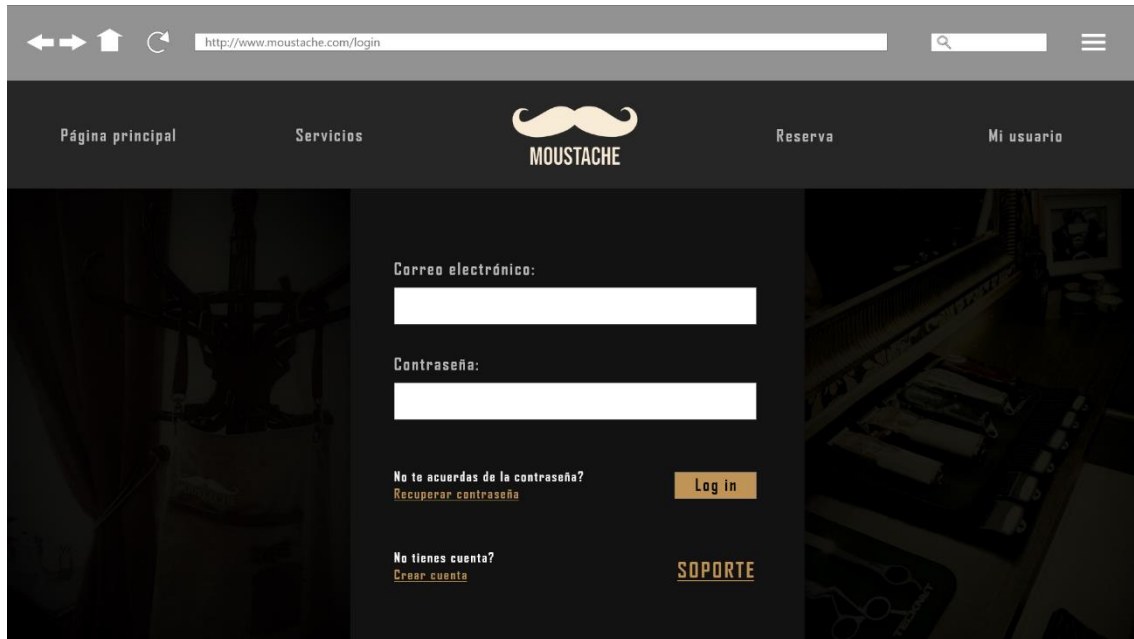


Figura 3.1

El inicio de sesión dispone de diferentes apartados:

❖ Campos de inicio de sesión:

El usuario introducirá el correo que introdujo a la hora de registrarse en el portal. A si mismo, introducirá la contraseña con la que ha vinculado a su cuenta.

Después de eso, hará clic sobre el botón Log In y accederá al portal, donde ya por fin podrá acceder a la página de reservas y modificación de usuario.

❖ Recuperar contraseña:

Habrà un botón que redirigirá a una interfaz alternativa donde podrá recuperar la contraseña de su cuenta.

❖ Crear cuenta:

Si el usuario hace clic sobre este botón, se redirigirá a la interfaz anterior para darse de alta.

❖ Soporte:

El usuario podrá ponerse en contacto con los desarrolladores de la página en caso de tener errores para su inicio de sesión.

Los botones “Recuperar contraseña”, “Crear cuenta” y “Soporte” tienen la misma tipología de letras; Naranja resaltante y subrayado. El subrayado se le aplica debido a que la mayoría de enlaces que redirigen a diferentes páginas suelen estar subrayados para que el usuario sepa que dicha/s palabra/s te redirigen a un sitio externo.

