

GA10-220501097-AA10-EV01 ELABORA DOCUMENTOS TÉCNICOS Y  
DE USUARIO DEL SOFTWARE

EDWIN DE JESÚS RESTREPO ATEHORTUA  
YIRA PAOLA URANGO TOVIO  
APRENDICES

FELIPE MARTINEZ LEIVA  
INSTRUCTOR

TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE  
SERVICIO NACIONAL DE APRENDIZAJE  
SENA

## **Introducción**

El desarrollo de sistemas de software exige la creación de documentación que garantice la correcta instalación, operación y mantenimiento de aplicaciones y páginas web, dentro de este proceso, el manual técnico cumple un papel esencial, pues proporciona a los usuarios y desarrolladores una referencia clara y detallada sobre cómo se ha construido y cómo funciona el sistema. Este manual es especialmente útil para aquellos que deseen instalar, modificar o ampliar las funcionalidades del software, ya que describe en detalle los aspectos técnicos clave del mismo.

En este documento, se presentará el manual técnico del sistema desarrollado a lo largo del programa de formación, este manual contiene una descripción completa de los elementos técnicos, estándares y tecnologías empleadas en el desarrollo del sistema, así como las instrucciones para su instalación y configuración.

---

## 1. Prerrequisitos de instalación del sistema

En este apartado, describe los requisitos de hardware, software y dependencias que se necesitan para instalar y ejecutar el sistema.

### Hardware:

- Procesador: Mínimo 2 núcleos, 2 GHz
- Memoria RAM: 4 GB o más
- Espacio en disco: 500 MB a 1GB de espacio libre
- Disco duro: Preferiblemente SSD para mayor velocidad.

### Software:

- Sistema operativo: Windows, Linux o MacOS
- Servidor web: Apache o Nginx
- PHP: Versión 8.2.12 o superior
- Node.js: Versión 20.9.0 o superior
- MySQL: Versión 5.7 o superior
- Yarn: Para la gestión de dependencias de ReactJS

### Dependencias:

- React: ^18.2.0
- Fetch (para las peticiones HTTP)
- PDO, PHP Data Objects (para la conexión a la base de datos)
- Browser: ^4.3.3
- React-paypal-js: ^8.5.0
- React-stripe-js: ^2.7.1
- Stripe-js: ^3.5.0
- Bootstrap: ^5.3.3
- React-bootstrap: ^2.10.3
- React-dom: ^18.2.0
- React-router-dom: ^6.23.1

---

## 2. Frameworks y estándares con los que se construyó el sistema

### Frontend:

- **ReactJS:** Librería de JavaScript para construir interfaces de usuario.
- **Bootstrap:** Utilizado para el diseño responsivo y el estilo general de la interfaz.

- **Vite:** Un bundler de aplicaciones que proporciona una experiencia de desarrollo rápida y optimizada. Utiliza técnicas de carga en caliente y es ideal para proyectos modernos con React.

#### Backend:

- **PHP puro:** Para manejar las peticiones desde el frontend, procesar lógica del negocio y gestionar la base de datos.
- **MySQL:** Sistema de gestión de base de datos relacional.
- **Apache/Nginx:** Servidor web para servir las aplicaciones PHP.

