



DOCUMENTACIÓN

PROYECTO INTEGRADO

FRANCISCO LÓPEZ DE LA FUENTE

Abstract

Se presenta una descripción concisa de la página web de LaFuente.

La plataforma ofrece a los clientes una amplia gama de platos, servicios de reserva sencillos y una experiencia memorable, todo ello en un entorno digital cuidadosamente diseñado.

La página web de LaFuente se convierte en el punto de partida para explorar una experiencia inigualable. Con un diseño elegante y una interfaz intuitiva, la página principal atrae a los visitantes con imágenes brillantes de los platos más emblemáticos del restaurante y una navegación fluida a través de las distintas secciones.

La página de reservas proporciona a los comensales una manera sencilla y conveniente de asegurar su lugar en este destino. A través de un sistema de reservas en línea fácil de usar, los clientes pueden seleccionar la fecha, la hora y el tamaño de su grupo.

La página web incluye una sección dedicada a las reseñas de los clientes, donde los visitantes pueden leer las experiencias y comentarios de aquellos que han disfrutado de la hospitalidad de LaFuente.

A concise description of the LaFuente website is presented.

The platform offers customers a wide range of dishes, easy booking services, and a memorable experience, all within a carefully designed digital environment.

The LaFuente website becomes the starting point for exploring an unparalleled experience. With an elegant design and an intuitive interface, the homepage attracts visitors with vibrant images of the restaurant's most iconic dishes and smooth navigation through different sections.

The reservations page provides diners with a simple and convenient way to secure their spot at this destination. Through an easy-to-use online booking system, customers can select the date, time, and size of their group.

The website includes a section dedicated to customer reviews, where visitors can read the experiences and comments of those who have enjoyed LaFuente's hospitality.

Indice

4. Justificación.....	4
4.1. Características generales.....	4
4.2. Restricciones generales, si las hay.....	4
4.3. Aspectos a cubrir junto con los que no se van a tratar.....	4
4.4. Estudios de las prestaciones de la herramienta que se propone.....	4
5. Justificación de las tecnologías empleadas.....	5
6. Requerimientos hardware y software.....	5
7. Análisis y diseño.....	6
8. Implementación.....	8
9. Relación detallada de uno de los ficheros más importantes.....	11
10. Evaluación y prueba.....	14
11. Manual de estilos en el que se recojan.....	16
Sketches:.....	16
11.2 Criterios de accesibilidad.....	19
11.3. Criterios de usabilidad.....	20
11.4. Fuentes principales y secundarias.....	20
11.5 Tamaños principales. Elementos donde se aplican.....	20
11.6 Mapa de colores de proyecto en tres formatos (RGB, Hexadecimal, nombre junto con el color). Elementos donde se aplica.....	22
11.7. Dispositivos/vistas para las que se ha diseñado el proyecto.....	23
12. Software utilizado.....	24
13. Mejoras posibles y aportaciones.....	24
Licencias.....	25
Bibliografía.....	25

4. Justificación

4.1. Características generales.

- Capacidad de autodeterminación, intuitiva y simple: para que la actualización de contenido sea constante, la web debe de ser de fácil gestión tanto si disponen de conocimientos técnicos o no.
- Página web adaptativa: ésta web está diseñada para que todos los usuarios tenga acceso a la navegación de la página, ya sea desde móvil, tablets, portátiles u ordenadores.
- Velocidad de carga: éste es el punto más importante, ya que la web no es lenta por lo que los usuarios no tendrán que esperar mucho tiempo para realizar cualquier tarea.
- Elementos de iteración: los usuarios podrán mantener contacto a través de iconos de las redes sociales.
- Usabilidad y organización: darle control absoluto al usuario facilita la navegación para que se sienta más cómodo y quiera pasar más tiempo.
- Diseño: ésta web tiene una imagen muy profesional ya que el diseño está dirigido a un personal adulto.

4.2. Restricciones generales, si las hay

Ésta web tiene 2 restricciones, la cuales son:

- 1.No está diseñada para la resolución 4k.
- 2.Al tener un hosting barato la funcionalidad de eliminar y editar no se podrán realizar.

4.3. Aspectos a cubrir junto con los que no se van a tratar

- Pedidos en línea: permitir a los clientes realizar pedidos y pagos en línea para llevar o entregar a domicilio.
- Reservas especiales: implementar sistema de reservas para eventos privados o celebraciones especiales.
- Recompensas: sistema para acumular punto cuando realice un pedido y poder obtener descuentos.

4.4. Estudios de las prestaciones de la herramienta que se propone

Comparado la web realizada con otras webs del mercado actuales, se diferencia por el acceso a toda la funcionalidad de la web, ya que el cliente se tendrá que registrar ya sea para hacer una reserva, opinar o una noticia necesita registrarse. Ésto tiene su explicación ya que si no existiese un sistema de registro cualquier persona inmadura podrá realizar varias reservas, y cuando llegue ese día al restaurante no le va a sentar bien tener todas esas reservas y que sean inventadas.

5. Justificación de las tecnologías empleadas

1.Laravel: es un framework de desarrollo de aplicaciones web de código abierto, escrito en PHP. Éste sigue la estructura de Modelo-Vista-Controlador (MVC), lo que significa que separa la lógica de la interfaz de usuario y permite un desarrollo estructurado y modular. Proporciona una amplia variedad de características y herramientas que facilitan el desarrollo rápido y eficiente de aplicaciones web.

2.Vue.js: es un framework de JavaScript que permite crear interfaces de usuario participativos y dinámicas. Es fácil de aprender y utilizar. El objetivo de Vue es crear componentes, los cuales se reutilizan y minimizan el código repetido. Esto facilita la creación de aplicaciones complejas, ya que puedes dividir tu código en componentes más pequeños y utilizarlos de manera independiente.

3.Tailwind CSS: es un framework de CSS que utiliza código abierto y permite construir interfaces web de manera rápida y eficiente. Tailwind no se basa en componentes predefinidos, sino que se enfoca en proporcionar una amplia categoría de clases los cuales se pueden aplicar directamente en el HTML.

He usado éstas tecnologías ya que las he utilizado a lo largo del curso y me desenvuelvo bien, son fáciles de aprender y tienen una sintaxis limpia, en cambio para darle un estilo a la web he utilizado Tailwind, una nueva herramienta, ya que es mucho mejor que Bootstrap, por varias razones y una de ellas es que se pueden personalizar y es fácil de entender,

6. Requerimientos hardware y software

A nivel de cliente:

Requerimientos de hardware

- Ordenador o dispositivo móvil: se necesita un dispositivo con acceso a Internet para poder acceder a a página web del restaurante.

Requerimientos de software

- Navegador web: se requiere un navegador actualizado para poder visualizar correctamente la web.
- Conexión a internet: es necesario tener acceso a Internet para poder cargar y navegar.
- Sistema Operativo: El dispositivo debe tener un sistema operativo compatible con el navegador web utilizado, como Windows, macOS, Linux, Android o iOS

A nivel de servidor:

Requerimientos de hardware

- Procesador: se recomienda tener un procesador con suficiente potencia para manejar las solicitudes de los visitantes.
- Memoria RAM: es necesario tener suficiente memoria para evitar cuellos de botella y garantizar un rendimiento óptimo.
- Almacenamiento: se necesita espacio de almacenamiento para almacenar el sitio web, imágenes, archivos y datos relacionados, Éste dependerá del tamaño del sitio y contenidos relacionados.
- Conectividad: se requiere de conexión a Internet de alta velocidad y confiable para garantizar que el sitio web esté siempre operativo.

Requerimientos de software

- Servidor Web: necesario para alojar el sitio web y mostrar las páginas a los visitantes.
- Base de datos: se requiere utilizar una base de datos para almacenar información como (MySQL)
- Seguridad: importante implementar medidas de seguridad, como cortafuegos, certificados SSL/TLS y sistemas de detección de intrusos, para proteger el servidor y los datos.