- Página principal

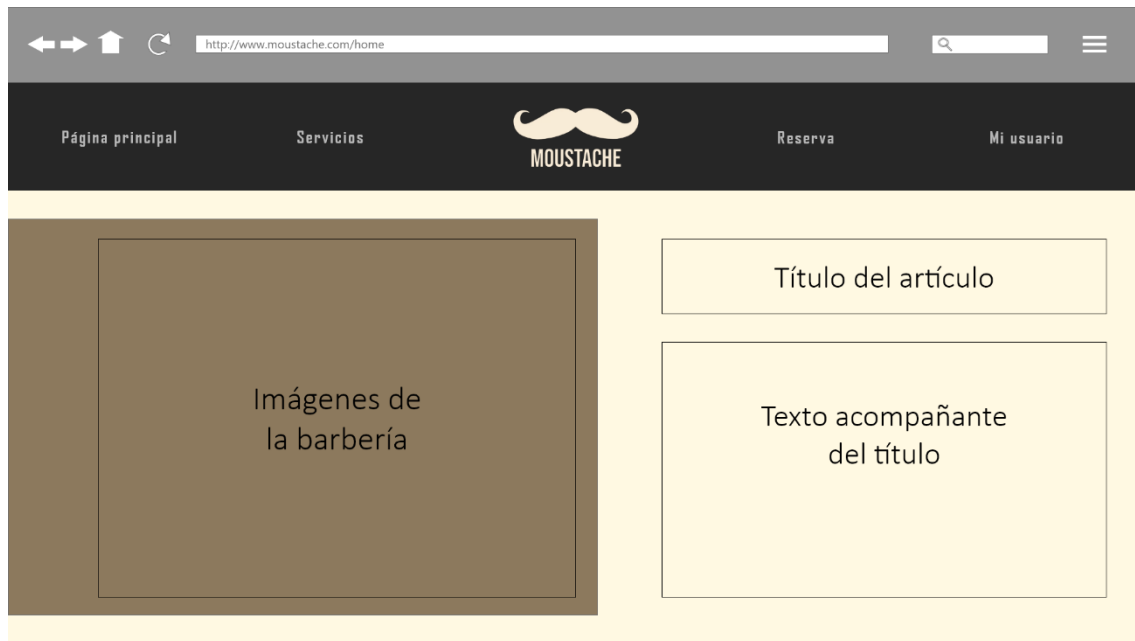


Figura 4.1

Habrán dos columnas a lo largo de la web, pero el contenido entre ellas se irá entrelazando. Es decir, en el primer artículo, las imágenes se mostrarán a la izquierda como se puede observar en la figura 4.1, pero el artículo siguiente a este, podrá tener las imágenes a la derecha. Esto se hace principalmente para que la web no sea tan monótona a la vista.

Esta página está distribuida en 2 partes:

❖ Primera parte: Las imágenes

Las imágenes irán sobre un recuadro marrón oscuro para que el contraste de estas con el texto sea más “amistoso”.

❖ Segunda parte: El texto

En la parte superior tendremos el título / frase que describa el apartado. Este título estará de color negro y con tipografía “Karsten” ya que destaca por encima de los demás textos que hay por pantalla, es donde queremos focalizar la atención del usuario.

Las características y el contenido del paquete serán textos normales con tipología “Habanera Light”.

Con esta web buscamos introducir al usuario a la barbería y ganarnos la confianza necesaria para que este haga su reserva en ella que, al fin y al cabo, es la finalidad de todo.

- Servicios

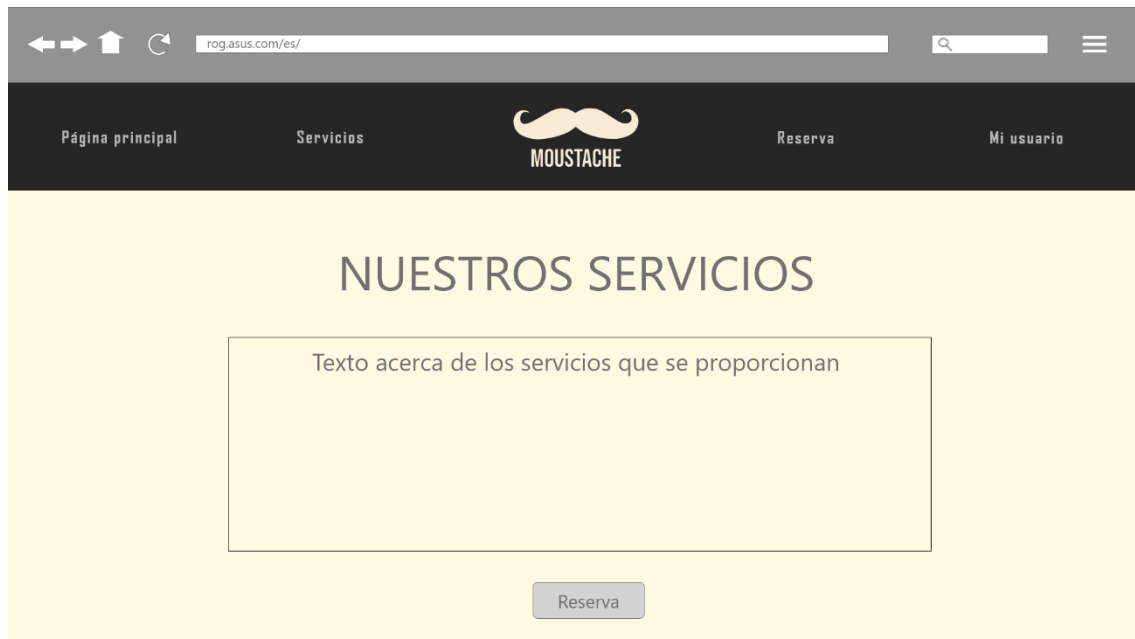


Figura 5.1



Figura 5.2

Cuando el usuario acceda a esta página va a encontrarse con una presentación sutil del apartado teniendo de primera impresión un título que lo situé dentro de nuestro portal web, junto a un texto acerca de los objetivos de la barbería y una imagen solapada abajo suya.

