

# **HITO 1 – DESARROLLO WEB EN ENTORNO CLIENTE**



**ALVARO BARRENA REVILLA**

**28/11/2023**

# ÍNDICE

<b>P1:</b> Identificar el propósito y los tipos de DNS, incluyendo explicaciones sobre cómo se organizan y gestionan los nombres de dominio.....	3
<b>P2</b> Explicar la finalidad y las relaciones entre los protocolos de comunicación en relación con el diseño, la publicación y el acceso a un sitio web.....	4
<b>P3</b> Discutir las capacidades y relaciones entre las tecnologías de frontend y backend de sitios web y explicar cómo se relacionan con las capas de presentación y aplicación.....	7
<b>P4</b> Analizar las diferencias entre las herramientas de creación de sitios web en línea y los sitios creados a medida.....	10
<b>P5</b> Crear un documento de diseño para un sitio web de marca, multipágina, apoyado con wireframes de fidelidad media y un conjunto completo de requisitos del cliente y del usuario.....	12
<b>P6</b> Utilizar el documento de diseño con los principios, estándares y directrices apropiados para producir un sitio web de marca, multipágina, apoyado con contenido realista.....	17
1. Introducción .....	17
2. Desarrollo del Proyecto .....	19
3. Descripción detallada en la parte Frontend .....	22
4. Resultados y Funcionalidades.....	23
5. Desafíos Enfrentados.....	24
<b>P7:</b> Crear y Utilizar un Plan de Pruebas para Revisar la Rendimiento y Diseño de un Sitio Web Multipágina.....	26
<b>M1</b> Analizar el impacto de las tecnologías y frameworks de desarrollo web más comunes en relación con el diseño, la funcionalidad y la gestión de sitios web. ....	28
<b>M2</b> Revisar la influencia de los motores de búsqueda en el rendimiento de los sitios web..	29
<b>M3</b> Analizar una serie de herramientas y técnicas disponibles para diseñar y desarrollar un sitio web a medida.....	31
<b>M4</b> Justificar las decisiones de implementación del sitio web multipágina con respecto al documento de diseño.....	33
<b>M5</b> Analizar la calidad aplicada durante el diseño de sitios web multipágina y fases de desarrollo.....	34
<b>D1:</b> Justificación de Tecnologías, Servicios de Gestión, Herramientas y Programas Informáticos para el Sitio Web DuBo Martial Arts .....	36
Tecnologías Web: .....	36
Servicios de Gestión de Datos: .....	36
Herramientas y Programas Informáticos: .....	36
Servidor y Certificado HTTPS: .....	37
Servicios Externos:.....	37

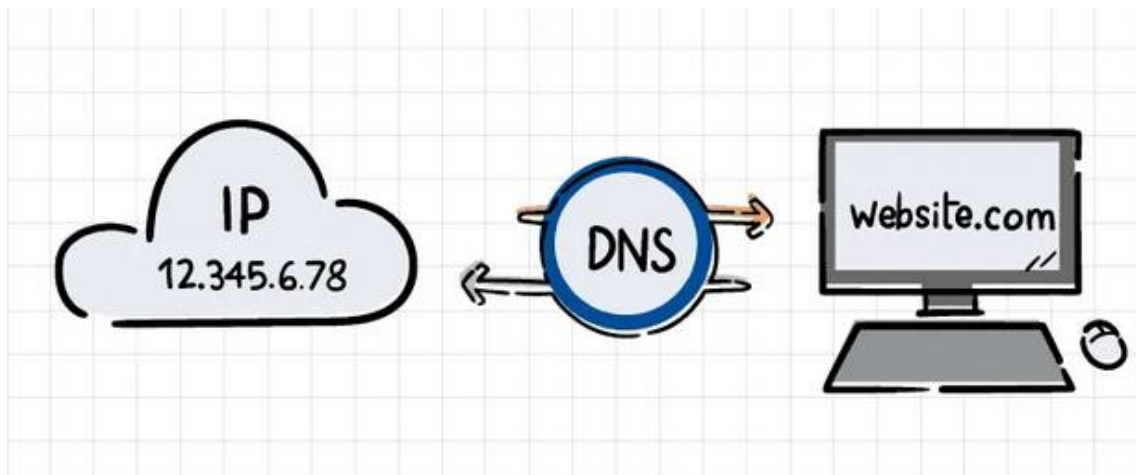
## **P1: Identificar el propósito y los tipos de DNS, incluyendo explicaciones sobre cómo se organizan y gestionan los nombres de dominio.**

Los Servidores de Nombres de Dominio (DNS) desempeñan un papel fundamental en la resolución de nombres de dominio a direcciones IP.

Su propósito principal es traducir los nombres de dominio legibles por humanos en direcciones IP que los dispositivos puedan comprender. Esto es esencial para permitir que los usuarios accedan a sitios web y servicios en línea.

Los tipos de DNS incluyen:

- **DNS Autoritativos:** son responsables de contener información sobre un **dominio específico**. Algunos tipos comunes son los DNS maestros (**primarios**) y los DNS esclavos (**secundarios**). Los DNS maestros son servidores que tienen autoridad para modificar y actualizar registros, mientras que los DNS esclavos replican esta información desde los DNS maestros.
- **DNS de Resolución:** se utilizan para **buscar información** de DNS en nombre del usuario. Se comunican con los servidores autoritativos para obtener información actualizada.

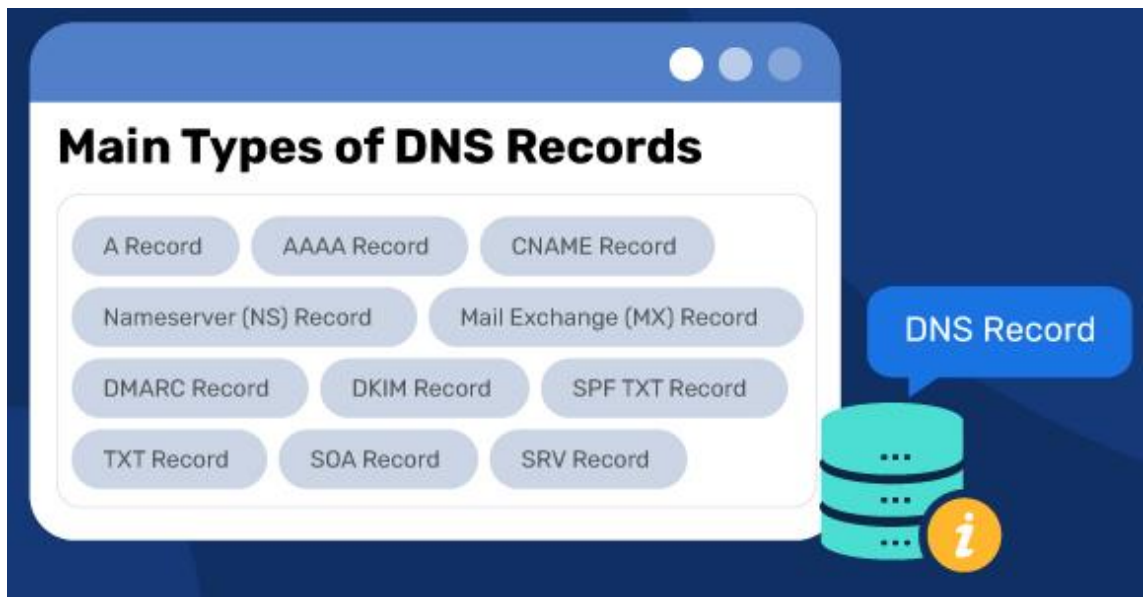


### **Organización y Gestión de Nombres de Dominio:**

La gestión de nombres de dominio implica la creación y actualización de registros DNS que asocian nombres de dominio con direcciones IP y otros datos. Estos registros incluyen:

- **Registro A:** Asocia un nombre de dominio a una dirección IP IPv4.
- **Registro AAAA:** Asocia un nombre de dominio a una dirección IP IPv6.
- **Registro MX:** Especifica servidores de correo para el dominio.
- **Registro CNAME:** Establece alias para nombres de dominio.
- **Registro TXT:** Almacena información de texto, como claves de verificación para servicios de correo electrónico.

- **Registro SOA (Start of Authority):** Contiene información autoritativa sobre la zona DNS.



La organización y gestión de nombres de dominio se lleva a cabo mediante registros en servidores DNS autoritativos. Los administradores de dominio pueden realizar cambios en estos registros según sea necesario para redirigir el tráfico, gestionar correos electrónicos, habilitar servicios y garantizar el acceso a sitios web. Además, las actualizaciones de DNS se propagan a través de Internet a medida que los servidores DNS de resolución buscan y obtienen información actualizada de los servidores autoritativos, asegurando la coherencia y la disponibilidad de los nombres de dominio en toda la web.

## **P2 Explicar la finalidad y las relaciones entre los protocolos de comunicación en relación con el diseño, la publicación y el acceso a un sitio web**

En el diseño, publicación y acceso a un sitio web, es fundamental comprender la interacción de varios componentes clave que colaboran para que el sitio web funcione de manera eficiente y segura:

- **Protocolos de Comunicación:** definen las **reglas** y el **formato** de cómo se comunican los clientes (navegadores web) y los servidores web.

Uno de los protocolos más utilizados es HTTP, que permite a los navegadores **solicitar recursos**, como páginas web, y a los servidores web responder con esos recursos.

HTTPS es una extensión segura de HTTP que utiliza cifrado SSL/TLS para proteger la **confidencialidad** y la **integridad** de los datos transmitidos.

Estos protocolos son **esenciales** para garantizar que las solicitudes de los usuarios se entreguen al servidor web y que las respuestas se envíen a los navegadores de manera segura y eficiente.



- **Hardware del Servidor:** comprende la **infraestructura física o virtual** que aloja y ejecuta el sitio web. Esto incluye componentes como procesadores, memoria, almacenamiento y adaptadores de red.

La elección del hardware adecuado depende de las **necesidades** del sitio web, como la cantidad de tráfico esperado, la complejidad de las aplicaciones y la escalabilidad requerida.

Por ejemplo, un sitio web con un gran volumen de tráfico podría requerir servidores de alta potencia y balanceo de carga para distribuir la carga entre **múltiples servidores**.



- **Sistemas Operativos:** actúan como **intermediarios** entre el hardware del servidor y el software del servidor web.

Son responsables de **administrar** los recursos del servidor, programar tareas, gestionar la memoria y garantizar la seguridad.

Los sistemas operativos comunes para servidores web incluyen Linux y Windows Server. La elección del sistema operativo también depende de factores como la compatibilidad con el software del servidor web y la preferencia del administrador del servidor.