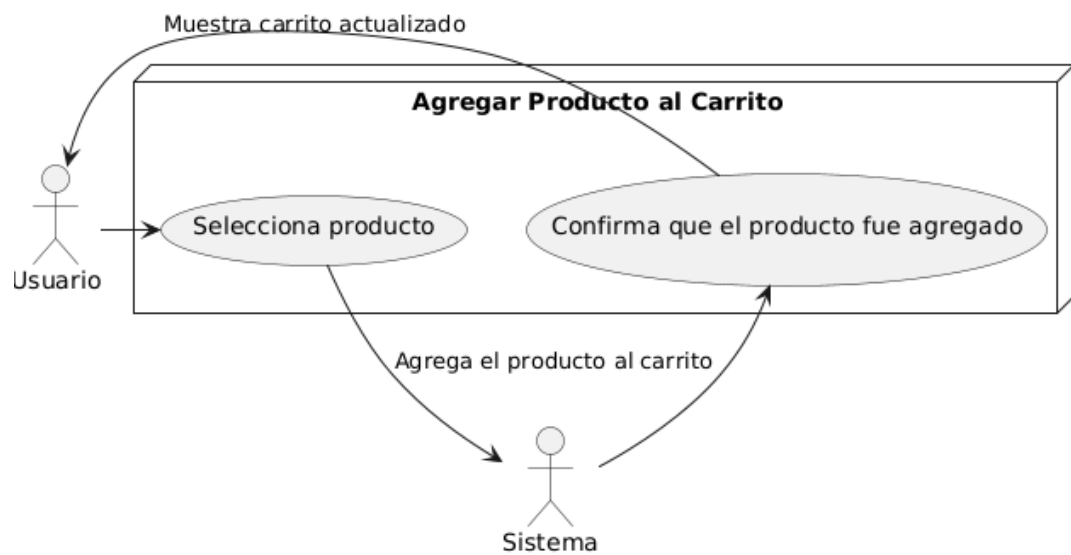
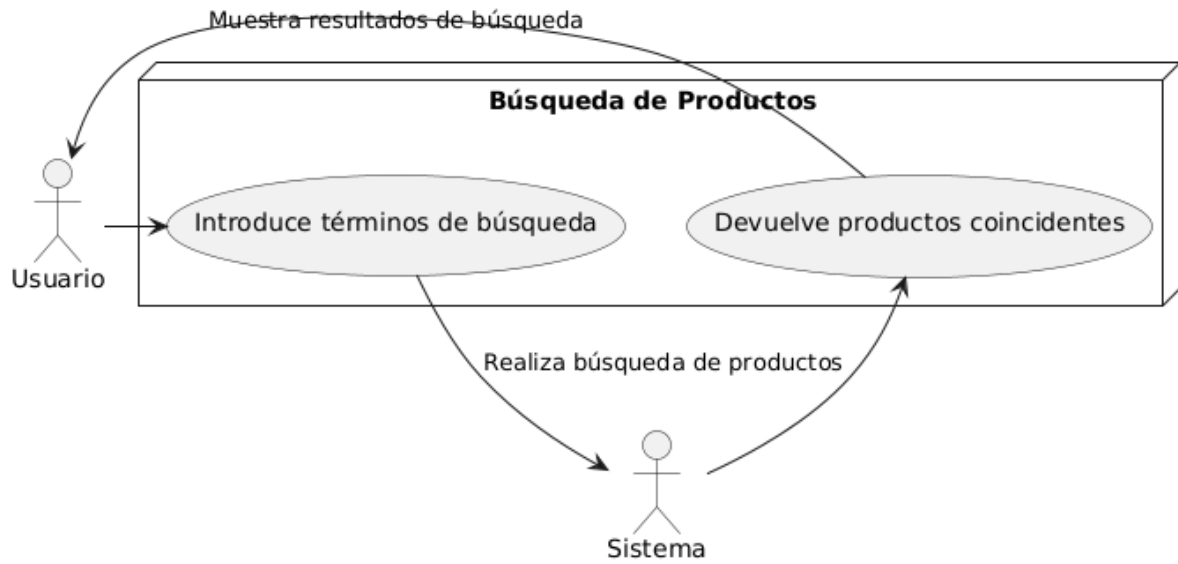
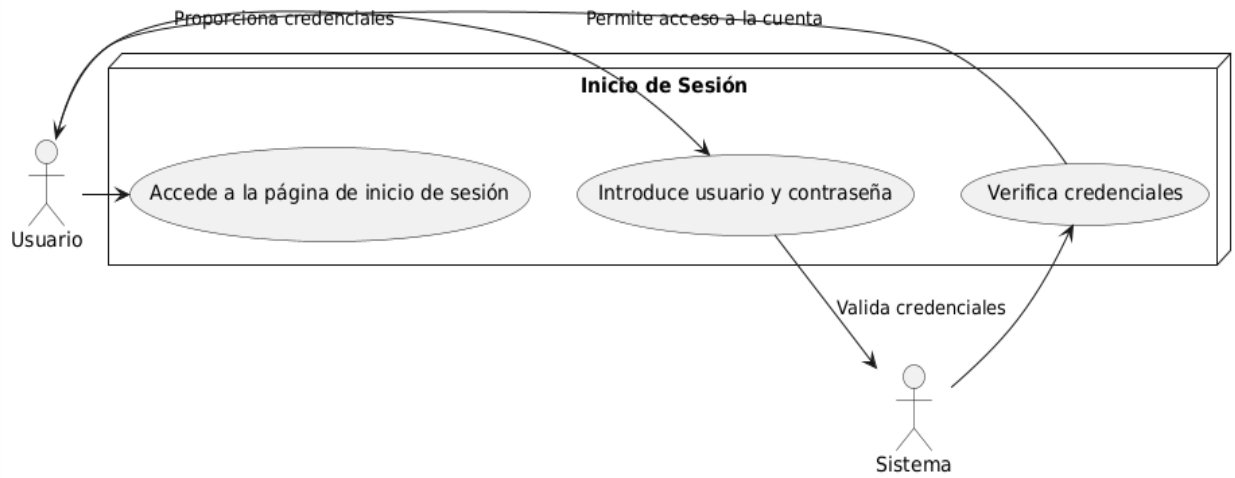
#### Estándares:

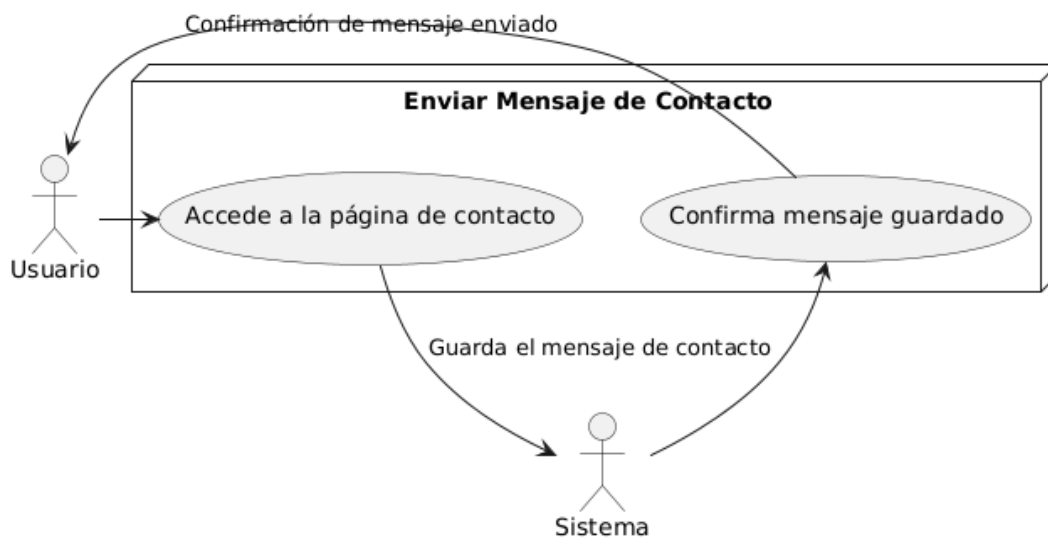
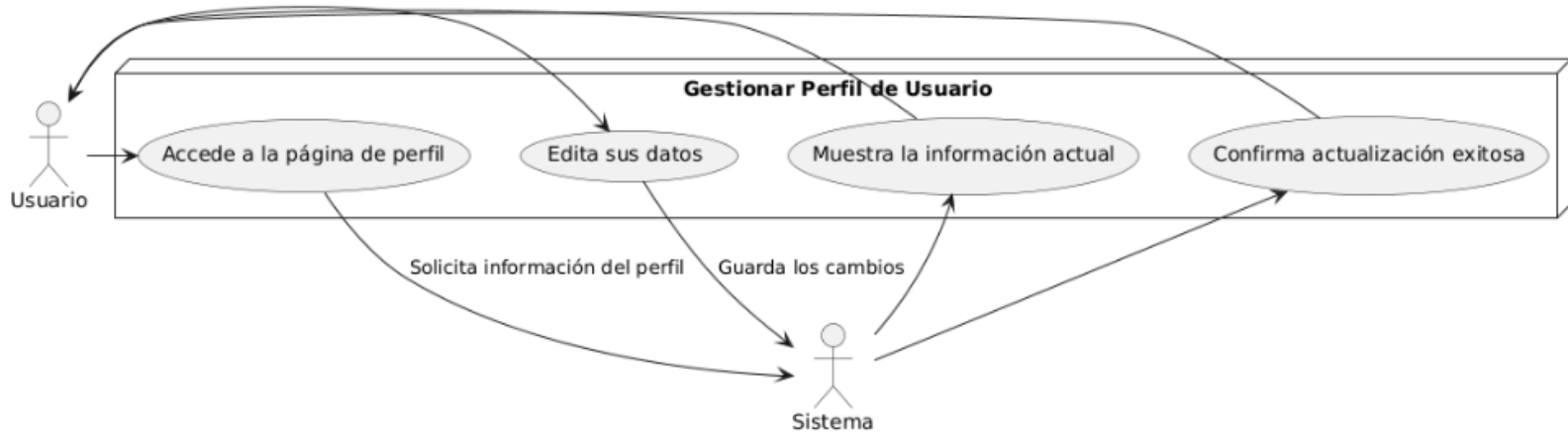
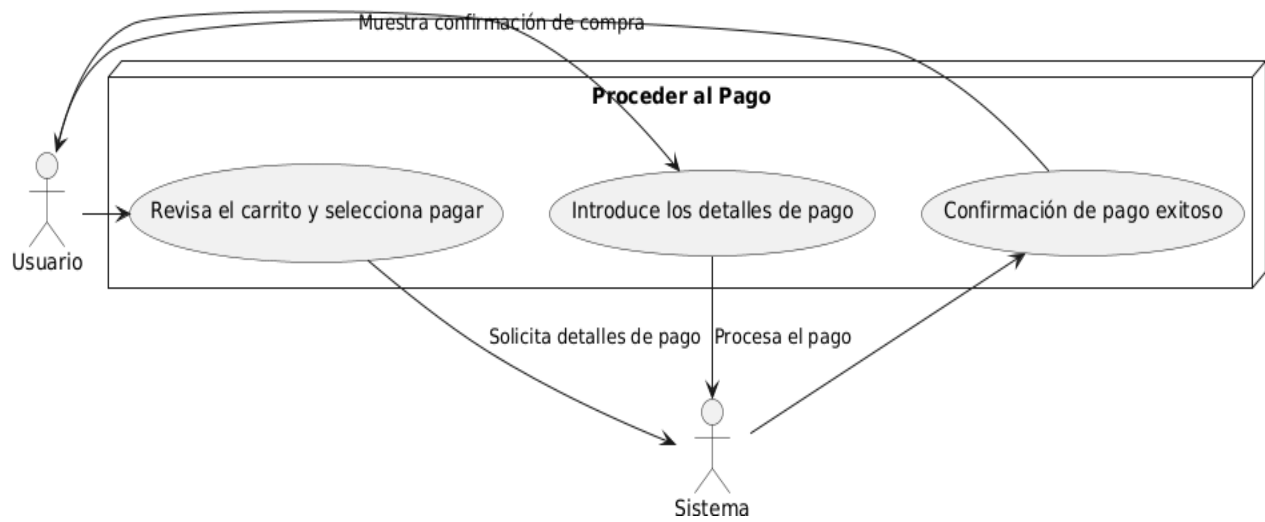
- **HTML5 y CSS3:** Para la estructura y estilos.
  - **RESTful API:** El sistema sigue el patrón de arquitectura REST para la comunicación entre el frontend y backend. También contamos con documentación clara y validación de datos, al igual que implementar medidas de seguridad como autenticación segura y encriptación.
  - **Git:** para control de versiones y para gestionar dependencias herramientas como npm/yarn.
- 