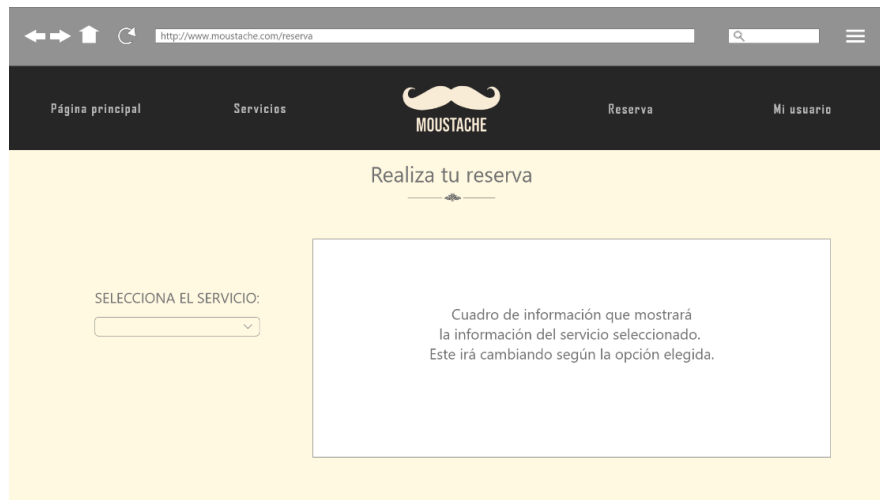
A parte de los elementos especificados anteriormente, nos encontramos que, debajo de todo esto tenemos un botón que nos redirige a la pestaña de “Reserva”, ya que la finalidad de la web es que el usuario realice una reserva en la barbería.

Más abajo nos encontraremos con una colección de cartas donde se puedan observar todos los servicios que presenta la barbería. Esta tendrá un título que resaltará a la vista, más abajo una imagen que resuma el título y un pequeño texto con la información necesaria para resumir el servicio. (*Figura 5.2*)

(Por añadir): Se está planteando de poner un botón que redirija al usuario a la pestaña de reserva habiendo seleccionado ya ese servicio.

- Reserva

Para este apartado contamos con los siguientes 3 mock-ups:



(Figura 6.1)



(Figura 6.2)



(Figura 6.3)

En la parte superior de la pestaña “Reserva” nos vamos a encontrar con un título estático que ayudará al usuario a situarse dentro de la web. Bajo este mismo título nos vamos a encontrar con un “separador” o subrayado de texto que le dará elegancia al propio título.

Esta página está dividida en 3 partes:

❖ Primera página: Selección de servicio.

En la primera pestaña contamos con un desplegable donde se van a mostrar todos los servicios que dispone la peluquería, justo arriba suyo nos vamos a encontrar un título que introducirá a la sección.

A la parte derecha, nos encontraremos con un recuadro que cambiará el contenido según la opción seleccionada dentro del formulario.

❖ Segunda página: Hora de reserva / Calendario de disponibilidad.

Esta parte está aún por decidir o pendiente de modificaciones ya que, como desarrollador junior, no tengo los suficientes conocimientos para llevar a cabo la idea que tengo en mente.

El planteamiento visual de esta interfaz es presentar al usuario un calendario donde pueda elegir una hora nada más haciendo clic sobre una de las franjas que esté disponible en este.

❖ Tercera página: Datos de reserva

Esta tercera página aún está por desarrollarse, pero la finalidad de esta es redirigir al usuario a la entidad bancaria de la barbería donde hará un prepago de su reserva. Para eso se han pensado en dos diseños para esta página. Primero de todo, a la izquierda del recuadro de petición de información habrá un título con la tipografía “Karsten”, bastante elegante a la vista. Por otra parte, a la derecha del este mismo nos encontramos con dos opciones diferentes:

- Un formulario para rellenar los campos necesarios para proceder con la compra:

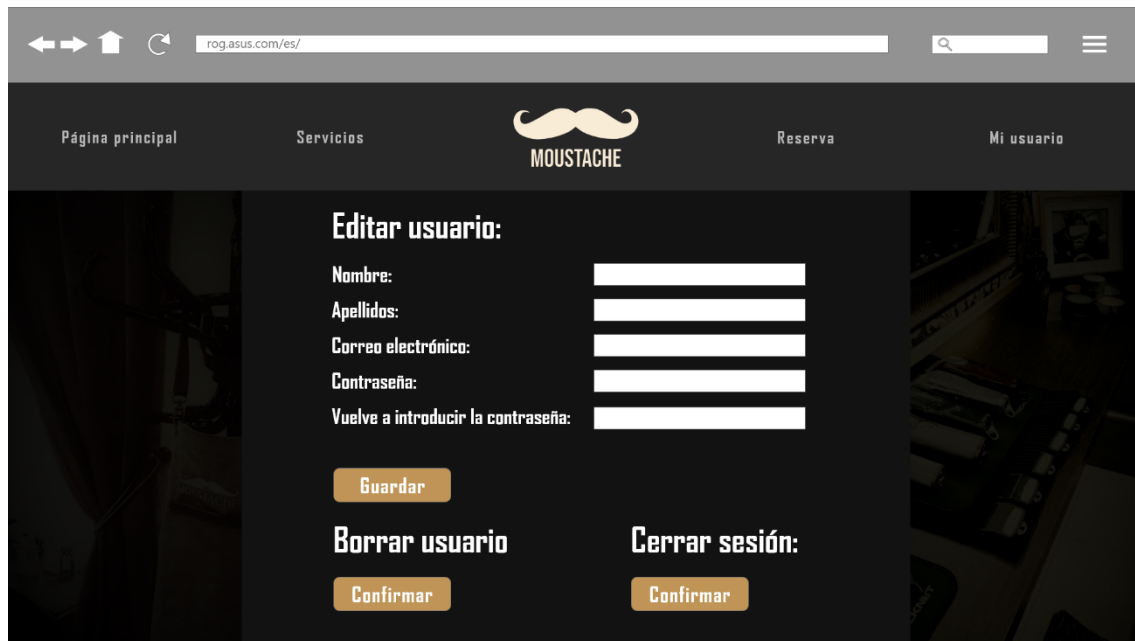
Este formulario contendrá diferentes campos para realizar la compra. Los textos representativos a los inputs tendrán tipografía “AgencyFB”, que le da una sensación de seriedad a la página.