7. Análisis y diseño

Diagrama de Casos de Uso

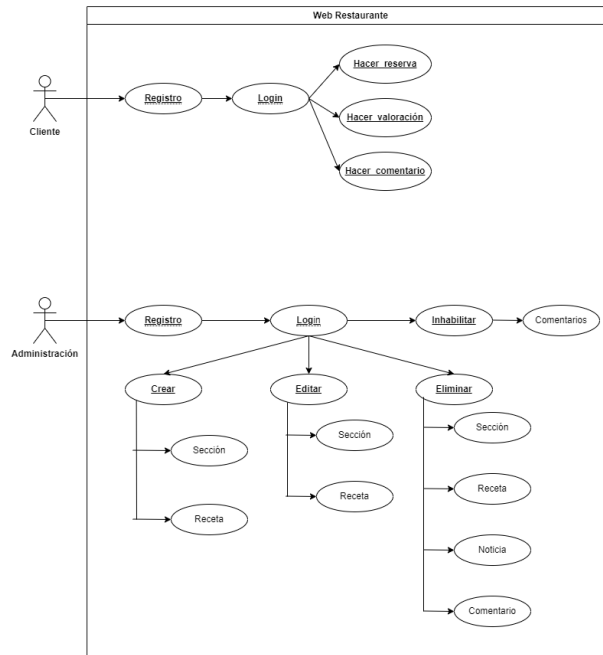
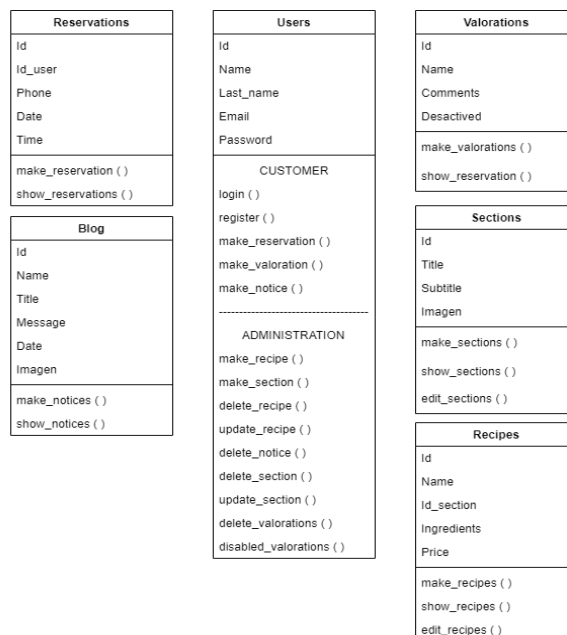
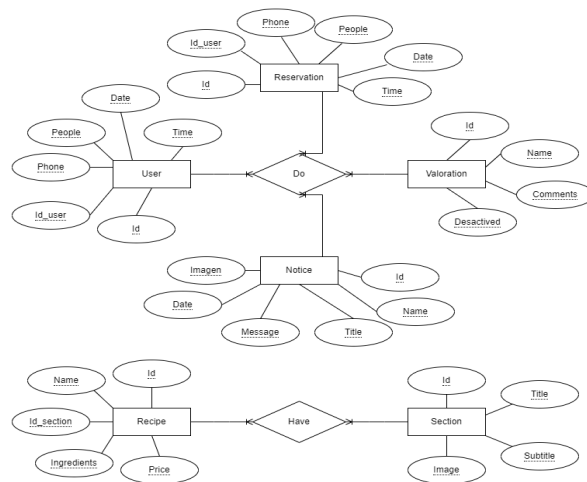


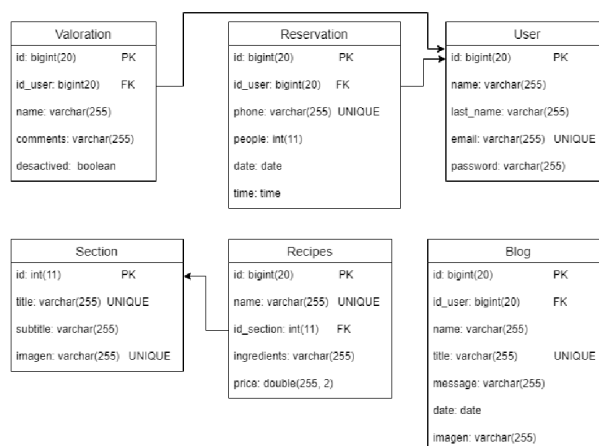
Diagrama de Clases



Modelo Entidad-Relación



Diseño Lógico de tablas



La BBDD de éste restaurante se almacenarán información que permite organizar y gestionar los datos relacionados con el funcionamiento del restaurante y sus servicios. Esta base de datos esta diseñada para almacenar y recuperar datos relevantes sobre los menús, clientes, reservas, opiniones, noticias y otra información importante para el restaurante.

-Usuarios: almacena información de los usuarios que han interactuado con la web.

-Menús: contiene información sobre los diferentes menús ofrecidos por el restaurante.

-Reservas: registra las reservas realizadas por los clientes.

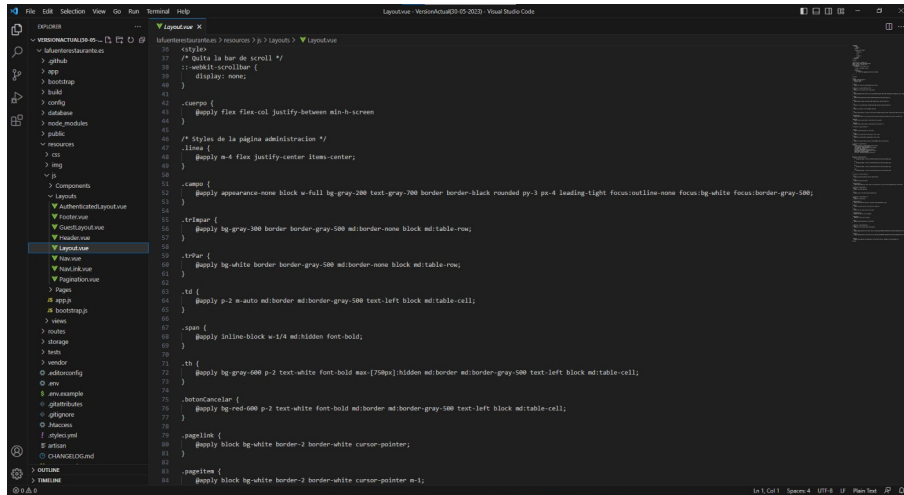
-Opiniones: se utiliza para almacenar las diferentes opiniones de los clientes sobre su experiencia.

-Noticias: se almacenaran las noticias que publiquen los usuarios para informar a otras personas con fotos del restaurante, platos, etc

8. Implementación

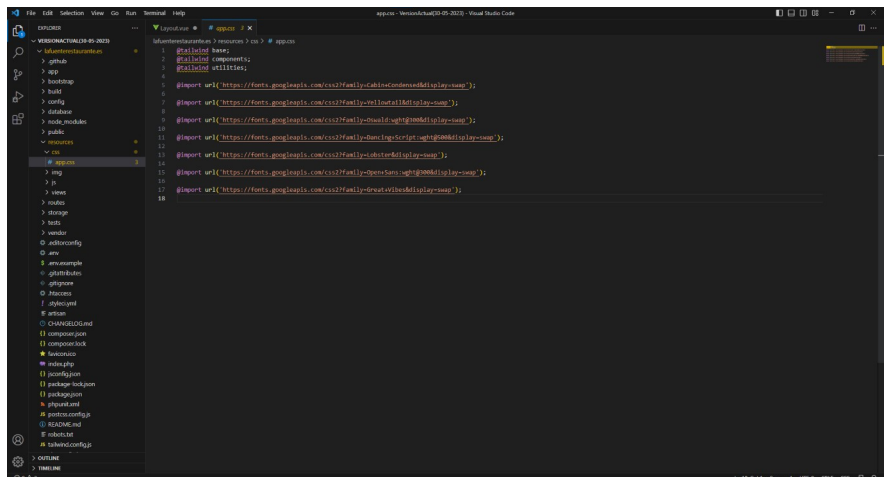
1.Hojas de estilo: para dar estilo y formato a la web he utilizado un framework de CSS llamado Tailwind, éste usa clases ya por defecto pero se pueden personalizar.