- **Software del Servidor Web:** es la aplicación que **gestiona** las solicitudes entrantes de los clientes y entrega las respuestas adecuadas.

Algunos Ejemplos de software de servidor web son **Apache, Nginx y Microsoft IIS**.

Estos servidores web **procesan solicitudes**, sirven contenido estático (como imágenes y archivos HTML que no cambian) y dinámico (que se genera en tiempo real según la interacción del usuario), y pueden ejecutar scripts para generar páginas web dinámicas (aquellas que se generan de forma personalizada en respuesta a las acciones de los usuarios).





Además, los servidores web a menudo se integran con **módulos de seguridad** y opciones de configuración avanzadas para garantizar un funcionamiento seguro y eficiente del sitio web.

### **P3 Discutir las capacidades y relaciones entre las tecnologías de frontend y backend de sitios web y explicar cómo se relacionan con las capas de presentación y aplicación.**

Las tecnologías de frontend y backend en el desarrollo de sitios web desempeñan roles distintos pero complementarios

#### **Capacidades del frontend:**

- **Interfaz de Usuario (UI):** El frontend se enfoca en la presentación visual de un sitio web, incluyendo el diseño, la navegación y la interacción. Tecnologías como HTML, CSS y JavaScript se utilizan para crear páginas web atractivas y responsivas.
- **Experiencia del Usuario (UX):** El frontend se centra en mejorar la experiencia del usuario al proporcionar una interfaz de usuario intuitiva y fácil de usar. Las técnicas de diseño, animaciones y efectos visuales son comunes en esta capa.

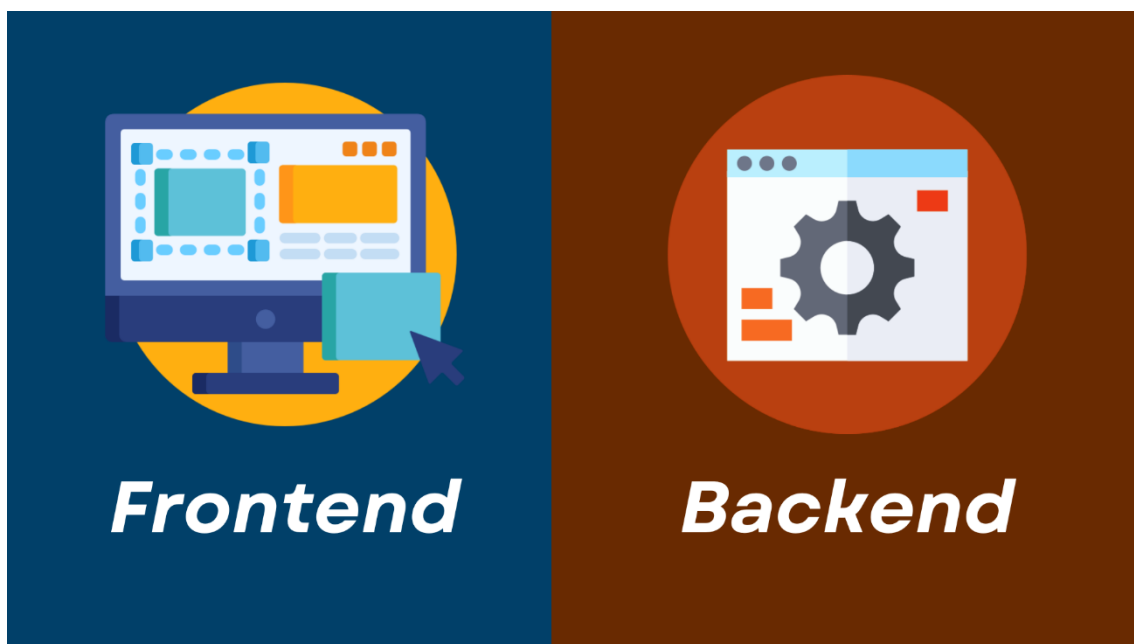
#### **Capacidades del backend:**

- **Lógica de Aplicación:** El backend maneja la lógica de la aplicación, procesando datos, gestionando usuarios, administrando bases de datos y realizando cálculos complejos. Utiliza lenguajes de programación como Node.js, Python o PHP.

- **Seguridad y Acceso a Datos:** La capa de backend es responsable de garantizar la seguridad de los datos y controlar el acceso a recursos sensibles. Autentica usuarios, gestiona sesiones y aplica medidas de seguridad como el cifrado y la autenticación.

#### Relaciones entre Frontend y Backend:

- **Comunicación:** se comunican a través de solicitudes y respuestas HTTP. El frontend envía solicitudes al backend para obtener o enviar datos, y el backend procesa estas solicitudes y devuelve datos al frontend.
- **Presentación y Lógica:** El frontend se encarga de la presentación visual, mientras que el backend se encarga de la lógica de la aplicación. La separación de estas capas permite una mayor modularidad y mantenibilidad del sistema.
- **Datos Dinámicos:** El frontend puede mostrar datos dinámicos obtenidos del backend. Por ejemplo, un sitio web de comercio electrónico mostrará detalles de productos almacenados en una base de datos en el backend.



#### **Capas de Presentación y Aplicación:**

- Capa de Presentación: es la parte de un sitio web con la que los usuarios interactúan directamente. Sus principales responsabilidades son:
  1. **Interfaz de Usuario (UI):** La capa de presentación se centra en la creación de una interfaz de usuario visualmente atractiva y funcional. Utiliza tecnologías como HTML para estructurar la información, CSS para diseñar la apariencia y JavaScript para agregar interactividad.



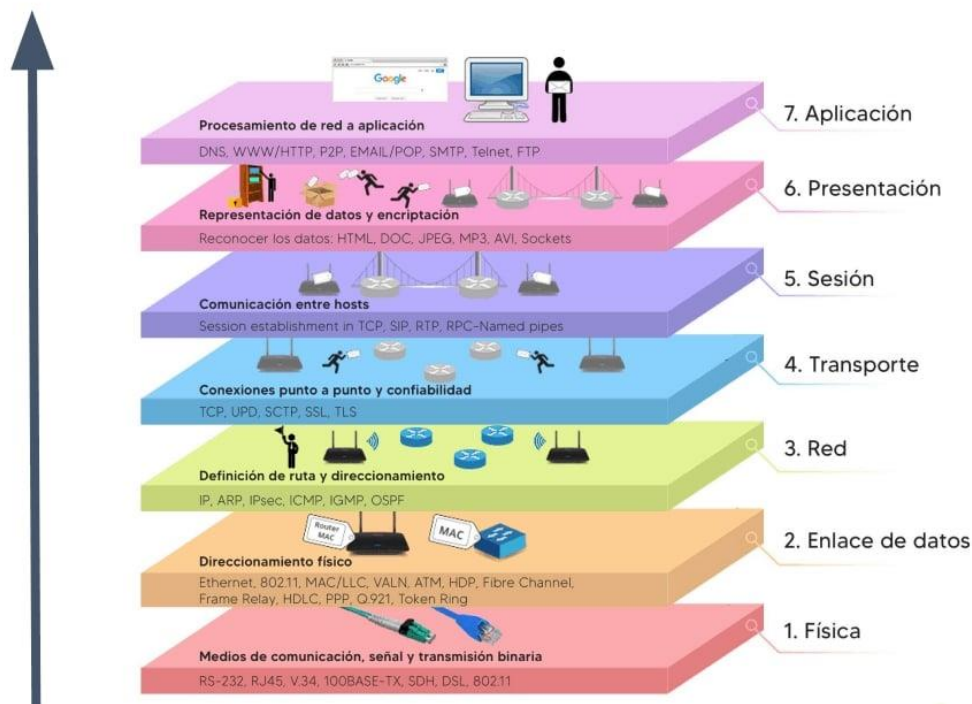
2. **Experiencia del Usuario (UX):** El objetivo principal de la capa de presentación es proporcionar una experiencia del usuario excepcional. Esto implica diseñar una navegación intuitiva, tiempos de carga rápidos y una presentación de la información que sea fácil de entender y utilizar.
3. **Interacción Usuario-Sitio:** El frontend permite la interacción directa del usuario con el sitio. Esto incluye hacer clic en enlaces, completar formularios, realizar búsquedas y experimentar animaciones o transiciones. La capa de presentación debe garantizar que estas interacciones sean fluidas y agradables.
4. **Adaptación a Dispositivos:** se encarga de la adaptación responsive, asegurando que el sitio web se vea y funcione de manera óptima en una variedad de pantallas y tamaños de dispositivos.

### **Capa de Aplicación:**

La capa de aplicación, conocida como backend, se encuentra detrás de escena y es responsable de gestionar la lógica de la aplicación y la gestión de datos. Sus funciones incluyen:

- **Lógica de Negocio:** El backend alberga la lógica de negocio de la aplicación. Esto implica tomar decisiones basadas en las acciones del usuario, procesar datos y realizar cálculos complejos. Por ejemplo, en un sitio web de comercio electrónico, la capa de aplicación manejaría el proceso de agregar productos al carrito, calcular el total de la compra y procesar el pago.
- **Seguridad y Autenticación:** El backend se encarga de la seguridad del sitio, autenticando a los usuarios, protegiendo los datos sensibles y aplicando medidas de seguridad, como la encriptación de datos y la prevención de ataques.
- **Acceso a la Base de Datos:** Para sitios web que almacenan información, el backend se comunica con la base de datos para realizar consultas, actualizaciones y recuperación de datos. Esto garantiza que la información esté disponible y actualizada.
- **Gestión de Sesiones:** En aplicaciones que requieren que los usuarios inicien sesión, el backend maneja la gestión de sesiones, lo que permite a los usuarios permanecer conectados de manera segura mientras interactúan con el sitio.
- **Comunicación con frontend:** La capa de aplicación responde a las solicitudes del frontend, procesa datos y devuelve respuestas, que se utilizan para actualizar la interfaz de usuario y proporcionar información dinámica.

La relación entre estas capas es esencial para el funcionamiento efectivo de un sitio web. El frontend se encarga de la presentación visual y la interacción, mientras que el backend maneja la lógica, la seguridad y la gestión de datos. Una comunicación fluida entre estas capas garantiza una experiencia de usuario coherente y eficiente, independientemente de la complejidad de la aplicación web.