Tras rellenar toda la información se redirigirá al usuario a la página oficial de la entidad bancaria de la barbería donde se realizará el pago de la prerreserva.

- Un recuadro con la información de la reserva con dos botones:
 - a. Reservar: Si este se pulsa, se redirigirá al usuario a la página oficial de la entidad bancaria.
 - b. Cancelar: Redirigirá al usuario a la página de reserva inicial (Figura 6.1).

Los botones de esta página tendrán el mismo formato que los botones de la figura 5.1.

- Modificar, dar de baja a un usuario y cerrar sesión.



The screenshot shows a web browser at the URL 'rog.asus.com/es/'. The page has a dark theme with a navigation bar at the top containing links: 'Página principal', 'Servicios', 'Reserva', and 'Mi usuario'. The 'Moustache' logo is centered in the navigation bar. The main content area is titled 'Editar usuario:' and contains five input fields for 'Nombre:', 'Apellidos:', 'Correo electrónico:', 'Contraseña:', and 'Vuelve a introducir la contraseña:'. Below these fields is a 'Guardar' button. To the right of the 'Guardar' button are two sections: 'Borrar usuario' with a 'Confirmar' button, and 'Cerrar sesión:' with a 'Confirmar' button. The background of the main content area features a faint image of a desk with a computer monitor and keyboard.

Figura 5.1

Esta interfaz está distribuida en tres partes:

❖ Editar usuario

En este apartado se podrá cambiar el nombre, apellido, el correo electrónico o la contraseña. Tras hacer los cambios, el usuario pulsará el botón de “Guardar” y se aplicarán los cambios si todos los campos cumplen con las reglas de la base de datos.

❖ Borrar usuario

En este apartado solo habrá un botón que confirme la eliminación de la información dentro de la base de datos.

❖ Cerrar sesión:

Aquí habrá un botón para cerrar la sesión del usuario.

DOCUMENTACIÓN DE LA APLICACIÓN

- Documentación técnica

En este apartado podremos encontrar la documentación necesaria para comprender el funcionamiento de toda la web, pasando tanto por los scripts de JavaScript (jQuery & Ajax incluido) como PHP. Toda página de esta web tiene un script de JavaScript (“JS[XXXX].js”), un script de PHP (“Codigo[XXXXXX].php”) y su propia página donde está todo el HTML con su correspondiente nombre. Para poder tener la información bien estructurada, la documentación estará separada por cada página que nos encontremos en la web. Estas son:

- [Scripts generales \(Scripts genéricos y código reutilizado de PHP\).](#)
- [La página principal \(“Home”\).](#)
- [Servicios.](#)
- [Reserva \(Primera, segunda, tercera y cuarta pantalla\).](#)
- [Panel de administrador](#)
- [Lista de reservas](#)
- [Log In.](#)
- [Alta.](#)
- [Modificar usuario.](#)

El apartado anterior puede servir al lector como índice, es decir, si se hace clic sobre el nombre, será redireccionado al correspondiente apartado de la documentación.

❖ Scripts generales

Es importante empezar por este apartado para poder comprender el funcionamiento de algunos apartados del sitio. Empezando por JavaScript, nos situamos dentro del directorio “./JS” y contaremos con los scripts:

➤ **cerrarSession.js**

Este script se ejecuta al pulsar el botón “Cerrar sesión” desde cualquier parte del sitio. Este ejecuta las órdenes del documento “cerrarSession.php” (explicado más adelante).

➤ **volverArriba.js**

Este script se activa tras pulsar el botón “Hacia arriba” que se encuentra en algunas páginas del sitio.

Este código está dividido en tres funciones:

- Una que detecta el scroll de la pantalla
- Otra que habilita o deshabilita el botón según la distancia bajada en la pantalla
- La función que hace volver al usuario a la parte superior de la pantalla.

Una vez explicados los scripts de JavaScript, pasamos a los trozos de código de PHP, para ello nos situamos dentro del directorio **“./PHP”** y veremos los siguientes ficheros:

➤ **BD_Connector.php**

Este trozo de código conecta el back-end con la base de datos proporcionada en la variable **“basededatos”** dentro del servidor determinado en la variable **“servidor”** con el usuario y contraseña en las variables **“usuario”** y **“password”**. La conexión se establece a través del objeto **“PDO”**, que proporciona una capa de seguridad más a esta conexión.

➤ **cerrarSession.php**

Haciendo énfasis en lo que se ha comentado anteriormente en el script de **“./JS/cerrarSession.js”**, el script ejecuta esta parte de PHP que es la encargada de destruir la sesión y redirigir al usuario a la pantalla principal.