Fichero dónde se realiza el diseño de las páginas:

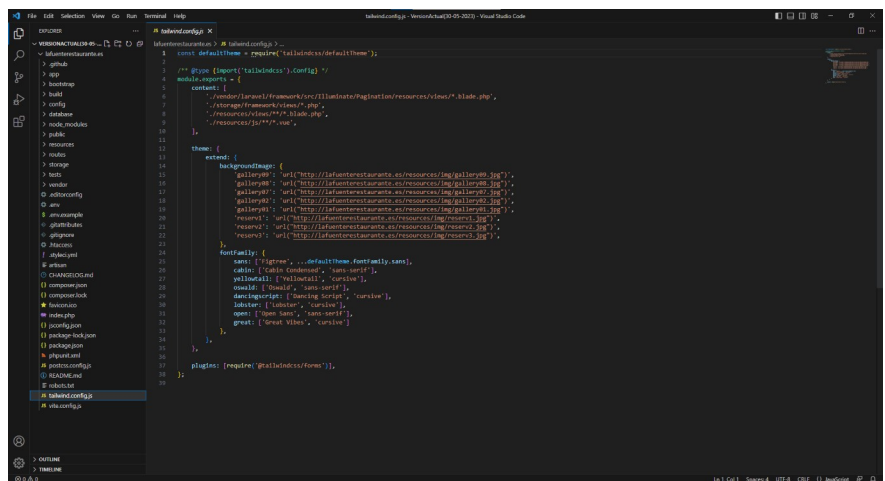


```
1 @tailwind base;
2 @tailwind components;
3 @tailwind utilities;
4
5 @import url('https://fonts.googleapis.com/css2?family=CabinCondensed&display=swap');
6 @import url('https://fonts.googleapis.com/css2?family=Inter&display=swap');
7 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&display=swap');
8 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&display=swap');
9 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&display=swap');
10 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&display=swap');
11 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&display=swap');
12 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&display=swap');
13 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&display=swap');
14 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&family=RobotoMono5&display=swap');
15 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&family=RobotoMono5&family=RobotoMono6&display=swap');
16 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&family=RobotoMono5&family=RobotoMono6&family=RobotoMono7&display=swap');
17 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&family=RobotoMono5&family=RobotoMono6&family=RobotoMono7&family=RobotoMono8&display=swap');
18 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&family=RobotoMono5&family=RobotoMono6&family=RobotoMono7&family=RobotoMono8&family=RobotoMono9&display=swap');
```

Para personalizar las clases de Tailwind se utiliza un archivo en concreto llamado: tailwind.config.js, hay que tener en cuenta que para que las diferentes fuentes funcionen correctamente debemos de tener otro archivo app.css donde se importará la ruta y el módulo de tailwind.

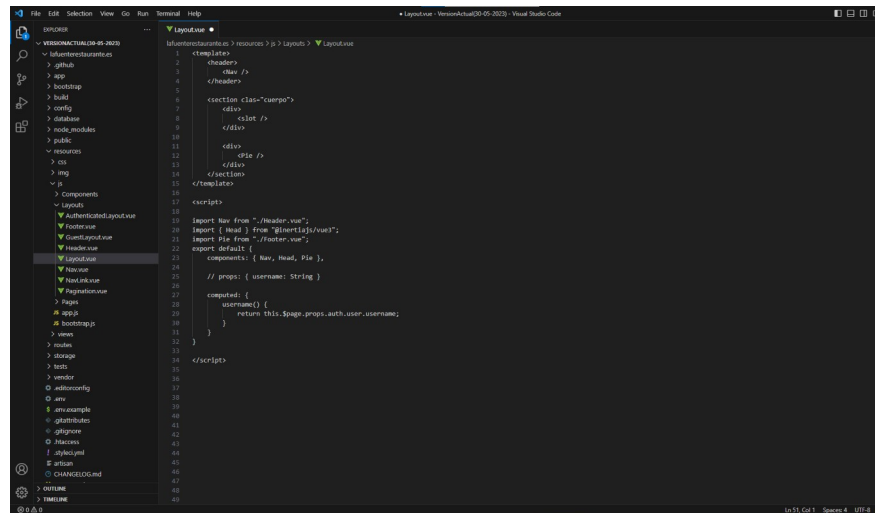


```
1 @tailwind base;
2 @tailwind components;
3 @tailwind utilities;
4
5 @import url('https://fonts.googleapis.com/css2?family=CabinCondensed&display=swap');
6 @import url('https://fonts.googleapis.com/css2?family=Inter&display=swap');
7 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&display=swap');
8 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&display=swap');
9 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&display=swap');
10 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&display=swap');
11 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&display=swap');
12 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&display=swap');
13 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&display=swap');
14 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&family=RobotoMono5&display=swap');
15 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&family=RobotoMono5&family=RobotoMono6&display=swap');
16 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&family=RobotoMono5&family=RobotoMono6&family=RobotoMono7&display=swap');
17 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&family=RobotoMono5&family=RobotoMono6&family=RobotoMono7&family=RobotoMono8&display=swap');
18 @import url('https://fonts.googleapis.com/css2?family=Inter&family=Roboto&family=RobotoCondensed&family=RobotoSlab&family=RobotoMono&family=RobotoMono2&family=RobotoMono3&family=RobotoMono4&family=RobotoMono5&family=RobotoMono6&family=RobotoMono7&family=RobotoMono8&family=RobotoMono9&display=swap');
```



```
1 const defaultTheme = require('tailwindcss/defaultTheme');
2
3 /** @type {import('tailwindcss').Config} */
4 module.exports = {
5   content: [
6     './pages/**/*.html',
7     './resources/**/*.php',
8     './resources/**/*.css',
9     './resources/**/*.js',
10     './resources/**/*.vue',
11   ],
12   theme: {
13     extend: {
14       fontFamily: {
15         sans: ['Inter', ...defaultTheme.fontFamily.sans],
16         serif: ['CabinCondensed', ...defaultTheme.fontFamily.serif],
17         mono: ['RobotoMono', ...defaultTheme.fontFamily.mono],
18         display: ['RobotoCondensed', ...defaultTheme.fontFamily.display],
19         script: ['RobotoMono', ...defaultTheme.fontFamily.script],
20         cursive: ['RobotoMono', ...defaultTheme.fontFamily.cursive],
21         sans-serif: ['Inter', ...defaultTheme.fontFamily.sans],
22         serif-serif: ['CabinCondensed', ...defaultTheme.fontFamily.serif],
23         mono-mono: ['RobotoMono', ...defaultTheme.fontFamily.mono],
24         display-display: ['RobotoCondensed', ...defaultTheme.fontFamily.display],
25         script-script: ['RobotoMono', ...defaultTheme.fontFamily.script],
26         cursive-cursive: ['RobotoMono', ...defaultTheme.fontFamily.cursive],
27       },
28     },
29   },
30   plugins: [require('@tailwindcss/forms')],
31 }
```