**P4 Analizar las diferencias entre las herramientas de creación de sitios web en línea y los sitios creados a medida**

En el ámbito del desarrollo web, la elección entre herramientas de creación en línea y sitios web a medida implica considerar diversas características que impactan directamente en el diseño, rendimiento, funcionalidad, experiencia del usuario (UX) y la interfaz de usuario (UI) de un sitio. Estas diferencias son cruciales para la toma de decisiones informadas y la satisfacción de los requisitos específicos del proyecto.

	Herramientas en línea	Sitios a medida
Flexibilidad del Diseño	Plataformas como WordPress o Wix proporcionan una solución rápida y accesible, pero la flexibilidad está limitada a las opciones de las plantillas predefinidas.	Ofrecen una flexibilidad ilimitada, permitiendo adaptar cada elemento del diseño según los requisitos únicos del cliente.
Rendimiento	Puede verse afectado por la cantidad de complementos y	Permiten un rendimiento optimizado, con la posibilidad de aplicar

	funciones añadidas, con limitaciones en la optimización.	prácticas avanzadas para lograr tiempos de carga más rápidos.
<b>Funcionalidad</b>	Dependiendo de complementos, la funcionalidad puede ser amplia pero a veces menos personalizable.	Posibilitan el desarrollo de funcionalidades personalizadas y complejas adaptadas a los requerimientos específicos del proyecto.
<b>Experiencia del Usuario (UX):</b>	Puede depender de las plantillas y ser limitada en términos de optimización.	Permite diseñar una experiencia de usuario única y personalizada desde cero, considerando las necesidades y preferencias del público objetivo.
<b>Interfaz de Usuario (UI)</b>	La interfaz de usuario se encuentra dentro de los límites de las opciones de diseño de las plantillas.	Facilitan la creación de una interfaz de usuario única, adaptada a la identidad de la marca y con interacciones y animaciones personalizadas.



Con plantilla



A medida

## **P5 Crear un documento de diseño para un sitio web de marca, multipágina, apoyado con wireframes de fidelidad media y un conjunto completo de requisitos del cliente y del usuario.**

### **Requisitos del Cliente:**

El diseño del sitio web para DuBo Martial Arts se centrará en capturar la esencia distintiva de la marca y proporcionar una experiencia de usuario excepcional. La navegación intuitiva permitirá a los visitantes acceder fácilmente a información vital, incluyendo detalles sobre clases, perfiles de instructores y eventos. Se priorizará la facilidad de gestión del contenido, garantizando que los administradores del sitio puedan mantener la información actualizada y relevante. La integración fluida con las redes sociales será esencial para fomentar la participación de la comunidad, y se incluirá una sección clara para inscripciones en clases y eventos.

### **Requisitos del Usuario:**

El diseño se enfocará en la creación de una interfaz intuitiva que satisfaga las necesidades de los usuarios. Se proporcionarán detalles exhaustivos sobre las clases de artes marciales, incluyendo horarios y requisitos de habilidad. Los perfiles de instructores destacarán las habilidades y experiencias únicas de cada profesional. Un calendario de eventos interactivo mantendrá a los usuarios informados sobre actividades futuras, y una sección educativa ofrecerá recursos valiosos para aquellos interesados en profundizar en las artes marciales.

### **Creación de los Wireframes:**

A continuación comparto el pdf en el que se podrán encontrar todos los wireframes del proyecto y posteriormente comenzará a explicar las características de cada uno de ellos

[Visualiza aquí los wireframes](#)

#### **1. Página de Inicio (será index.html en la web)**

##### ***Experiencia del Usuario (UX):***

La página de inicio de "DuBo Martial Arts" proporciona una experiencia inicial envolvente. La barra de navegación es intuitiva, facilitando la transición entre secciones. La sección de bienvenida en la parte superior utiliza una llamada a la acción impactante, invitando a los usuarios a explorar más.

##### ***Recomendaciones:***

- **Responsividad Móvil:** Aunque la página es visualmente atractiva, se podría mejorar para garantizar una experiencia perfecta en dispositivos móviles.

- **Testimonios y Comentarios:** Incorporar testimonios de alumnos para agregar autenticidad y persuasión.

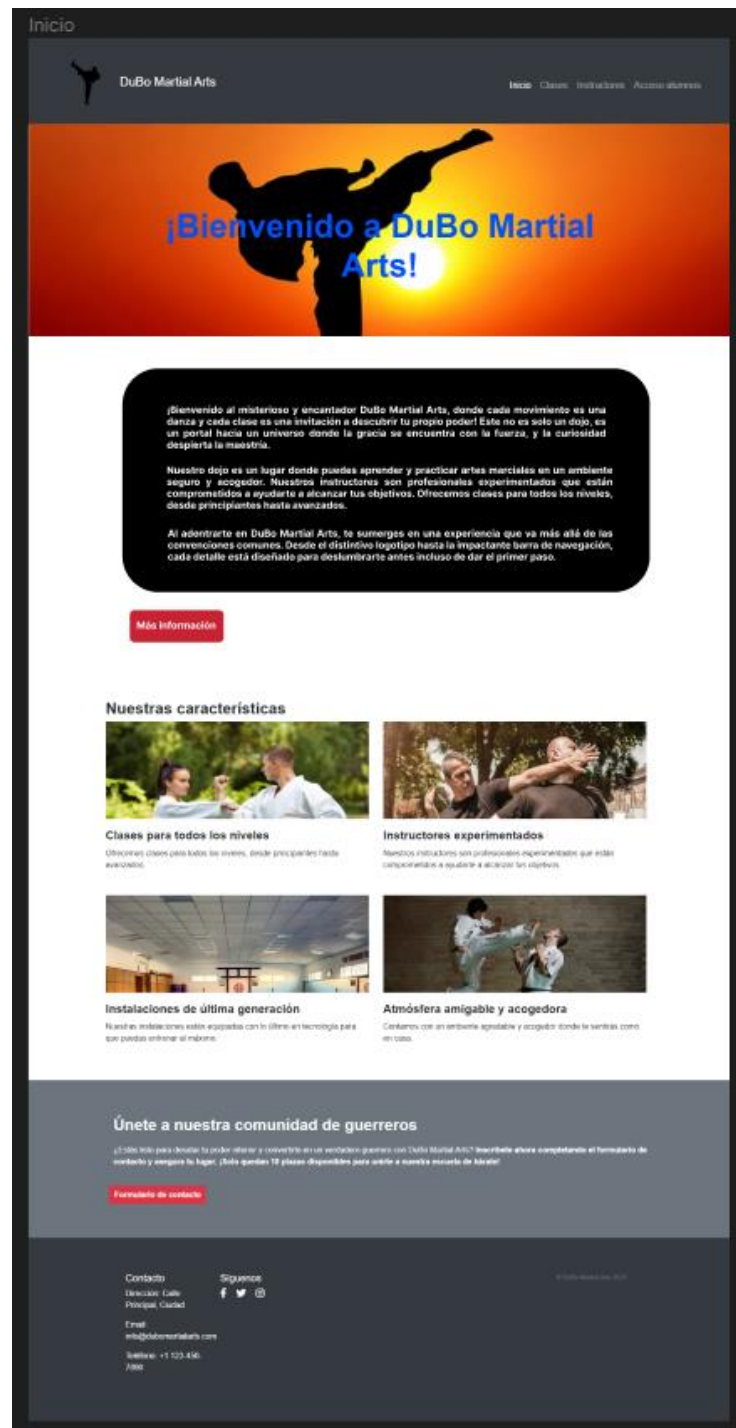


Imagen 1: Wireframe de la página de Inicio

## 2. Página de Clases (será clases.html en la web)

### **Experiencia del Usuario (UX):**

La página de clases presenta de manera efectiva la información sobre las ofertas de clases. Las imágenes proporcionan una vista previa visual, y la disposición general es clara y accesible.

### **Recomendaciones:**

- **Galería de Imágenes:** Agregar una galería de imágenes para cada clase, permitiendo a los usuarios visualizar las instalaciones y actividades.
- **Horarios de Clases:** Incluir un horario detallado para una comprensión clara de las opciones de clases.

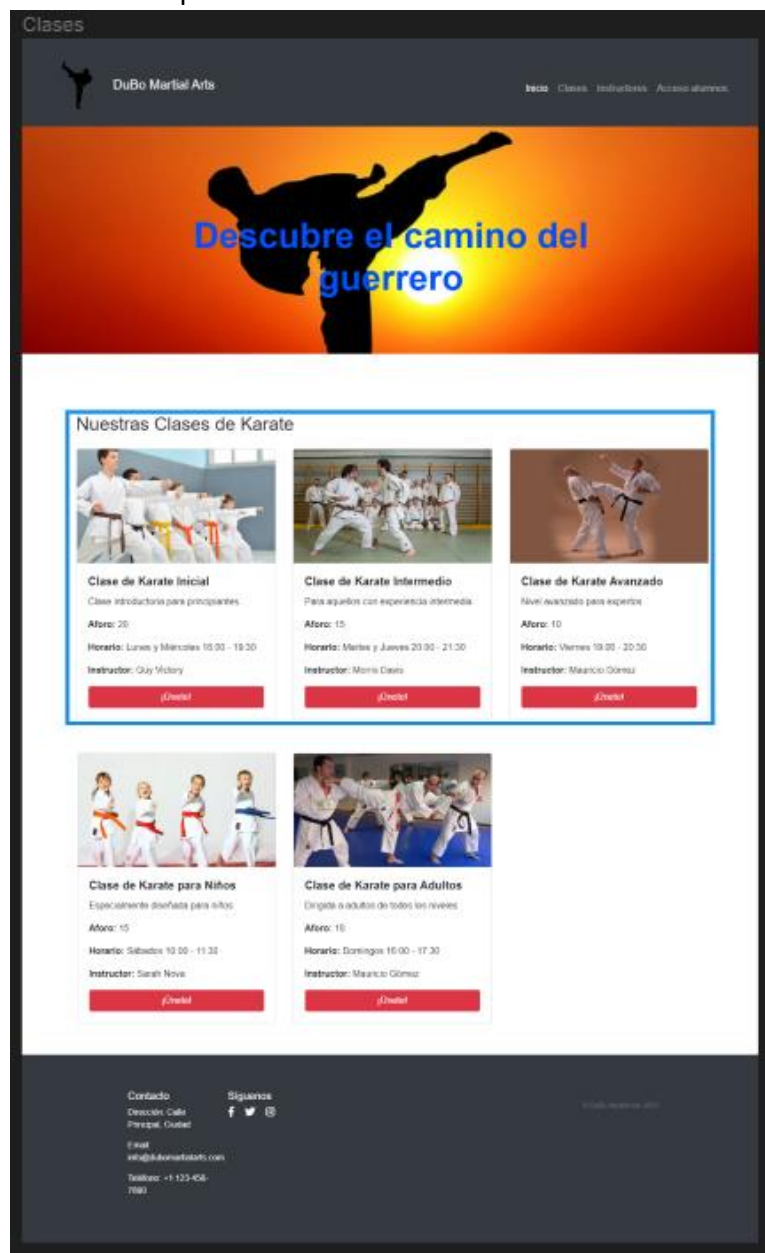


Imagen 2: Wireframe de la página de clases



### 3. Página de Instructores (será instructores.html en la web)

#### **Experiencia del Usuario (UX):**

La sección de instructores destaca a los líderes del dojo de manera efectiva. Las imágenes y la información clave son visibles, y la barra de pie de página ofrece detalles adicionales.