➤ **EncriptarPassword.php**

Este archivo PHP no encripta la contraseña, sino que calcula el hash con la cadena de texto que se le pasa por **“POST”** al PHP. Usa el método **“SHA-256”**. Posteriormente a eso se devuelve la cadena y ya se opera con ella según las necesidades. Para mayor seguridad, la cadena de texto la cual se calcula el hash empieza por una palabra que solamente el administrador conoce.

❖ **Página principal**

El código back y front-end de esta web es muy escaso, ya que como el nombre indica, es una página de inicio, que ofrece información al usuario y lo redirige por las diferentes páginas del sitio, donde realmente queremos que llegue, por lo que aquí únicamente se muestra información relevante.

❖ **Servicios**

Siguiendo el patrón anterior, se empezará comentando el JS y luego el código PHP. Para esta página nos situamos dentro del directorio: **./JS/Servicio**. Dentro de ella nos encontramos con los siguientes scripts:

➤ **Carroussel.js**

Este script se encarga de toda la funcionalidad del Carroussel que hay dentro de la página. Asimismo, este está dividido en 4 funciones:

- La función que guarda en la variable de **“imagenCentral”** el elemento donde se va a poner la imagen del medio de la pantalla.
- La función **“imgClick”** que recibe por parámetro el número de la imagen que se ha pulsado. Esta función carga en la imagen central de la pantalla la imagen según el número recibido por parámetro.
- La función **“imgIzq”** que cambia la imagen central por la anterior a esta misma.
- La función **“imgDer”** que hace lo mismo que la anterior función, pero con la imagen siguiente.

➤ JSServicios

[vacío]

❖ Reserva

Nuevamente, empezaremos explicando la funcionalidad con los scripts de JavaScript. Para ello nos situamos dentro de la carpeta “./JS” y veremos los siguientes scripts:

➤ GenerarRegistros.js

Este script genera los inputs que nos encontramos dentro de la pantalla “./HTML/Reserva2.php”, es decir, todas las horas que dispone la semana. El script usa tanto JavaScript, como jQuery como Ajax. Para comprender dicho fichero de 500 líneas lo dividiremos en 2 partes:

- Las funciones que asignan las fechas a las variables:

Estas funciones, desde el mismo momento en el que se accede a esta pantalla se carga, modifica y guardan una serie de fechas en sus correspondientes variables. Las funciones que nos encontramos son las siguientes:

- La función que, nada más cargar la pantalla, carga las fechas en las variables “fechaComoCadena”, “fechaComoCadenaModificada” y “fechaComoCadenaInicioSemana”.
- La función CargarRegistros que, como su nombre indica, carga los registros y las fechas según la semana en la que se encuentre el usuario. Esta función recibe por parámetro la acción que se ejecuta (si se pasa a la semana adelante o atrás en caso de pulsar un botón), y el servicio que ha elegido el usuario (ya sea corte de pelo o retocado de barba).

Esta función se encarga de formatear las fechas que ya se han citado anteriormente y, según el día de la semana en el que se encuentra el usuario, asigna unos valores en la cadena que se encuentra en la parte superior de los botones de ir a la semana siguiente o ir a la semana anterior dentro del fichero “./HTML/Reserva2.php”.

Tras hacer este proceso, el programa guarda el año, mes, día, hora y minutos de ese preciso momento en el siguiente formato:

YYYYMMDDHHmm³. Conocer el formato anterior nos servirá para más adelante.

Más adelante tenemos un “for” que recorre los 7 días de la semana y genera sus correspondientes inputs. También hay un control de clics que va contando o descontando los clics que hace el usuario a los botones de “semana siguiente” o “semana anterior”.

³ YYYY: Año (4 dígitos), MM: Mes (2 dígitos), DD: Día (2 dígitos), HH: Hora (2 dígitos), mm: Minutos (2 dígitos). Conjunto: YYYYMMDDHHmm.