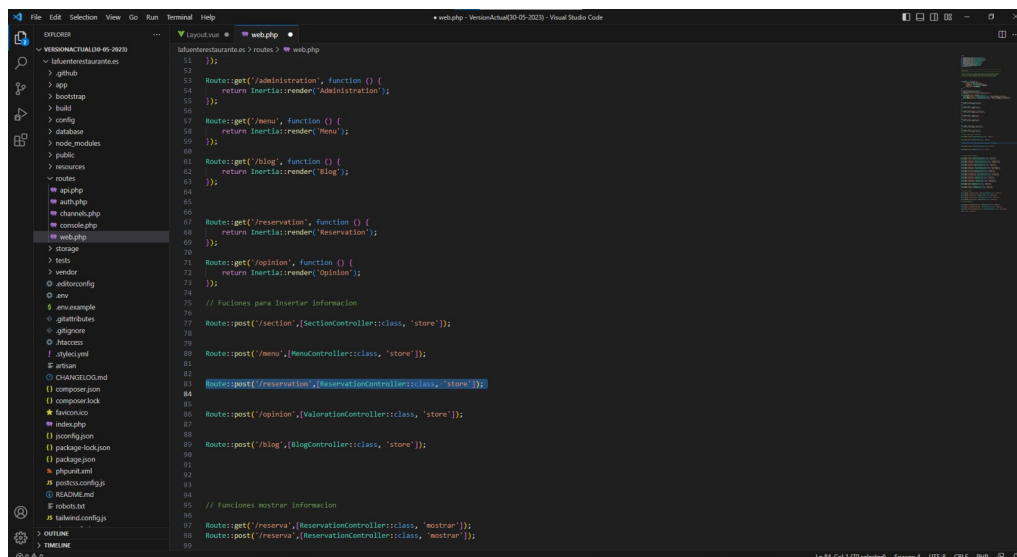

2.Plantillas: para optimizar código y no estar utilizando el mismo en diferentes ficheros he creado una plantilla la cual va a ser el cuerpo de la web.



```
1 <template>
2   <div>
3     <h1>
4   </h1>
5   <div class="cuerpo">
6     <div>
7       <h2>
8     </h2>
9   </div>
10  </div>
11  </div>
12  </div>
13  </div>
14  </div>
15  </div>
16  </div>
17  </div>
18  </div>
19  </div>
20  </div>
21  </div>
22  </div>
23  </div>
24  </div>
25  </div>
26  </div>
27  </div>
28  </div>
29  </div>
30  </div>
31  </div>
32  </div>
33  </div>
34  </div>
35  </div>
36  </div>
37  </div>
38  </div>
39  </div>
40  </div>
41  </div>
42  </div>
43  </div>
44  </div>
45  </div>
46  </div>
47  </div>
48  </div>
49  </div>
50  </div>
51  </div>
52  </div>
53  </div>
54  </div>
55  </div>
56  </div>
57  </div>
58  </div>
59  </div>
60  </div>
61  </div>
62  </div>
63  </div>
64  </div>
65  </div>
66  </div>
67  </div>
68  </div>
69  </div>
70  </div>
71  </div>
72  </div>
73  </div>
74  </div>
75  </div>
76  </div>
77  </div>
78  </div>
79  </div>
80  </div>
81  </div>
82  </div>
83  </div>
84  </div>
85  </div>
86  </div>
87  </div>
88  </div>
89  </div>
90  </div>
91  </div>
92  </div>
93  </div>
94  </div>
95  </div>
96  </div>
97  </div>
98  </div>
99  </div>
100 </div>
</pre>
```

3. Formularios: las diferentes funcionalidades de la web son realizar reservas, comentarios, noticias y registrarse. Para ello se hace uso de diferentes ficheros los cuales hacen distintas funcionalidades. A continuación muestro todos los archivos que necesita la funcionalidad de hacer una reserva.

Éste primer archivo se utiliza para manejar las rutas y redirigir las solicitudes a los controladores correspondientes, éste determina que controlador utiliza y la acción que debe de realizar una solicitud HTTP específica.



```
1 <?php
2 use Illuminate\Support\Facades\Route;
3
4 // Funciones para Insertar Información
5
6 Route::get('/administracion', function () {
7     return inertia::render('Administracion');
8 });
9
10 Route::get('/menu', function () {
11     return inertia::render('Menu');
12 });
13
14 Route::get('/blog', function () {
15     return inertia::render('Blog');
16 });
17
18 Route::get('/reservacion', function () {
19     return inertia::render('Reservacion');
20 });
21
22 Route::get('/opiniones', function () {
23     return inertia::render('Opinion');
24 });
25
26 // Funciones para Insertar Información
27
28 Route::post('/section',[SectionController::class,'store']);
29
30 Route::post('/menu',[MenuController::class,'store']);
31
32 Route::post('/reservacion',[ReservationController::class,'store']);
33
34 Route::post('/opiniones',[ValoracionController::class,'store']);
35
36 Route::post('/blog',[BlogController::class,'store']);
37
38 // Funciones mostrar Información
39
40 Route::get('/reserva',[ReservationController::class,'mostrar']);
41
42 Route::post('/reserva',[ReservationController::class,'mostrar']);
43
44 </pre>
```

La ruta seleccionada dirige la solicitud al controlador ReservationController.

El segundo archivo es el controlador donde realizan la creación de la Reserva, después de que cumplan las diferentes condiciones.



4.Consultas a base de datos: para mostrar los datos en general he utilizado el método SELECT

Este método lo utilizo por ejemplo en el controlador del Menú:

```
/**
 * Muestra los datos los platos que hay en el menu.
 */
public function mostrar()
{
    $datosPlatos = Menu::select('id','name', 'price','ingredients','id_section')->get();
    return $datosPlatos;
}
```

9. Relación detallada de uno de los ficheros más importantes

La página más relevante de la web se trata de la administración que permite al administrador o propietario gestionar y actualizar datos de la web. A continuación se detallan las funcionalidades ya acciones posibles:

1. Gráficas: compuesta por dos graficas para mostrar datos importantes. Una muestra el numero de reservas realizadas, lo cual permite visualizar la cantidad de clientes que han reservado mesas en un periodo determinado.
La otra graficas muestra e numero de platos pro secciones lo que te permite obtener una visión general de la distribución de los platos en cada sección del menú.
2. Tablas: compuesta por cinco tablas principales:
 1. a)Platos: muestra una lista de los platos disponibles en el menú, puede ver información como el nombre, descripción, precio y la sección a la que pertenece. Además cuenta con un botón para editar los datos de cada plato y otro para eliminarlo de la base de datos

2. Secciones: muestra un lista de las secciones del menú, se puede observar el nombre de casa sección, descripción y la imagen, y utilizar el botón de editar para modificar los datos. También tienes la opción de eliminar secciones si ya no son necesarias
3. Reservas: en esta tabla se encuentra registradas todas las reservas realizadas por los clientes. Puede ver información como el nombre, email, teléfono, fecha y hora.
4. Comentarios: aquí se muestra los comentarios dejados por los usuarios en la web. Puede ver el contenido de cada uno de ellos y, si es necesario, inhabilitarlo utilizando un botón específico. Esto permite gestionar los comentarios y evitar aquellos que sean inapropiados.
5. Noticias: En esta tabla se listan todas las noticias o anuncios publicados en tu sitio web. Puedes ver el título, la fecha de publicación y otro detalle relacionado. También puedes eliminar noticias antiguas cuando sea necesario.
6. Formularios: Tienes dos formularios que te permiten añadir información a la base de datos de tu página web:
 1. Formulario de secciones: Aquí puedes agregar nuevas secciones al menú. Puedes introducir los datos y guardarlos en la base de datos.
 2. Formulario de platos: Con este formulario puedes añadir nuevos platos al menú. Puedes ingresar el nombre del plato, una descripción, el precio y la sección a la que pertenece. Al igual que con las secciones, puedes guardar estos datos en la base de datos.