#### **Recomendaciones:**

- **Biografías de Instructores:** Agregar breves biografías o logros para establecer una conexión emocional con los usuarios.
- **Programas Dirigidos:** Incluir información sobre programas específicos liderados por instructores.

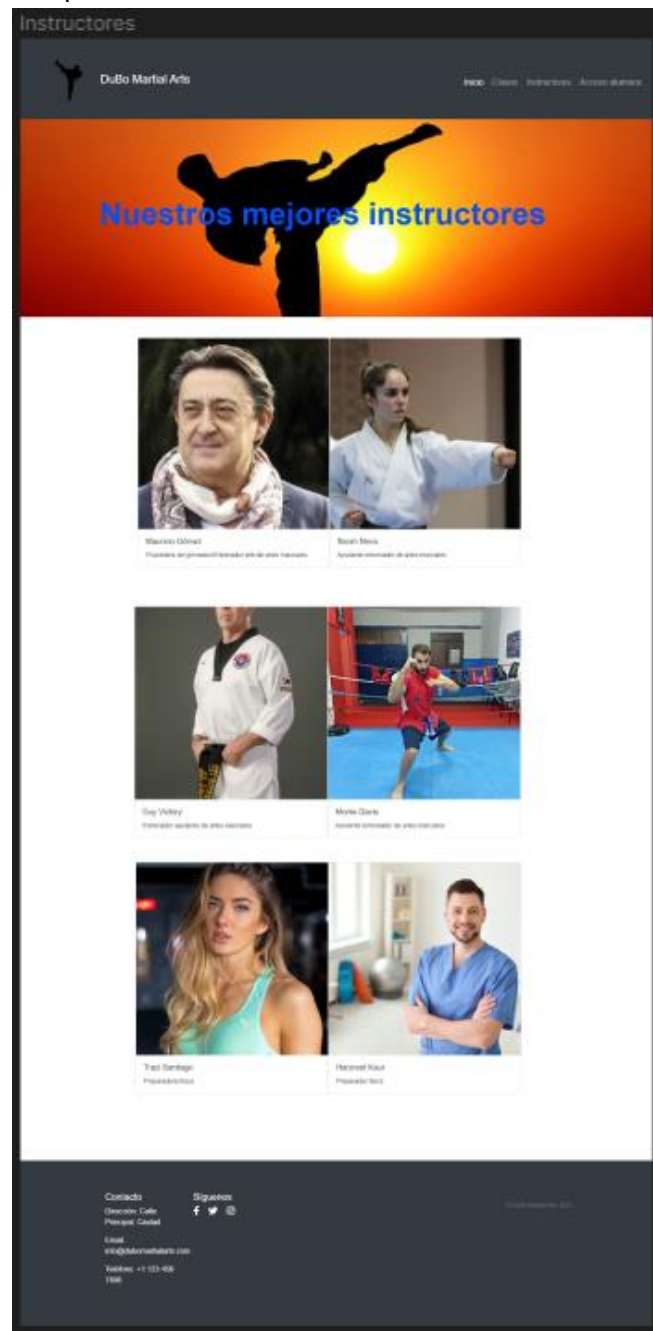


Imagen 3: Wireframe de la página de Instructores

#### 4. Página de Inicio de Sesión (será login.html en la web)

##### **Experiencia del Usuario (UX):**

La página de inicio de sesión ofrece dos opciones claras: "Crear una cuenta" e "Iniciar Sesión". La incorporación de iconos de redes sociales agrega una capa adicional de conveniencia.

##### **Recomendaciones:**

- **Seguridad de Datos:** es importante almacenar el registro/inicio de sesión con un token para evitar visualizar los datos en texto plano.
- **Enlaces Informativos:** Proporcionar enlaces a políticas de privacidad y términos de servicio para la transparencia.

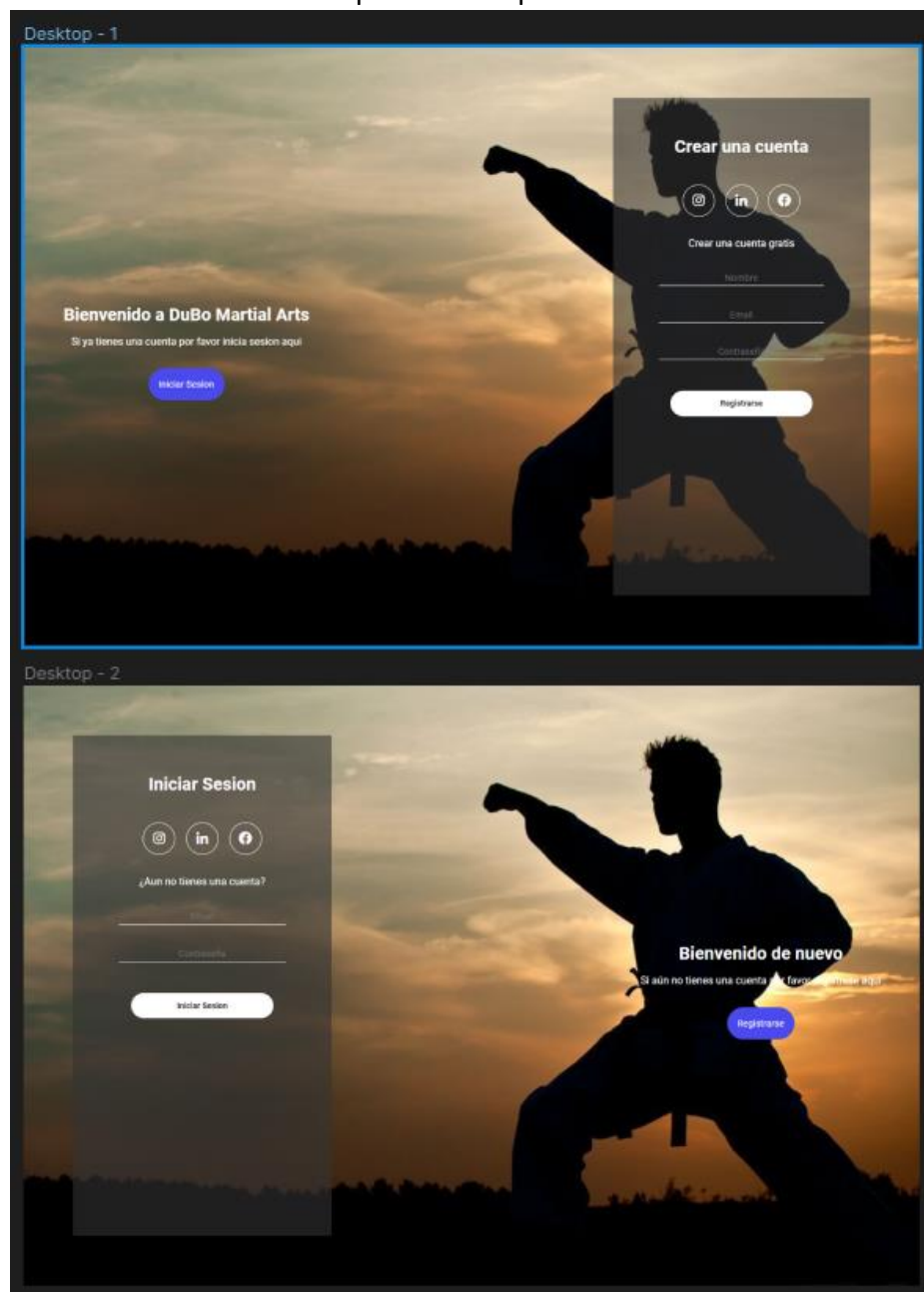


Imagen 4: Wireframe de la página de inicio de sesión y registrarse

## Características Comunes:

- **Consistencia en la Navegación:** La barra de navegación mantiene consistencia en todas las páginas, facilitando la navegación del usuario.
- **Estética Coherente:** La estética general del sitio es coherente con la temática del dojo, creando una experiencia visualmente agradable.

## Sugerencias Globales:

- **Optimización para Dispositivos Móviles:** Asegurar que todas las páginas estén totalmente optimizadas para dispositivos móviles, garantizando una experiencia uniforme.
- **Elementos Interactivos:** Considerar la inclusión de elementos interactivos para aumentar la participación del usuario.
- **Análisis de Métricas:** Implementar herramientas de análisis web para comprender mejor el comportamiento del usuario y ajustar estrategias en consecuencia.

## P6 Utilizar el documento de diseño con los principios, estándares y directrices apropiados para producir un sitio web de marca, multipágina, apoyado con contenido realista.

### 1. Introducción

El proyecto DuBo Martial Arts se ha concebido con el objetivo principal de proporcionar una experiencia en línea integral para los estudiantes y posibles nuevos miembros del dojo. La plataforma web desarrollada tiene como propósito no solo informar sobre las clases y los instructores, sino también involucrar a la comunidad a través de funciones interactivas y contenido relevante. Este informe ofrece una visión detallada de los aspectos clave del desarrollo y la implementación de este proyecto.

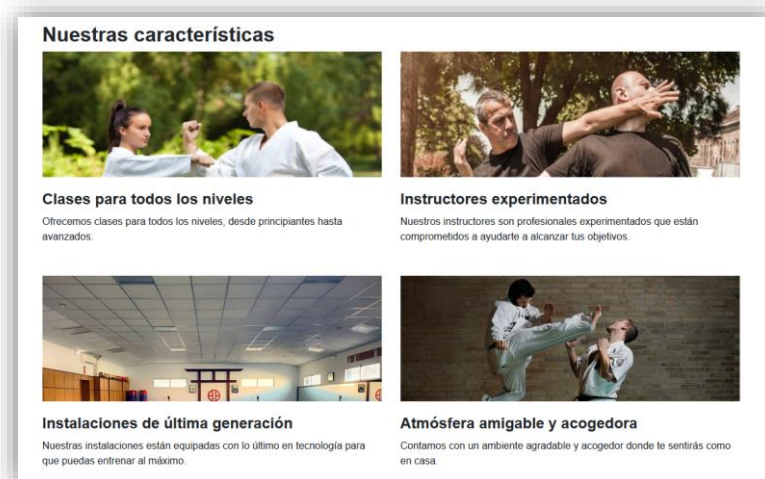


## 1.1. Descripción del proyecto

DuBo Martial Arts es un dojo de renombre que ha decidido llevar su presencia a la web para satisfacer las crecientes demandas de la era digital. El sitio web interactivo desarrollado no solo presenta información básica sobre las clases y los instructores, sino que también permite a los alumnos acceder a recursos exclusivos, participar en eventos y mantenerse conectados con la comunidad.