- Las funciones que generan los registros:
Tras conocer lo que hacen las funciones anteriores, se explicará el funcionamiento de las siguientes funciones:
 - Las funciones GenerarRegistrosPelo y GenerarRegistrosBarba, como su nombre indica, generan los inputs en los correspondientes días de la semana según la iteración del “for” en la que se encuentren. Por ejemplo, los lunes y los domingos son festivos, por lo que no se generan registros, pero de lunes a sábado, el programa genera registros tanto para la mañana como para la tarde (excluyendo los sábados).
Retomando el formato que se ha especificado anteriormente ⁽³⁾, cada botón que se encuentra en el calendario tendrá una id correspondiente a sus campos dentro del formato. Por lo que todos los botones tendrán una id distinta, y será más fácil operar con ellos.
Los parámetros recibidos son: El nombre de la semana sin tilde, la fecha correspondiente a cada botón, y la fecha del día actual, todo con su correspondiente formato.
El nombre de la semana se usa para determinar la columna donde se van a añadir los inputs (Columna del martes, miércoles, ...).
 - Tras verificar y otorgar toda la información a los botones, se compara el día actual con el de la id que corresponde en la iteración del “for” y, si el id es menor que el id de la fecha actual, se acude a la función “crearBotonExpirado”, pero si la fecha actual es menor al de id del botón (es decir, el botón de la hora corresponde a una fecha adelantada) se acude a la función “crearBoton”. A continuación, se explicará que hace cada una de las funciones:
 - CrearBotonExpirado:
Parámetros: botones, horasArray, nombreSemanaSinTilde.
El parámetro “botones” determina la columna donde va a ir el botón. “horasArray” determina el contenido que va a tener el input.
Como dice el nombre, esta función se llama en la función “GenerarRegistro[XXXXX]” en el caso de que la hora actual sea mayor que la id correspondiente al botón, se creará este botón deshabilitado que pondrá “EXPIRADO” en su value.
 - CrearBoton:
Parámetros: botones, IDPreparada, Servicio, horasArray, nombresSemanaSinTilde.
Esta función llama a Ajax a través de jQuery. Se conecta al fichero **./PHP/Reserva/ComprobarReservaDisponible.php** y le envía por “POST” la id que le corresponde al botón y el servicio que corresponde a la consulta. En caso de que la id ya exista en la base de datos con el nombre del servicio, se retorna el mensaje de “Ocupado” al JavaScript. Este deshabilita el botón en este caso, pero

en caso contrario, crea un input normal para que el usuario haga clic en él.

➤ **JSReserva.js**

Este es el fichero JavaScript que corresponde a las diferentes páginas de Reserva. En él nos encontramos 2 funciones:

- “volverReserva”:
Esta función ejecuta el fichero **“./PHP/Reserva/BorrarSessionReserva.php”** y recarga la página.
- “AbrirModal”:
Parámetros: botonPresionado, diaSemana.
Esta función se ejecuta en la página “Reserva2.php” cuando se pulsa sobre un input habilitado en el calendario. Formatea su id a Date para poder mostrar el mensaje que aparece en el modal.
Dentro del modal se guardan en unos inputs type hidden el valor del id que ha recuperado el modal a través de sus parámetros (“botonPresionado”); en uno de ellos lo deja como string y en el otro en el formato Date que el propio modal ha creado.

Una vez explicado todas las funciones de JavaScript, procedo a explicar el código PHP. Para ello nos situamos en el directorio **“./PHP/Reserva”**. Nos encontramos con los siguientes ficheros:

➤ **BorrarSessionReserva.php**

Este trozo de código de PHP resetea las variables de sesión y las deja vacías. Este trozo de código se ejecuta en el script “volverReserva” que ya se ha comentado anteriormente.

➤ **ComprobarReserva.php**

Redirige al usuario al login en caso de que no haya iniciado sesión.

➤ **CodigoReserva1.php**

Comprueba si se ha pulsado alguno de los botones de los servicios. Tras ser pulsado, comprueba el servicio y lo guarda en la variable de sesión “servicio”. Tras eso, se redirige al usuario a la página “Reserva2.php”.

➤ **ComprobarReservaDisponible.php**

Este bloque de código se ejecuta en el trozo de código Ajax que se encuentra en la función “crearBoton” del script “GenerarRegistros.js”. Comprueba que el id recibido por post a través del data de Ajax exista en la base de datos, en caso de que exista se le devuelve el mensaje “Ocupado” al script y este ya lo gestiona.

➤ **CodigoReserva2.php**

Este trozo de código cuenta con 3 fases de comprobación:

- Comprueba que el usuario haya seleccionado un servicio
- Comprueba que se haya pulsado el botón para volver atrás
- Comprueba que se haya pulsado el botón del modal, en el caso de que se haya pulsado, recibe el contenido del input hidden que se ha comentado anteriormente y lo guarda en la variable de sesión “fechaReserva”

➤ **CodigoReserva3.php**

Aquí tenemos 3 fases de comprobación, igual que en el bloque anterior:

- Uno comprueba que se haya seleccionado una hora de reserva
- Otra comprueba que se haya pulsado el botón de volver
- El último comprueba que se haya realizado el pago correctamente.

➤ **CodigoReserva4.php**

Este último se encarga de reunir toda la información que se ha ido rellenando durante las 3 pantallas de reserva y las incluye en un “insert” a la base de datos. De esta manera, se guarda la reserva que ha realizado el usuario.

❖ Panel de administrador

Esta página, junto a “Lista de reservas”, son páginas que únicamente puede ver el administrador, por lo que, en la barra de navegación tendremos una comprobación con PHP que comprobará que, en la variable de sesión, el ID sea 0 (el administrador). Una vez comentado esto, procedemos a situarnos en el directorio “./JS/PanelAdministrador” y comentaremos el código JavaScript que usa esta página:

➤ **JSPanelAdministrador.js**

Esta página está separada en dos partes; la función que ejecuta el modal de confirmación y la que determina la hora del reloj digital que tenemos en la parte superior de la página.

- La función que gestiona el reloj se trata de un “setInterval” que va ejecutando una función anónima cada segundo. Esta le da los valores correspondientes al reloj digital y, además, guarda en una variable hidden el valor del tiempo ⁽³⁾.
- La función que abre el modal muestra un texto de confirmación conforme se va a borrar el contenido de la base de datos que corresponda a las reservas anteriores a la fecha en ese mismo momento.