A continuación se mostrará como he realizado algunas de estas funcionalidades.

Tablas de Comentarios.

En primer lugar nos creamos un componente Administartion.vue en el cual introduciremos toda las funcionalidades, dentro de éste creamos 2 partes, una parte de HTML donde crearemos las tablas y los formularios y otra para la parte de JavaScript.

```

</template>...
</template>

<script>
import axios from 'axios';
import Layout from '@Layouts/Layout.vue';
import { ref, watchEffect } from 'vue';
import { Bar, Pie } from 'vue-chartjs'
import { Chart as ChartJS, Title, Tooltip, ArcElement, Legend, BarElement, CategoryScale, LinearScale } from 'chart.js'
import Swal from 'sweetalert2';
import { reactive } from 'vue';

ChartJS.register(Title, Tooltip, ArcElement, Legend, BarElement, CategoryScale, LinearScale);

export default {...
};

</script>

```

En segundo lugar comenzamos con la creación de la tabla.

```

<div class="m-4 mb-12">
  <h1 class="font-semibold text-2xl mb-4">Comentarios</h1>

  <table class="min-w-full border-collapse block md:table">
    <thead class="block md:table-header-group">
      <tr class="border border-grey-500 md:border-none block md:table-row">
        <th class="th">Nombre</th>
        <th class="th">Opinion</th>
        <th class="th">Fecha</th>
        <th class="th">Eliminar</th>
        <th class="th">Inhabilitar</th>
      </tr>
    </thead>
    <tbody class="block md:table-row-group">
      <tr class="trImpar" v-for="(datos, index) in paginatedComentarios" :key="datos.id">
        <td class="td"><span class="span">Nombre</span>{{ datos.name }}</td>
        <td class="td"><span class="span">Opinion</span>{{ datos.comments }}</td>
        <td class="td"><span class="span">Fecha</span>{{ formatDate(datos.created_at) }}</td>
        <td class="td"><button class="bg-red-600 p-2 text-white rounded text-sm uppercase"
          @click="eliminarComentario(datos.id)">Eliminar</button></td>
        <td class="td"><button class="bg-red-600 p-2 text-white rounded text-sm uppercase"
          @click="desactivarComentario(datos.id)">Inhabilitar</button></td>
      </tr>
    </tbody>
  </table>

```

En tercer lugar, en la parte de JavaScript realizamos la petición GET la cual solicita la obtención de datos del servidor a la ruta específica. Utilizamos watchEffect para realizar un seguimiento de los cambios de una variable y ejecutar una función cada vez que cambie, con esto mantenemos la interfaz del usuario actualizada y sincronizamos los datos en tiempo real.

```

setup() {
  const comentariosPersonas = ref([]);

  watchEffect(() => {
    axios.get('/opiniones')
      .then(response => {
        comentariosPersonas.value = response.data;
        // console.log(comentariosPersonas);
      })
      .catch(error => {
        console.log('Error al obtener los datos:', error);
      });
  });
}

```

A continuación en el fichero Router.php insertaremos la ruta que especifiquemos en la petición. Éste archivos dirige las solicitudes del cliente a diferentes controladores y acciones correspondientes. En este caso realiza la acción mostrar del controlador ValorationController.

```

Route::get('/opiniones',[ValorationController::class, 'mostrar']);
Route::post('/opiniones',[ValorationController::class, 'mostrar']);

```

Y por último el fichero ValorationController.php el cual realizara un consulta a la base de datos para recoger los datos específicos, en este caso esta recogiendo los datos los cuales en el campo desactived es igual a 0 ordenándolos por la fecha de manera descendente.

```

/**
 * Muestra todas las opiniones visibles ordenadas
 */
public function mostrar()
{
    $valoracion = Valoration::select('id','name', 'comments','desactived','created_at')->where('desactived', 0)->orderByDesc('created_at')->get();
    return $valoracion;
}

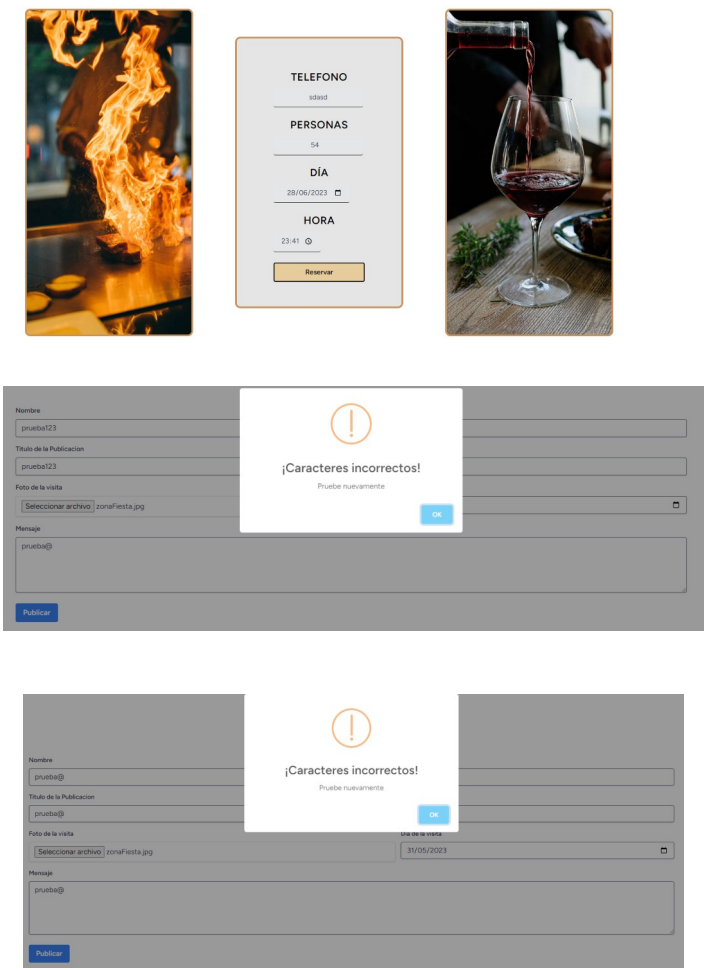
```

De esta manera se realiza la muestra de datos de todas tablas.