## 1.2. Objetivos

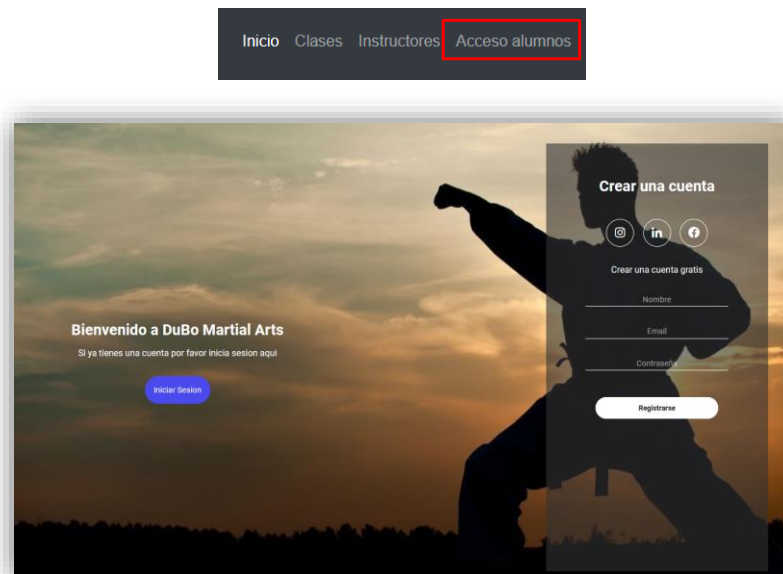
- **Creación de una Plataforma Atractiva y Funcional:** El diseño del sitio web se enfoca en atraer a nuevos estudiantes mediante un diseño atractivo y una interfaz fácil de usar. Además, se busca mejorar la experiencia de los alumnos actuales proporcionando recursos valiosos en línea.



- **Información Detallada sobre Clases e Instructores:** El sitio incluye secciones específicas, como "Clases" y "Instructores", que ofrecen detalles completos sobre los diferentes tipos de clases disponibles, horarios, y perfiles detallados de los instructores.



- **Facilitar el Acceso de Alumnos:** Se ha implementado una sección de "Acceso Alumnos" que permite a los estudiantes registrados acceder a material exclusivo, como videos de entrenamiento, materiales educativos y comunicarse con otros miembros.



## 2. Desarrollo del Proyecto

### **2.1 Tecnologías Utilizadas**

El proyecto hace uso extensivo de tecnologías web modernas para garantizar un rendimiento óptimo y una experiencia de usuario fluida.

Primero está el proyecto del servidor, que se usa como “CDN” local para brindar una base de datos en formato JSON:

```
const express = require('express');
const cors = require('cors');
const DB = require('./db.json');

const app = express();
const PORT = process.env.PORT || 3001;

// Configuración CORS
app.use(cors());

app.use(express.json());

app.get('/usuarios', (req, res) => {
  req.DB = DB;
  return res.json(req.DB.usuarios);
});

app.get('/instructores', (req, res) => {
  req.DB = DB;
  return res.json(req.DB.instructores);
});

app.get('/clases', (req, res) => {
  req.DB = DB;
  return res.json(req.DB.clases);
});

app.listen(PORT, () => {
  console.log(`App listening on port ${PORT}`);
});
```

Imagen 5: Implementación del servidor para la crear la API

En este archivo JavaScript, utilizado para crear el servidor web en Express creo toda la lógica para implementar la base de datos en JSON.

Hago redirecciones según el tipo de API que necesite y también implemento un middleware (funciones que se ejecutan durante el procesamiento de una solicitud HTTP) llamado CORS, que es un mecanismo que permite o restringe las solicitudes de recursos web desde un origen (dominio, protocolo y puerto) a otro. Esto ayuda a prevenir ataques de seguridad, al restringir qué dominios tienen permiso para realizar solicitudes a un servidor

Para iniciar el servidor backend, simplemente abro la terminal, pongo la ruta del servidor y escribo 'node index.js' y ya se inicia. Luego para acceder pongo 'localhost:3001/usuarios' si quiero acceder a los usuarios, pero si quisiera acceder a clases e instructores, sustituiría usuarios por clases o instructores

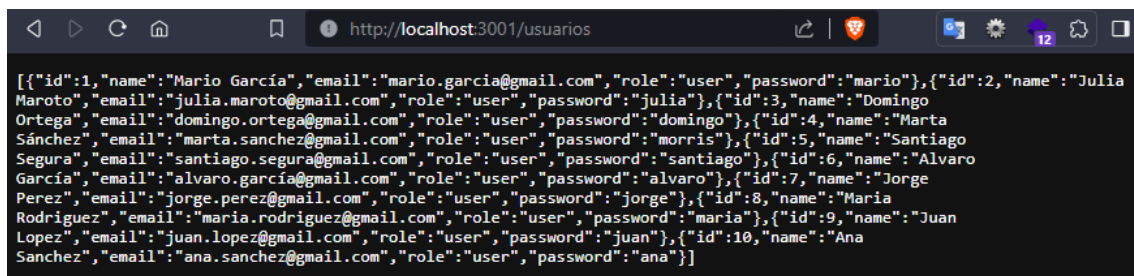


Imagen 6: API usuarios

Para el **servidor backend** creé una base de datos de clientes, instructores y clases en un servidor **Express** con **NodeJS** en formato **JSON** para poder implementar el login de usuarios y la creación de las webs de clases e instructores por medio de una **API** usando la función **fetch()**, que devuelve una **promise** a la respuesta de la solicitud, que permite trabajar con las respuestas de manera **asíncrona** y manejarlas eficientemente.

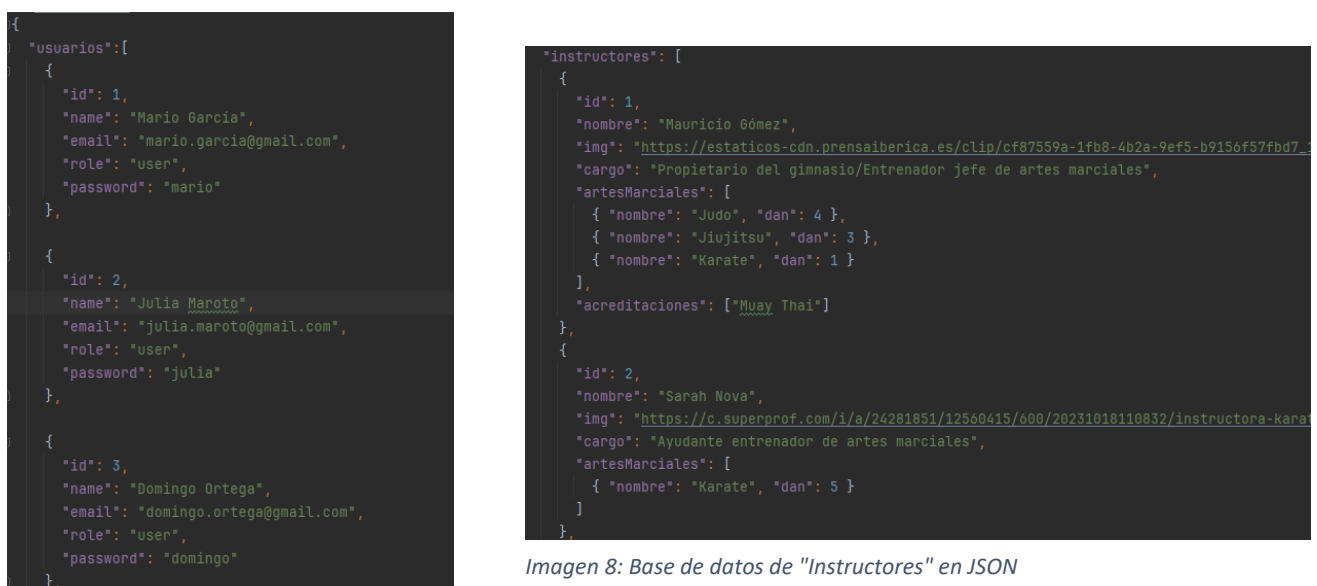


Imagen 8: Base de datos de "Instructores" en JSON

Imagen 7: Base de datos "Usuarios" en JSON



```

"clases": [
  {
    "id": 1,
    "nombre": "Clase de Karate Inicial",
    "descripcion": "Clase introductoria para principiantes",
    "img": "https://xn--qymnuez-8za.com/wp-content/uploads/2015/12/karate5-1024x683.jpg",
    "aforo": 20,
    "horario": "Lunes y Miércoles 18:00 - 19:30",
    "instructor": "Guy Victory",
    "users": [1, 4, 5]
  },
  {
    "id": 2,
    "nombre": "Clase de Karate Intermedio",
    "descripcion": "Para aquellos con experiencia intermedia",
    "img": "https://www.rivasciudad.es/wp-content/uploads/2013/01/ClubJiuJitsu2013_0116010.jpg",
    "aforo": 15,
    "horario": "Martes y Jueves 20:00 - 21:30",
    "instructor": "Morris Davis",
    "users": [6, 7, 8, 9, 10]
  }
]

```

*Imagen 9: Base de datos de "Clases" en JSON*

Entre las tecnologías clave para la parte del cliente se incluyen:

- **HTML5 y CSS3:** Para la estructura y el estilo del sitio web, asegurando una presentación visualmente atractiva y compatible con los estándares actuales.
- **Bootstrap:** Se utiliza para garantizar un diseño responsive, facilitando la visualización en dispositivos de diferentes tamaños y resoluciones.
- **JavaScript y jQuery:** Para la implementación de características interactivas, como formularios de registro e inicio de sesión, y para realizar solicitudes *asíncronas* al servidor backend.
- **Backend personalizado:** Se ha desarrollado un servidor backend para gestionar la lógica de negocio y la recuperación de datos externos, utilizando tecnologías como Node.js y Express.