Tras explicar las partes de JavaScript que tiene esta página pasamos a la parte de PHP. Para ello nos situamos en el directorio “./PHP/PanelAdministrador”.

➤ **CodigoPanelAdministrador**

Este trozo de código cuenta con varias fases de validación; estamos hablando de que, desde este panel, todo usuario tiene la capacidad de administrar las reservas. Para la seguridad de dicha página tenemos las siguientes funciones:

- Las funciones que controlan que el usuario haya iniciado sesión y que, dicho usuario sea el administrador de la página.

Las demás funciones controlan las acciones del usuario dentro de la página:

- Las funciones que desplazan al usuario por las distintas páginas del programa según lo que hayan pulsado.
- La función que se encarga de que, en caso de que se confirme que se borren todos los registros anteriores a esa fecha, se borren en la base de datos.

❖ Lista de reservas

Esta página muestra las reservas que dispone la barbería por días. Hay 3 maneras de ver las reservas; de todos los servicios, de corte de pelo o de reservas de barbas. El JavaScript se encarga de generar los registros según la información recuperada. Para ver las funciones JavaScript nos situamos en el directorio **“./JS/ListaReservas”**. Ahí nos encontraremos con los siguientes bloques.

➤ **JSListadoReserva.js**

Este archivo JavaScript está dividido en varias funciones, pero las tres principales son:

- **“CargarRegistros”:**
Esta función actualiza el día del que se quiere extraer las reservas.
- **“RecuperarRegistros”:**
Recibe el día en que se encuentra el usuario por parámetro. Esta función hace una petición Ajax que devuelve en formato JSON los datos de las reservas que hay ese día. Si no se encuentra ningún registro, se activa un **“alert”** conforme no se ha recuperado nada.
- **“GenerarRegistros”:**
Este método recibe los datos de la petición Ajax por parámetro, junto al servicio seleccionado en el desplegable. Tras ello, guarda los datos en una variable y, con ayuda de jQuery, se crean los registros en formato de lista cargando lo que se encuentra dentro de la variable.

Estas tres funciones son muy parecidas a las que se encuentran en el fichero **“./JS/Reserva/GenerarRegistros.js”**, salvo que estas cargan los datos de un solo día, y no una semana entera.

Tras explicar el JavaScript que hay en la página, procedemos a explicar los bloques PHP que se encuentran dentro del directorio **“./PHP/ListaReservas”**.

➤ **CargarRegistros.php**

Este bloque es el PHP que se ejecuta tras la petición Ajax que hay en el JavaScript comentado anteriormente devuelve los registros de las bases de datos según el servicio seleccionado y la fecha que se quiera consultar.

Tras cargar los datos, los parsea a JSON y en caso de que se consulten todos los registros de ambas tablas, se organizan según su fecha de reserva.

➤ **CodigoListaReservas.php**

Igual que con el panel de administrador, aquí nos encontramos con dos funciones que controlan el acceso a esta página. Si el usuario es el administrador, puede acceder aquí.

❖ Log in

El Log in es una de las páginas con menos código. En este caso, la página no cuenta con código JavaScript, porque la totalidad de este es PHP, que compara la información proporcionada del usuario con la que hay en el servidor. Para poder comentar este trozo de código, nos situaremos en el directorio **“./PHP/Login”**:

➤ **CodigoLogin.php**

Este bloque de código recibe por “POST” los dos valores del login; el correo con el que se ha registrado el usuario y su contraseña respectivamente. Para iniciar sesión, el código PHP acude al fichero “BD_Connector” para conectarse a la base de datos y tras eso, calcula el hash de la contraseña introducida por el usuario con el bloque de código que se encuentra dentro del fichero “EncriptarPassword.php”.

Tras calcular el hash, se compara el usuario y el hash calculado con los datos de la base de datos, si devuelve un registro, inicia sesión, sino se muestra un mensaje según el error ocurrido.

❖ Alta

A diferencia del Login, la página de alta registra según los datos introducidos por el usuario, este mismo a la base de datos. Para ello tenemos tanto código JavaScript como PHP. Como de costumbre, empezaremos comentando el código JavaScript, y para ello, nos situaremos en el directorio **“./JS/Alta”**:

➤ **JSAlta.js**

En este fichero contamos con 5 funciones.

- **“llamarServidor” y “respuestaServidor”**:
La función llamarServidor se ejecuta cuando se pierde el foco en el input del correo. Esta llama al código PHP “estadoUsuRegistro.php” (comentada más adelante) y según la respuesta, el div de comprobación mostrará un contenido u otro.
- **“estadoCheckbox” y “habilitarLogin”**:
Si el checkbox de términos y condiciones está activo y se cumplen los requisitos para iniciar sesión, el botón de “Crear cuenta” se habilitará y permitirá al usuario crear su cuenta.
- **“comprobarEstadoInputs”**:
Por último pero no menos importante, esta función recibe por parámetro el número del input del cual se está perdiendo el foco. Si este está vacío, se usa una de las clases que tiene Bootstrap para remarcar el input de color rojo y muestra un pequeño texto debajo de este advirtiéndolo del campo vacío.