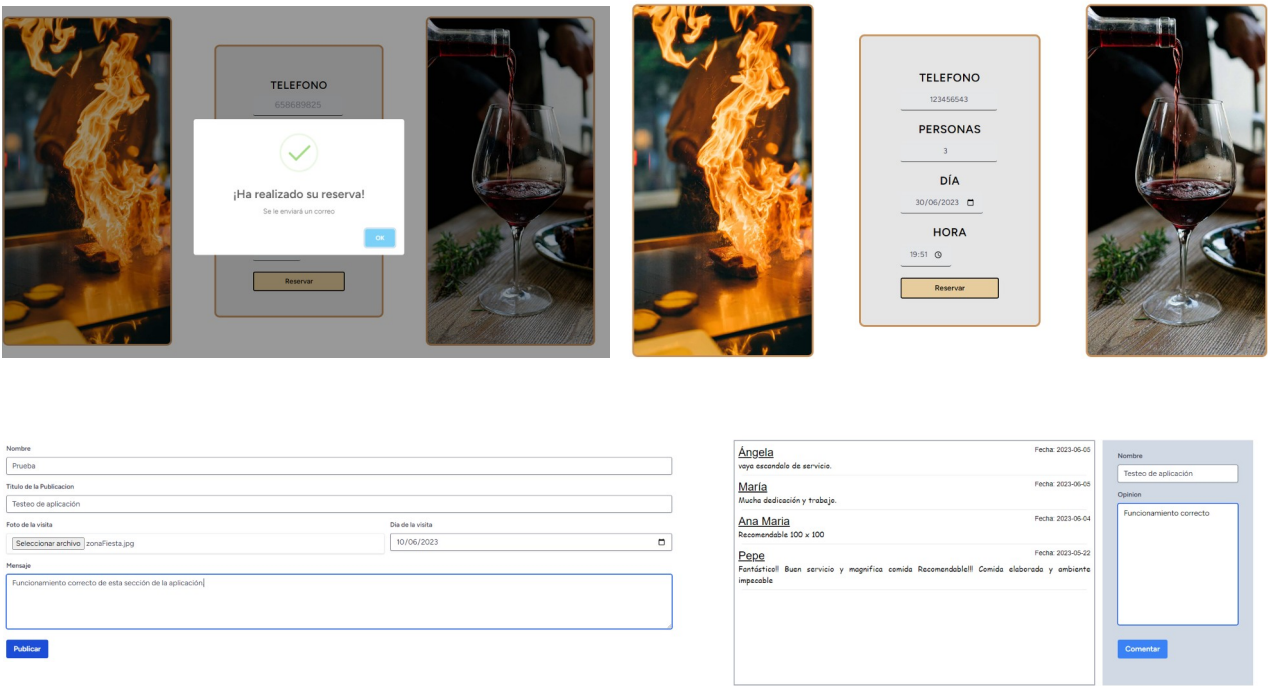
10. Evaluación y prueba

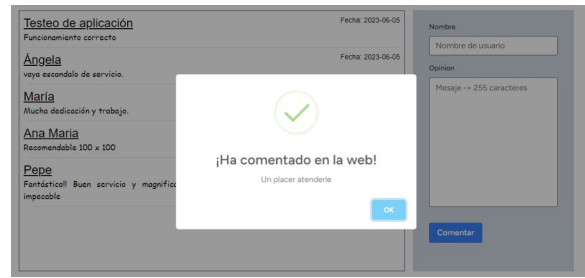
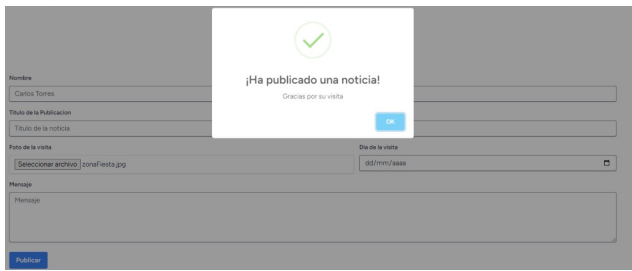
El testeo de la aplicaciones web como de la base de datos se ha realizado manualmente. En los distintos formularios de la web se han introducido valores incorrectos para comprobar que todos los errores se controlan y que la función de cada uno sea correcta. A continuación se les mostrarán el testeo de los diferentes formularios:

Errores al introducir erróneamente los datos



Funcionamiento correcto

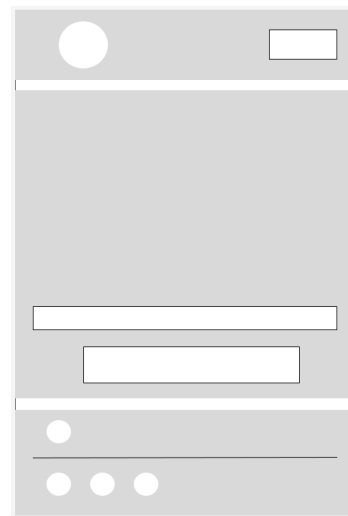
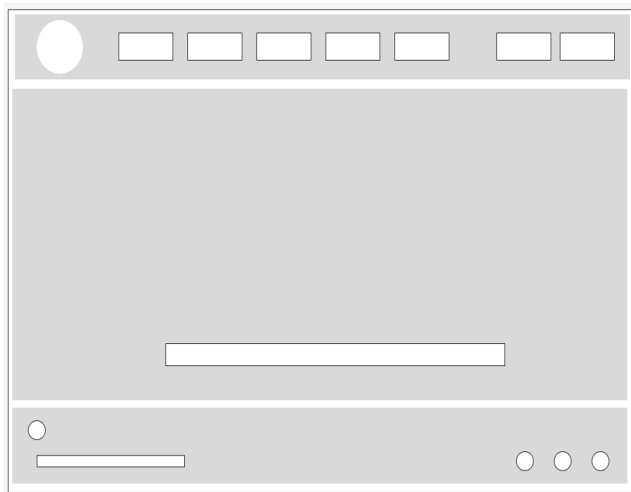




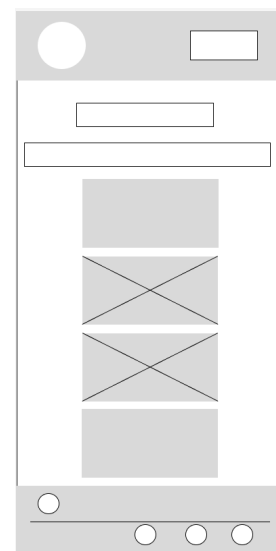
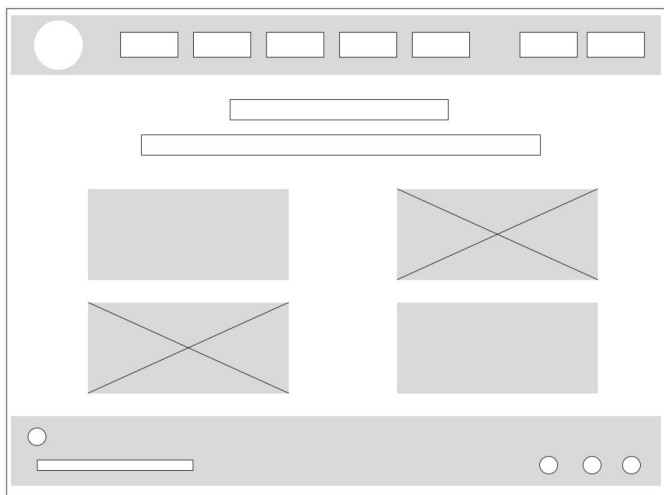
11. Manual de estilos en el que se recojan

Sketches:

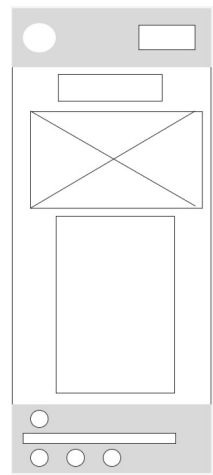
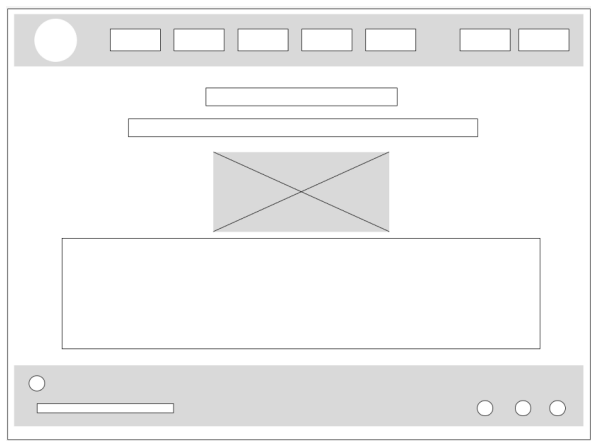
Home