He decidido usar **Node.js**, ya que permite el desarrollo eficiente de la lógica del servidor mediante JavaScript, lo que simplifica la coherencia del código a lo largo de la aplicación.

Con su arquitectura basada en eventos y su naturaleza **asíncrona**, Node.js es especialmente adecuado para manejar **múltiples solicitudes** de manera eficiente, lo que garantiza una respuesta rápida y una experiencia de usuario ágil en la plataforma web de DuBo Martial Arts.

Además, la gran cantidad de módulos y paquetes disponibles a través de npm facilita la incorporación de funcionalidades adicionales y la integración con servicios externos, contribuyendo así a la escalabilidad y versatilidad del sistema.

Como servidor he decidido utilizar **JSON** en vez de **MongoDB** porque, en el contexto de una aplicación más pequeña como la plataforma web de DuBo Martial Arts.

JSON ofrece una solución **ligera y sencilla** para almacenar datos.

MongoDB, siendo una base de datos NoSQL, puede ser una elección excelente para aplicaciones más **grandes** y **complejas** que requieran escalabilidad horizontal y un manejo eficiente de grandes conjuntos de datos.

Sin embargo, para un proyecto más modesto, optar por JSON como almacenamiento puede *simplificar* el desarrollo y *reducir* la complejidad del sistema.

## **3. Descripción detallada en la parte Frontend**

### **3.1 Páginas Principales**

#### **1. Página de Inicio (index.html)**

La página de inicio ha sido diseñada para captar la atención del visitante desde el primer momento. Incluye:

- **Barra de Navegación Intuitiva:** Una barra de navegación fácil de usar que proporciona acceso rápido a las secciones principales del sitio.
- **Mensaje de Bienvenida Impactante:** Un mensaje de bienvenida que resalta la esencia única del dojo y motiva a los visitantes a explorar más.
- **CTA (Llamada a la acción):** Contiene una llamada a inscribirse con una oferta de una clase gratis y una limitación temporal de 10 clientes para conseguir que las personas interesadas se decidan rápido.

#### **2. Página de Clases (clases.html)**

La página de clases ofrece una visión detallada de las diferentes clases disponibles. Cada clase se presenta de la siguiente manera:

- **Información Específica:** Detalles como nombre, descripción, horario, instructor y aforo de cada clase.
- **Opción de inscripción**

#### **3. Página de Instructores (instructores.html)**

La página de instructores proporciona perfiles detallados de cada instructor, lo que incluye:

- **Información Personal y Profesional:** Detalles como nombre, especialidad, experiencia y una breve biografía.
- **Imágenes y Videos:** Se incorporan elementos multimedia para dar a los visitantes una visión más completa de cada instructor.

#### **4. Página de Acceso Alumnos (login.html)**

Esta sección está diseñada específicamente para los alumnos que están registrados o quieren registrarse y ofrece:

- **Formularios de Registro e Inicio de Sesión:** Interfaz intuitiva para facilitar el acceso a los recursos exclusivos.
- **Opciones de Autenticación:** Integración con redes sociales para facilitar el proceso de inicio de sesión.

## **3.2 Estilos y Diseño**

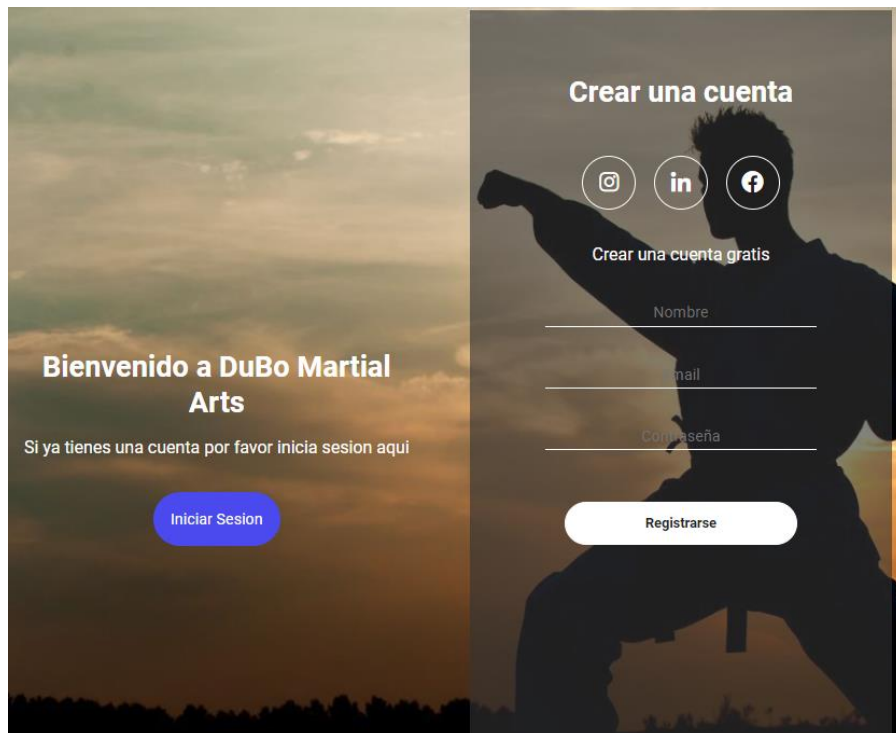
El diseño del sitio se ha abordado con atención al detalle y coherencia en toda la plataforma. El uso de Bootstrap garantiza un diseño responsive, adaptándose de manera efectiva a diferentes dispositivos, desde dispositivos de escritorio hasta móviles.

## **4. Resultados y Funcionalidades**

### ***4.1 Interactividad***

El sitio web ofrece varias funciones interactivas destinadas a mejorar la experiencia del usuario:

- **Formularios Dinámicos:** Los formularios de registro e inicio de sesión utilizan JavaScript para validar datos en tiempo real y proporcionar retroalimentación instantánea.



*Imagen 10: Formulario de registro*

- **Carga Dinámica de Contenido:** Se ha implementado una lógica en el frontend para cargar dinámicamente información sobre clases e instructores a través de APIs, mejorando la eficiencia del sitio.

## 5. Desafíos Enfrentados

Durante el desarrollo, encontré desafíos significativos:

- **Coordinación de Datos entre Frontend y Backend:** La sincronización eficiente de datos entre el frontend y el backend la abordé mediante la implementación de API RESTful para gestionar solicitudes y respuestas.
- **Login y registro de usuarios:** en la creación de la página de login.html y su correspondiente JavaScript conseguí validar los usuarios de la base de datos de JSON mediante un Local Storage con los usuarios ya registrados en la base de datos de db.json pero no conseguí registrar usuarios.

Al principio no conseguí encriptar los datos del login con un token pero acabé consiguiéndolo. A continuación explico:

- Importé la siguiente librería que implementa varios algoritmos criptográficos, como AES (Advanced Encryption Standard) para cifrado y SHA256 (Secure Hash Algorithm 256 bits) para funciones hash. En concreto esta línea está utilizando la desestructuración de objetos de JavaScript para extraer las funciones AES y SHA256 del objeto window.CryptoJS.

```
const { AES, SHA256 } = window.CryptoJS;
```

- Creo las funciones para encriptar y desencriptar la información usando un token.

```
// Función para encriptar la información usando un token
1 usage
function encryptData(data, token) {
    const encryptedData = AES.encrypt(data, token).toString();
    return encryptedData;
}

// Función para desencriptar la información usando un token
1 usage
function decryptData(encryptedData, token) {
    const decryptedData = AES.decrypt(encryptedData, token).toString(
        window.CryptoJS.enc.Utf8);
    return decryptedData;
}
```

- Si el inicio de sesión es exitoso, se encriptan los datos del usuario, en este caso correo electrónico y contraseña utilizando la función **'encryptdata'**

y se almacenan en el almacenamiento local del navegador con la clave **'loggedInUser'** y se redirige al usuario a la página de inicio.

```
if (user) {
    alert('¡Inicio de sesión exitoso!');

    // Encriptar la sesión antes de almacenarla en el
    almacenamiento local
    const token :string = 'tuTokenSecreta';
    const encryptedSession = encryptData(JSON.stringify( { value: {
        email, password } }), token);
    localStorage.setItem('loggedInUser', encryptedSession);

    window.location.href = '/';
} else {
    alert('Credenciales incorrectas. Por favor, inténtalo de
    nuevo.');
```

- Por último, se ejecuta al cargar la página y busca datos de sesión previamente almacenados (**'loggedInUser'**). Si encuentra una sesión, la desencripta utilizando `decryptData` y redirige al usuario a la página principal. Este enfoque ofrece inicio de sesión persistente para mantener a los usuarios autenticados incluso después de recargar la página.

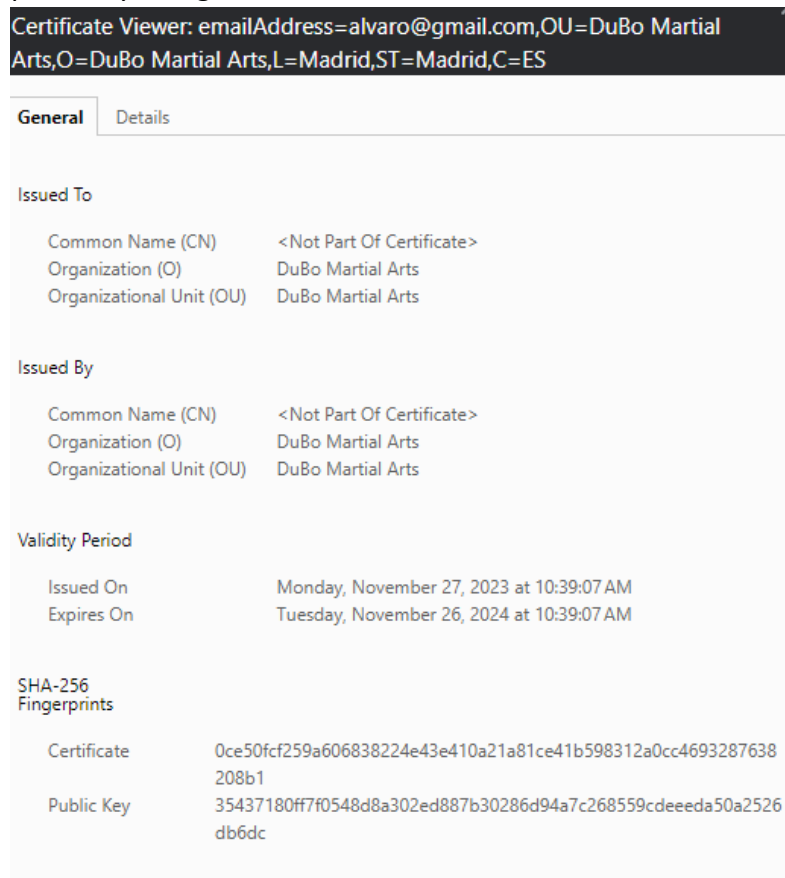
```
// Comprobar si hay una sesión activa al cargar la página
document.addEventListener( type: 'DOMContentLoaded', listener: () :void => {
    const loggedInUser :string = localStorage.getItem( key: 'loggedInUser');

    if (loggedInUser) {
        // Desencriptar la sesión al cargar la página
        const token :string = 'tuTokenSecreta';
        const decryptedSession = decryptData(loggedInUser, token);

        window.location.href = '/';
    }
});
```