Tras comentar el fichero JavaScript, procedemos a comentar el PHP. Para ello nos situamos dentro del directorio **“./PHP/Alta/”**:

➤ **estadoUsuRegistro.php**

Este bloque de código PHP se ejecuta a través del bloque Ajax de la función “llamarServidor” que ya se ha comentado anteriormente. Este bloque comprueba que el correo que ha introducido el usuario ya esté registrado en la base de datos, en ese caso, bajo el input del correo saldrá una alerta que advertirá al usuario que el correo ya existe y el programa propondrá dos correos nuevos para que el usuario los pueda usar.

➤ **CodigoAlta.php**

En este trozo de código se inserta en la base de datos todos los datos introducidos en los inputs por el usuario. Inicialmente tenemos un método de comprobación que comprueba que el usuario ya esté con la sesión iniciada, en ese caso, este se redirigirá a la página principal.

Posteriormente pasamos a los métodos de comprobación necesarios para proceder con la inserción de los datos en la base de datos. Inicialmente comprobamos que no haya ningún campo vacío, si no lo hay, se comparan las dos contraseñas introducidas por el usuario. Si ambas son iguales, entonces se procede con el cálculo del hash de la contraseña y se introduce los datos del usuario a la base de datos.

❖ **Modificar usuario**

Esta última página cuenta con JavaScript como PHP. Como las anteriores veces, empezaremos a comentar el código JavaScript, para ello, nos situamos en el directorio **“./JS/ModificarUsuario”**:

➤ **JSModificarUsuario**

Dentro de este script contamos con dos funciones; una realiza una petición Ajax y la otra opera con el resultado.

Como en el alta de usuario, se debe comprobar antes de modificar el usuario que el campo del correo electrónico no esté duplicado, por lo que esta función Ajax hace una consulta a la base de datos y recupera los correos que cuenten con esa dirección de email. Si no existe ningún registro con dicho correo se habilitará el botón de modificar usuario.

Pasamos al código PHP, el cual se almacena dentro del directorio **“./PHP/ModificarUsuario”**:

➤ **validarEmail.php**

Este bloque de código es completamente igual que el que tenemos en el fichero de **“./PHP/ estadoUsuRegistro.php”**; Comprobamos la existencia del correo en la base de datos y en el caso de que exista, el programa lanzará un mensaje de error.

➤ **CodigoModificarUsuario.php**

En este bloque inicialmente precargamos el nombre, el apellido y el email del usuario en los inputs correspondientes como “placeholder”, por lo que el usuario podrá ver en la interfaz que datos tiene guardados en la base de datos. También tenemos el método de control que redirige al usuario a la página principal en caso de no haber iniciado sesión.

Este bloque de código está separado en dos funcionalidades:

- **Modificar usuario:**
Según los campos que haya rellenado el usuario, la query “Update” que se ejecuta en la base de datos se va rellenando con los valores que ha introducido el usuario.
- **Botón de borrar:**
Este botón está situado en un modal que se ejecuta cuando se pulsa el botón de “Borrar usuario” que se encuentra en la parte inferior de la página. Si, al ejecutarse el modal, se pulsa al botón de confirmación para borrar el usuario, se lanzará una query “delete” para borrar al usuario de la base de datos.

- Documentación de usuario (POR HACER)

CONCLUSIONES

La aplicación web de gestión de reservas para una barbería ha demostrado ser una herramienta útil y efectiva para facilitar la programación de citas y gestionar los usuarios de manera eficiente. Los resultados obtenidos muestran una mejora en la organización de la barbería y en la satisfacción de los clientes al poder reservar con facilidad y accesibilidad.

A pesar de algunas dificultades técnicas encontradas durante el desarrollo del proyecto, se han podido solucionar satisfactoriamente. Se ha conseguido resumir los principales aspectos del proyecto y los resultados obtenidos, sin añadir información nueva y sin repetir lo ya expuesto.

La importancia de la aplicación web para la barbería ha sido destacada en la conclusión, mejorando así la eficiencia del negocio y la satisfacción de los clientes.

Se ha motivado a seguir reflexionando sobre el tema y se ha sugerido la posibilidad de implementar nuevas funcionalidades para mejorar aún más la gestión de reservas y usuarios. En general, la conclusión invita a continuar trabajando en el desarrollo de la aplicación web para mejorar y optimizar la gestión de la barbería.

WEBGRAFÍA

Viabilidad del proyecto → Estudio de mercado → Aplicaciones / webs que cumplan una función similar y el valor añadido de la nuestra.

- Trivago (2023 ©). Todas las ofertas de tus webs de reserva favoritas. Recuperado de: <https://www.trivago.es/>
- SegurCaixa Adeslas (2023 ©). Servicios y seguros. Recuperado de: <https://www.segurcaixaadeslas.es/>

Viabilidad del proyecto → Project Charter → Descripción del proyecto → Presupuesto estimado

- Hostalia (2001 - 2023 ©). Registra tus dominios y protege tu marca. Recuperado de: <https://www.hostalia.com/dominios/dominios-territoriales/>