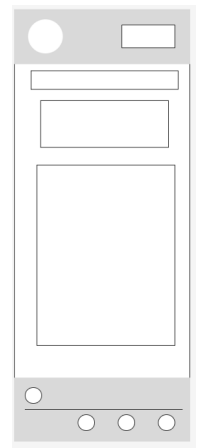
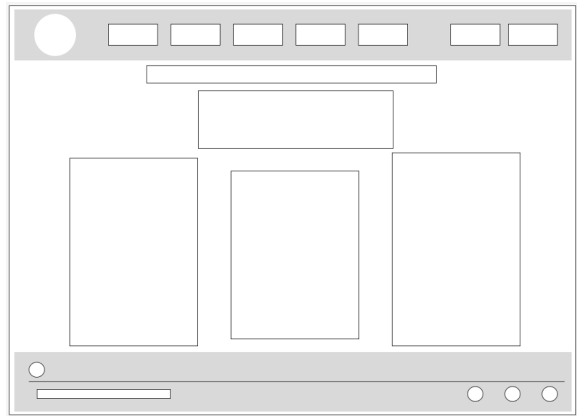
Sobre Nosotros



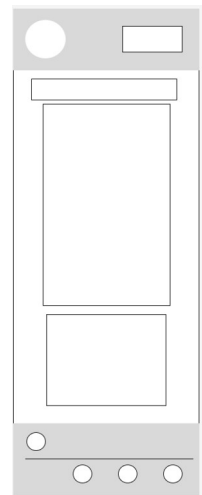
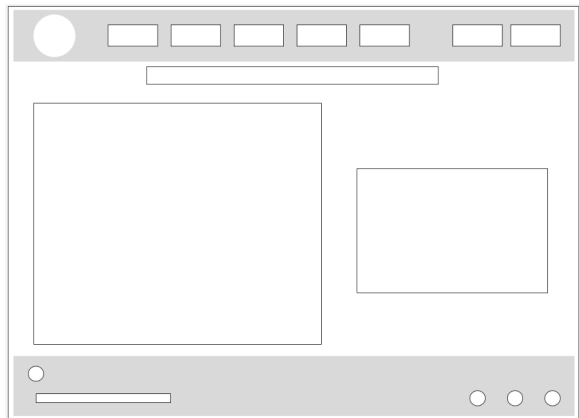
Menu



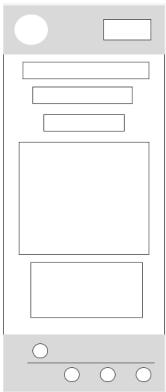
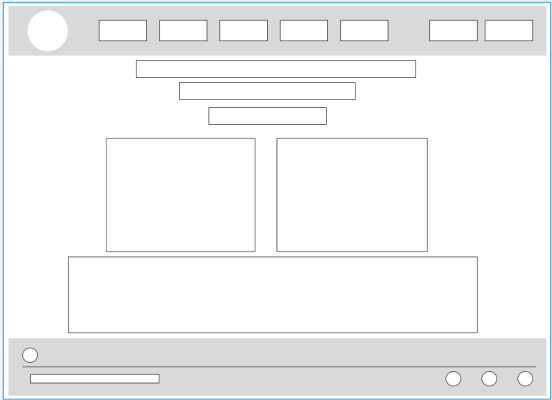
Reservas



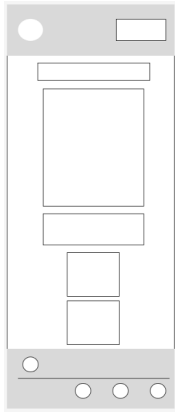
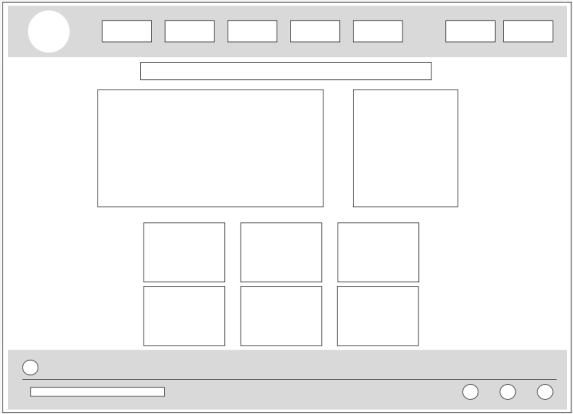
Ubicación



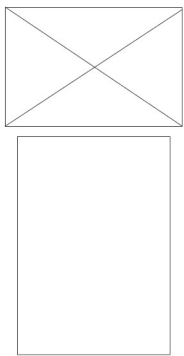
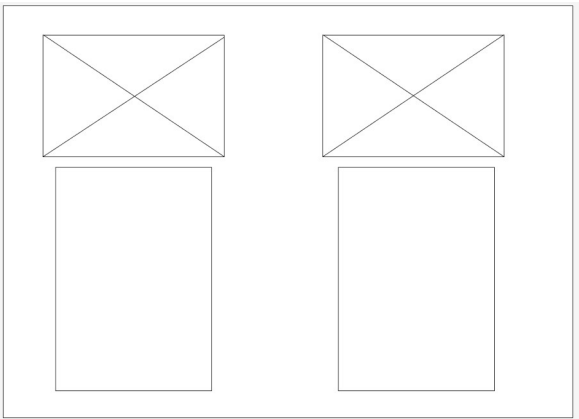
Blog



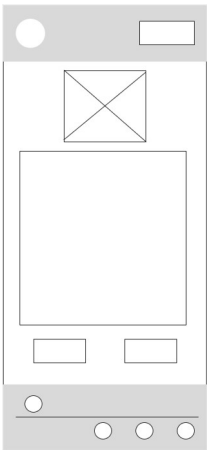
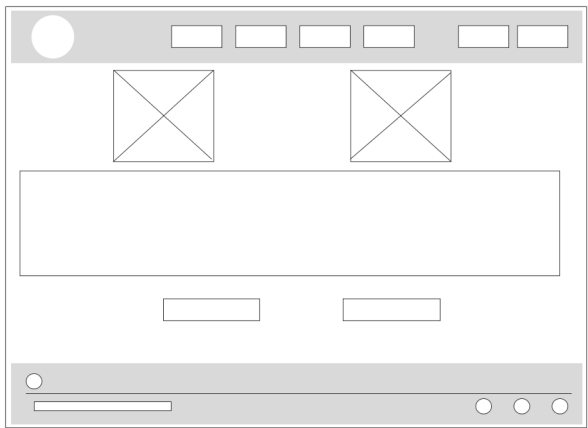
Opinión



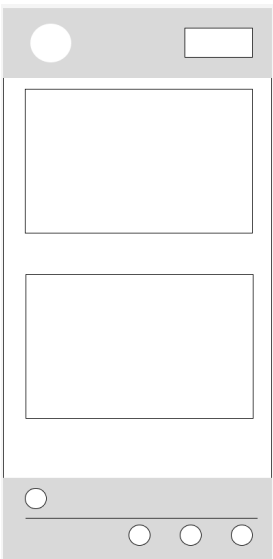
Login y registro



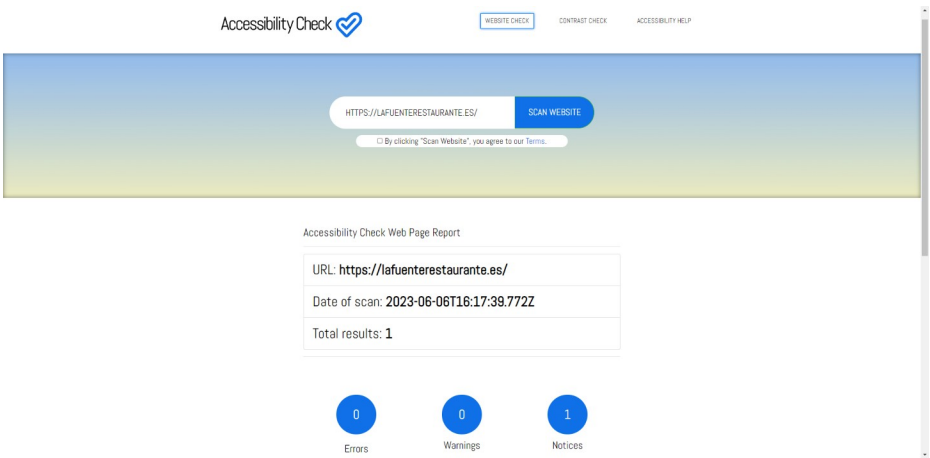
Administración



Perfil



11.2 Criterios de accesibilidad



Los criterios que se le pueden aplicar a esta web son:

1. Navegación clara: fácil de usar, con menús claros y estructurados, para que los usuarios puedan moverse sin dificultad y encontrar la información que necesitan.
2. Constaste de color: utilizo esta combinación de colores para que tengan contraste suficiente para que las personas con discapacidad visual o dificultades de visión pueda comprender el contenido fácilmente.
3. Tiempos de espera adecuados: evito utilizar tiempos de espera excesivamente cortos en redirecciones.
4. Facilidad de lectura: utilizo fuente legibles y tamaños adecuados, evito el uso excesivo de efectos visuales animaciones que puedan dificultar la lectura del contenido.

11.3. Criterios de usabilidad

La usabilidad web se refiere al grado de facilidad para acceder y navegar por una página de forma intuitiva, sencilla y rápida.

Éste indica la calidad de la misma: cómo de fácil es realizar las diferentes tareas de forma rápida y efectiva lo que, a su vez, afecta en la satisfacción del usuario.

Así, para determinar la usabilidad web se suele tener en cuenta:

- Eficiencia: el usuario logra cubrir sus expectativas/necesidades. Por ejemplo, ver los comentarios de otros clientes.
- Eficacia: el usuario consigue su objetivo en un tiempo breve. Por ejemplo, realizar una reserva o ver las noticias de diferentes clientes.
- Satisfacción: el usuario considera que su navegación por el sitio web fue sencilla, rápida y eficiente.

11.4. Fuentes principales y secundarias



Esto es un ejemplo donde uso 2 de las 3 fuentes de las web.

- Cabin

- Yellowtail

Fuentes principales → oswald

Fuentes secundarias → cabin, yellowtail,

11.5 Tamaños principales. Elementos donde se aplican.

Los tamaños principales al usar tailwind son:

- 3xl (text-3xl) equivale a 30px
- xl (text-xl) equivale a 20px
- base (text-base) equivale a 16px

Elementos donde se aplican:

Opiniones

Ángela

vaya escandalo de servicio.

Fecha: 2023-06-05

Blog

NOTICIAS

Muestra tus fotos de la visita al mejor restaurante de Andalucía!

Rellena tu Noticia

Menú

POLLUELA

Pollo, ajo, laurel, vino ...9.50€

SECRETO

Secreto de cerdo iberico, patatas, sal ...13.50€

—ENSALADA—

EXCLUSIVA Y DELICIOSAS ENSALADAS

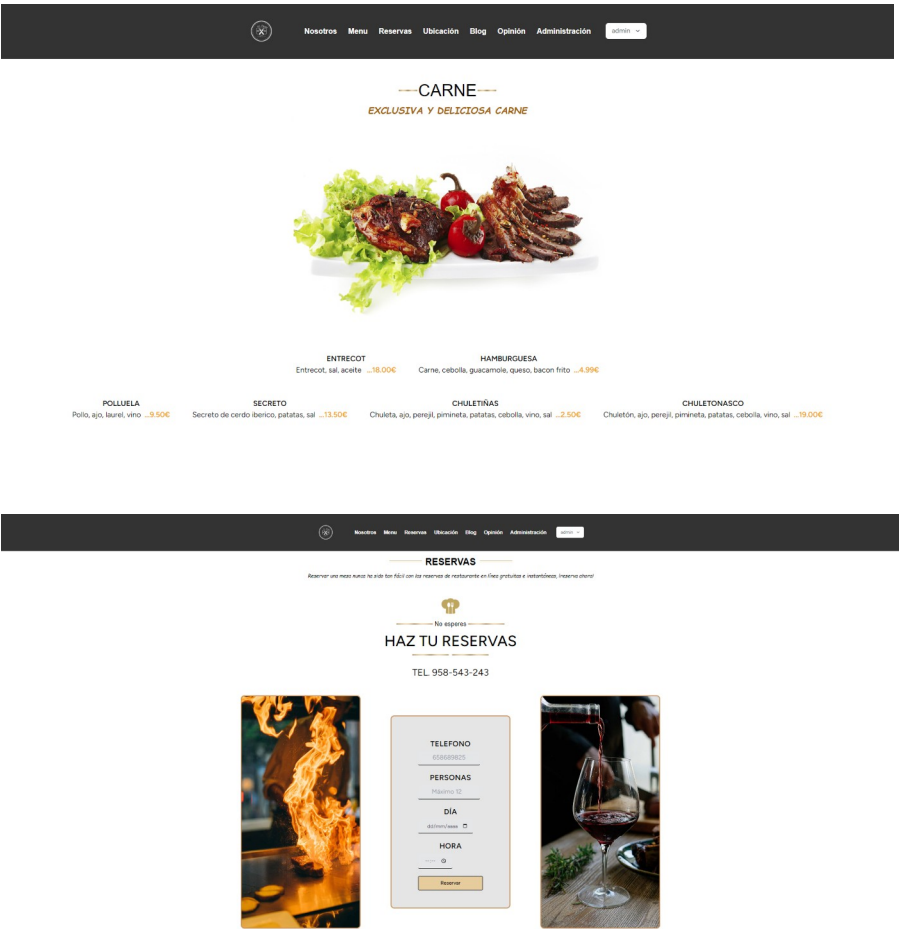
11.6 Mapa de colores de proyecto en tres formatos (RGB, Hexadecimal, nombre junto con el color). Elementos donde se aplica.

Estos colores se escogieron por el logo, junto con otro color secundario dorado



RGB	Hexadecimal	Nombre del color
0,0,0	#000000	Negro
255.255.255	#ffffff	Blanco
166,168,173	#a6a8ad	Gris
249,189,58	#F9BD3A	Dorado

Elementos donde se aplican:



11.7. Dispositivos/vistas para las que se ha diseñado el proyecto.

La web esta diseñada para todos los dispositivos, ya sea móvil, tablets, ordenador.

Puntos de Ruptura móvil → 320px – 425px



Puntos de Ruptura tablets → 768px



Puntos de Ruptura laptop → 1024px - 1440px



12. Software utilizado

1. Visual Studio Code: para implementar el código de la web y también para iniciar el servidor de Apache para mostrar la Web en local.
2. Gimp: Modificación de imágenes
3. Xampp: para gestionar la base de datos MySQL
4. GitHub: utilizado para guardar las diferentes modificaciones hechas en el código.

13. Mejoras posibles y aportaciones

La única posible mejora sería añadir la funcionalidad de realizar un pedido, con lo cual el cliente no necesitaría dirigirse al restaurante para poder degustar los diferentes platos que poseen. Y por otra parte mejorar el SEO sería una muy buena mejora ya que mejoraría la visibilidad y el posicionamiento del sitio web en los resultado de búsqueda como Google.

Licencias

En la página web del restaurante no necesita ninguna licencia, ya que las imágenes son sin copyright.

Bibliografía

Imágenes → <https://www.freepik.es/>

Laravel → <https://laravel.com/>

Vue.js → <https://vuejs.org/>

Tailwind → <https://tailwindcss.com/>

Accesibilidad a la web → <https://accessibilitycheck.org/>

Paleta de colores → <https://colorkit.co/color-palette-generator/000000-a3a3a3-ffffff-d9c45d/>

Tipografía → <https://fonts.google.com/>