Storage	Key	Value
Local storage	loggedInUser	U2FsdGVkX1/GnRQD8mf43LUANlc+iIMYhzl...
https://localhost:3000		
Session storage		

- Implementé también un **certificado SSL** en la parte del servidor y la del cliente para mayor seguridad



## **P7: Crear y Utilizar un Plan de Pruebas para Revisar la Rendimiento y Diseño de un Sitio Web Multipágina**

Para garantizar el rendimiento y diseño óptimos de un sitio web multipágina, es crucial implementar un Plan de Pruebas integral. Este plan se enfocará en aspectos clave como la velocidad de carga, la usabilidad, la compatibilidad con múltiples dispositivos y la coherencia del diseño. Aquí se presenta un plan de pruebas detallado para llevar a cabo esta revisión:

### **Objetivos de las Pruebas:**

#### **1. Rendimiento:**

- Evaluar y optimizar los tiempos de carga de cada página.
- Verificar la velocidad de respuesta del servidor.

#### **2. Usabilidad:**

- Analizar la navegación en todas las páginas.
- Verificar la claridad y efectividad del diseño de la interfaz de usuario.



### **3. Compatibilidad:**

- Probar la visualización y funcionalidad en diferentes navegadores (Chrome, Firefox, Safari, Edge).
- Asegurar la adaptabilidad a diferentes tamaños de pantalla (móviles, tabletas, computadoras).

### **4. Diseño:**

- Confirmar la coherencia del diseño en todas las páginas.
- Verificar la legibilidad del contenido y la visibilidad de los elementos clave.

## **Estrategia de Pruebas:**

### **1. Pruebas de Rendimiento:**

- Utilizar herramientas como Lighthouse o PageSpeed Insights para evaluar el rendimiento.
- Medir y comparar los tiempos de carga de las páginas principales.
- Simular conexiones de red lentas para identificar posibles cuellos de botella.

### **2. Pruebas de Usabilidad:**

- Realizar pruebas de usuario reales para evaluar la facilidad de uso.
- Verificar la accesibilidad del sitio para usuarios con discapacidades.
- Evaluar la coherencia de la navegación y la lógica del flujo de usuario.

### **3. Pruebas de Compatibilidad:**

- Validar la apariencia y funcionalidad en diferentes navegadores y versiones.
- Utilizar herramientas de emulación para probar en dispositivos con diversos tamaños de pantalla.
- Asegurar que las características clave funcionen correctamente en todos los dispositivos.

### **4. Pruebas de Diseño:**

- Revisar la consistencia del diseño en todas las páginas.
- Verificar la alineación y legibilidad de textos e imágenes.
- Evaluar la estética general y la adherencia a los estándares de diseño establecidos.

## **M1 Analizar el impacto de las tecnologías y frameworks de desarrollo web más comunes en relación con el diseño, la funcionalidad y la gestión de sitios web.**

En la era digital actual, las tecnologías y frameworks de desarrollo web desempeñan un papel fundamental en la creación y gestión de sitios web efectivos. Estas herramientas no solo **influyen** en el aspecto visual de un sitio, sino que también tienen un profundo impacto en su *funcionalidad, rendimiento y experiencia* del usuario.

En esta sección, explicaré cómo las tecnologías de desarrollo web más comunes han transformado la forma en que diseñamos, desarrollamos y gestionamos sitios web, y cómo estas decisiones técnicas pueden **repercutir** en la satisfacción de los usuarios y el éxito del sitio web.

A continuación hago un análisis más detallado de cómo estas tecnologías influyen en diferentes aspectos de un sitio web:

- **Diseño Web:** Las tecnologías y frameworks modernos permiten un diseño web más *atractivo y adaptable*.

El uso de **CSS3** y **HTML5** facilita la creación de diseños *responsive* que se ajustan a diferentes tamaños de pantalla, lo que mejora la experiencia del usuario en dispositivos móviles y de escritorio.

Los frameworks como **Bootstrap** proporcionan una estructura y componentes predefinidos para el diseño, lo que *acelera* el proceso de diseño web.

- **Funcionalidad:** Los frameworks de desarrollo web, como *React* y *Angular*, ofrecen una funcionalidad avanzada en el lado del **cliente**.

Estos *frameworks* permiten la **creación** de aplicaciones web interactivas y dinámicas sin necesidad de recargar la página completa.

Esto mejora la **usabilidad** al proporcionar a los usuarios una experiencia similar a la de una aplicación de escritorio.

- **Gestión de Contenido:** Los sistemas de gestión de contenido (CMS) como *WordPress* han revolucionado la forma en que se gestiona el contenido en línea.

Permiten a los usuarios sin conocimientos técnicos **actualizar** y **administrar** fácilmente el contenido de sus sitios web.

Esto **ahorra** tiempo y recursos, ya que no se requiere programación manual para hacer cambios en el contenido.

- **Optimización para Motores de Búsqueda (SEO):** Las tecnologías de desarrollo web también tienen un impacto en el *SEO*.

Los sitios web bien **estructurados** y *compatibles* con los estándares web tienden a clasificarse mejor en los motores de búsqueda.

El uso adecuado de etiquetas *HTML*, **metadescripciones**, URLs amigables para SEO y la **velocidad de carga** del sitio son factores clave que las tecnologías web pueden influir.

- **Gestión y Mantenimiento:** Los frameworks y tecnologías modernas facilitan la *gestión* y el *mantenimiento* de sitios web.

Los sistemas de control de versiones como **Git** permiten a los equipos de desarrollo *colaborar* y *rastrear* cambios en el código.

Las actualizaciones y parches de seguridad son más fáciles de implementar, lo que garantiza que el sitio web se mantenga **seguro** y **actualizado**.

## **M2 Revisar la influencia de los motores de búsqueda en el rendimiento de los sitios web**

Los motores de búsqueda juegan un papel fundamental en la visibilidad y el éxito de los sitios web. La forma en que un sitio web se clasifica en los resultados de búsqueda tiene un impacto directo en la cantidad y calidad del tráfico que recibe. Este criterio se centra en comprender la influencia de los motores de búsqueda y cómo la Optimización de Motores de Búsqueda (SEO) puede marcar la diferencia.

A continuación, explicaré este concepto:

1. ***Influencia de los motores de búsqueda:*** Los motores de búsqueda operan mediante complejos algoritmos que determinan la relevancia y la calidad de un sitio web en relación con las consultas de los usuarios. La posición en los resultados de búsqueda, conocida como *SERP*, desempeña un papel crucial en la visibilidad en línea.
2. ***SEO como estrategia esencial:*** es una estrategia proactiva que busca *mejorar la visibilidad* de un sitio web en los resultados de búsqueda **orgánicos**. Esto implica la optimización de diversos aspectos, como el contenido, la estructura del sitio, la velocidad de carga, la calidad de los enlaces...

3. **Pruebas y análisis fundamentales:** son esenciales para mejorar el *rendimiento* de un sitio web en los motores de búsqueda.

Esto incluye:

- **Auditorías** para identificar problemas técnicos y de contenido.
- **Investigación de palabras clave** para optimizar el contenido
- **Seguimiento del tráfico orgánico** para entender cómo los usuarios llegan al sitio
- **Análisis de competidores** para aprender de estrategias exitosas
- **Seguimiento de métricas clave** como la clasificación y la tasa de clics para evaluar el impacto de las estrategias y realizar ajustes.

4. **Mejora constante del valor de índice y rango:** Basándonos en las pruebas y el análisis, es necesario hacer *estrategias específicas* y cambios que tengan un impacto positivo en el **valor de índice** del sitio web.

Esto puede incluir:

- Sugerencias de **optimización** de contenido
- Mejoras técnicas
- Estrategias de construcción de enlaces
- Otras tácticas de SEO destinadas a mejorar la **posición** en los resultados de búsqueda.

5. **Seguimiento y ajustes continuos:** la mejora de la clasificación y el rendimiento en los motores de búsqueda es un proceso continuo.

### **M3 Analizar una serie de herramientas y técnicas disponibles para diseñar y desarrollar un sitio web a medida.**

El proceso de diseñar y desarrollar un sitio web a medida implica la selección cuidadosa de herramientas y técnicas para garantizar un proyecto exitoso. Aquí se presenta un análisis detallado de diversas herramientas y técnicas disponibles para este propósito:

#### **1. Lenguajes de Programación:**

- **JavaScript:** Esencial para el desarrollo del lado del cliente. Facilita la creación de interactividad y dinamismo en las páginas web.
- **Node.js** (JavaScript en el lado del servidor): Permite la construcción de aplicaciones web escalables y eficientes en el servidor. Facilita la gestión de dependencias y la implementación de lógica del lado del servidor.
- **HTML5 y CSS3:** Fundamentales para la estructura y estilo del contenido web. HTML5 ofrece nuevas etiquetas para una estructura más semántica, mientras que CSS3 proporciona estilos avanzados y animaciones.



#### **2. Entorno de Desarrollo:**

- **Visual Studio Code:** Un editor ligero y potente que soporta numerosas extensiones para facilitar el desarrollo en JavaScript, Node.js, HTML y CSS.
- **Node Package Manager (npm):** Permite gestionar las dependencias del proyecto y descargar bibliotecas de terceros de manera sencilla.



### **3. Frameworks y Bibliotecas:**

- **Express.js:** Un framework de Node.js que simplifica la creación de aplicaciones web y la gestión de rutas.
- **React.js:** Una biblioteca de JavaScript para construir interfaces de usuario interactivas. Ideal para la creación de componentes reutilizables.
- **Bootstrap:** Un marco de diseño CSS que agiliza el desarrollo web responsivo y facilita la creación de interfaces atractivas.

### **4. Bases de Datos:**

- **MongoDB:** Una base de datos NoSQL que se integra bien con Node.js. Ideal para proyectos donde la flexibilidad en la estructura de datos es crucial.
- **Mongoose (ODM para MongoDB):** Facilita la interacción con bases de datos MongoDB mediante JavaScript, proporcionando una capa de abstracción.

### **5. Control de Versiones:**

- **Git:** Esencial para el control de versiones del código fuente. Permite el seguimiento de cambios, colaboración efectiva y la gestión de ramas para el desarrollo concurrente.

### **6. Pruebas y Automatización:**

- **Jest:** Un marco de prueba para JavaScript que garantiza la calidad del código.
- **Selenium:** Para pruebas de extremo a extremo, asegurando que la aplicación funcione correctamente en diferentes navegadores.
- **Webpack:** Para el empaquetado de módulos y la optimización del código.

### **7. Diseño y Prototipado:**

- **Figma o Sketch:** Herramientas de diseño para la creación de prototipos de interfaz de usuario.

### **8. Despliegue y Hosting:**

- **Heroku:** Una plataforma en la nube que permite el despliegue sencillo de aplicaciones web. Soporta múltiples lenguajes y ofrece una capa gratuita con opciones de escalabilidad.
- **Netlify:** Especializado en la implementación de sitios web estáticos y aplicaciones de una sola página (SPA). Proporciona funciones como implementaciones automáticas desde repositorios de Git y URL personalizadas.
- **Firebase:** Parte del conjunto de herramientas de desarrollo de Firebase de Google. Ofrece un hosting rápido y seguro con entrega de contenido a través de CDN, integración sencilla con Firebase Authentication y funciones para la ejecución de código en el lado del servidor. Con el plan Spark no se puede desplegar un proyecto Node.js



## **9. Seguridad:**

- **Helmet:** Un conjunto de medidas de seguridad para aplicaciones web Express.
- **bcrypt.js:** Para el cifrado seguro de contraseñas.

## **10. Monitoreo y Analítica:**

- **Google Analytics:** Para obtener información sobre el tráfico y el comportamiento del usuario.
- **New Relic o Datadog:** Herramientas de monitoreo para identificar y solucionar problemas de rendimiento.

### **M4 Justificar las decisiones de implementación del sitio web multipágina con respecto al documento de diseño.**

El diseño e implementación del sitio web multipágina para DuBo Martial Arts se ha llevado a cabo considerando diversas decisiones que buscan optimizar la experiencia del usuario, la mantenibilidad del código y la seguridad. A continuación, se justifican algunas de las decisiones clave tomadas en el proceso de desarrollo en relación con el documento de diseño inicial:

#### **1. Multipágina vs. Página Única:**

- Se optó por un enfoque multipágina debido a la naturaleza clara y diferenciada de las secciones del sitio, como "Inicio", "Clases", e "Instructores".
- Cada sección tiene un propósito específico y proporciona información única. Este enfoque facilita la navegación y comprensión de los usuarios sobre la estructura del sitio.

#### **2. Barra de Navegación:**

- La barra de navegación se ha implementado de manera que sea clara y fácil de usar. La decisión de utilizar Bootstrap facilita un diseño responsive y atractivo.
- Se incluyeron enlaces directos a las secciones principales, mejorando la accesibilidad y la experiencia de usuario.

#### **3. Uso de Bootstrap:**

- La elección de utilizar Bootstrap como framework de diseño proporciona un conjunto de herramientas útiles y estilos predefinidos que aceleran el desarrollo y garantizan una apariencia consistente en diferentes dispositivos y navegadores.

#### **4. Uso de HTTPS y Certificado SSL:**

- Se implementó HTTPS mediante un certificado SSL para garantizar la seguridad de la comunicación entre el usuario y el servidor. Esto es esencial,

especialmente al manejar información sensible, como datos de inicio de sesión de los usuarios.

### **5. *Página de Inicio Atractiva:***

- La sección de inicio se diseñó con un enfoque visual atractivo para captar la atención de los visitantes. Se utiliza una combinación de imágenes llamativas y texto claro para transmitir la esencia del dojo y alentar a los usuarios a explorar más.

### **6. *Secciones Específicas para Clases e Instructores:***

- Cada sección, ya sea "Clases" o "Instructores", tiene su propia página dedicada. Esto facilita la organización de la información y mejora la capacidad de los motores de búsqueda para indexar contenido específico, lo que beneficia al SEO del sitio.

### **7. *Formulario de Contacto y Enlaces a Redes Sociales:***

- La inclusión de un formulario de contacto brinda a los usuarios una manera fácil de comunicarse con el dojo.
- Los enlaces a redes sociales permiten a los usuarios conectarse y seguir las actualizaciones del dojo en otras plataformas, fortaleciendo la presencia en línea.

### **8. *Seguridad en Autenticación de Usuarios:***

- En la sección de "Acceso Alumnos", se implementa una funcionalidad básica de inicio de sesión. La información de inicio de sesión se maneja con encriptación utilizando CryptoJS para garantizar la seguridad de las credenciales del usuario.

### **9. *Carga de Datos desde una API:***

- Para la sección de "Clases" y "Instructores", se optó por cargar dinámicamente la información desde una API en lugar de codificarla directamente en las páginas HTML. Esto mejora la mantenibilidad y facilita la actualización de datos sin necesidad de modificar el código fuente.

## **M5 Analizar la calidad aplicada durante el diseño de sitios web multipágina y fases de desarrollo.**

El proceso de diseño y desarrollo del sitio web multipágina para DuBo Martial Arts se ha llevado a cabo con un enfoque centrado en la calidad en diversas fases del proyecto. A continuación, se analiza la calidad aplicada en cada etapa del diseño y desarrollo:

### **1. *Fase de Planificación y Requisitos:***

- **Calidad de Requisitos:** Se realizaron entrevistas y reuniones con DuBo Martial Arts para comprender a fondo sus necesidades y expectativas. La calidad de los

requisitos se aseguró mediante la identificación clara de características clave, como secciones específicas para "Clases" e "Instructores", y la integración de un formulario de contacto.

## ***2. Diseño de Interfaz de Usuario (UI):***

- **Usabilidad y Accesibilidad:** Se aplicaron principios de usabilidad para garantizar que la interfaz fuera intuitiva y fácil de navegar. Se consideró la accesibilidad, asegurándose de que la información crítica fuera accesible para usuarios con discapacidades mediante el uso de prácticas de diseño inclusivas.
- **Diseño Responsive:** Se implementó un diseño responsivo utilizando Bootstrap, asegurando una experiencia consistente en una variedad de dispositivos, desde computadoras de escritorio hasta dispositivos móviles.

## ***3. Desarrollo Frontend:***

- **Estándares de Codificación:** Se siguieron estándares de codificación limpios y consistentes para mejorar la mantenibilidad del código. El uso de HTML semántico y CSS modular contribuyó a la claridad y legibilidad del código.
- **Compatibilidad entre Navegadores:** Se realizaron pruebas exhaustivas en diferentes navegadores para garantizar la compatibilidad y la coherencia visual en todos los entornos.

## ***4. Integración de Bootstrap y Tecnologías Externas:***

- **Actualizaciones y Parches:** Se aplicaron las últimas versiones de Bootstrap y otras tecnologías externas para aprovechar las características más recientes y abordar posibles problemas de seguridad. Se estableció un plan para aplicar actualizaciones y parches de manera regular.

## ***5. Backend y Seguridad:***

- **Manejo de Datos Sensibles:** Se implementaron prácticas de seguridad, como el uso de HTTPS y certificados SSL, para garantizar la protección de datos sensibles, como la información de inicio de sesión de los usuarios.
- **Autenticación Segura:** En la sección de "Acceso Alumnos", se aplicaron prácticas seguras de autenticación utilizando técnicas de encriptación para proteger las credenciales de los usuarios.

## **D1: Justificación de Tecnologías, Servicios de Gestión, Herramientas y Programas Informáticos para el Sitio Web DuBo Martial Arts**

El desarrollo del sitio web para DuBo Martial Arts se basa en la elección de varias tecnologías, servicios de gestión, herramientas y programas informáticos que se seleccionaron cuidadosamente para garantizar un rendimiento óptimo y una experiencia positiva tanto para los usuarios como para los administradores del sitio. A continuación, se presenta una justificación detallada de estas elecciones:

### **Tecnologías Web:**

#### **1. HTML, CSS, Bootstrap:**

- Lenguajes base para la estructura y el estilo del sitio. Bootstrap, un framework de diseño de código abierto, se utiliza para garantizar un diseño receptivo y móvil, así como para aprovechar componentes predefinidos que mejoran la consistencia y la estética del sitio.

#### **2. JavaScript (ES6):**

- Lo utilizo para agregar interactividad y dinamismo al sitio. La versión ES6 se elige por sus características modernas, facilitando el mantenimiento del código y mejorando la legibilidad.

#### **3. Node.js y Express:**

- Creación del servidor web. Estas tecnologías basadas en JavaScript permiten la construcción de aplicaciones web rápidas y eficientes, facilitando la creación de rutas y el manejo de solicitudes HTTP.

### **Servicios de Gestión de Datos:**

#### **1. JSON para Datos Estáticos:**

- La información estática del sitio, como instructores y clases, se almacena en archivos JSON. Esta elección simplifica la gestión de datos y facilita su integración en el sitio web sin la necesidad de un sistema de gestión de bases de datos completo.

### **Herramientas y Programas Informáticos:**

#### **1. Visual Studio Code:**

- Se elige como el entorno de desarrollo principal debido a su interfaz, amplia gama de extensiones y soporte sólido para JavaScript y Node.js.

## **2. *CryptoJS:***

- Se incorpora para proporcionar funciones de cifrado y descifrado en el módulo de inicio de sesión, asegurando la seguridad de la información del usuario durante el proceso de autenticación.

## **Servidor y Certificado HTTPS:**

### **1. *Node.js con módulo HTTPS:***

- Utilizo Node.js con un servidor HTTPS para garantizar una conexión segura entre el servidor y los clientes. Esto es crucial, especialmente al manejar información confidencial como las credenciales de inicio de sesión de los usuarios.

### **2. *Certificados SSL:***

- Se implementan certificados SSL para cifrar la comunicación entre el servidor y los clientes, protegiendo la privacidad de los datos transmitidos.

## **Servicios Externos:**

### **1. *Bootstrap, jQuery, Font Awesome:***

- Utilizo estos servicios externos se utilizan para acceder a bibliotecas y recursos que mejoran la apariencia y la funcionalidad del sitio sin tener que alojar los archivos localmente.