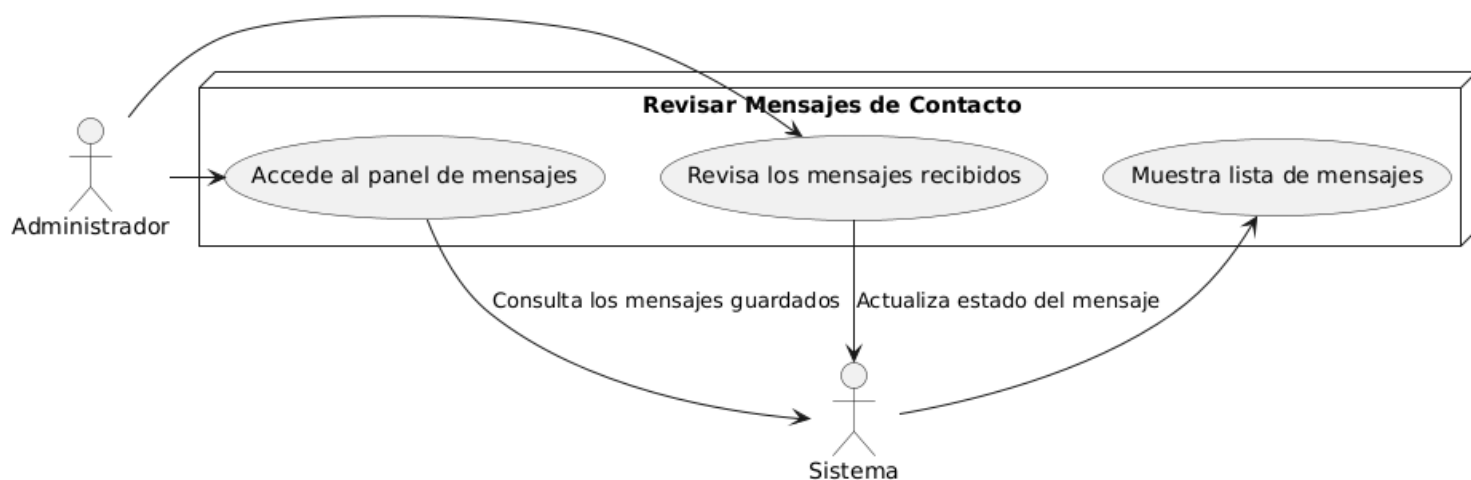
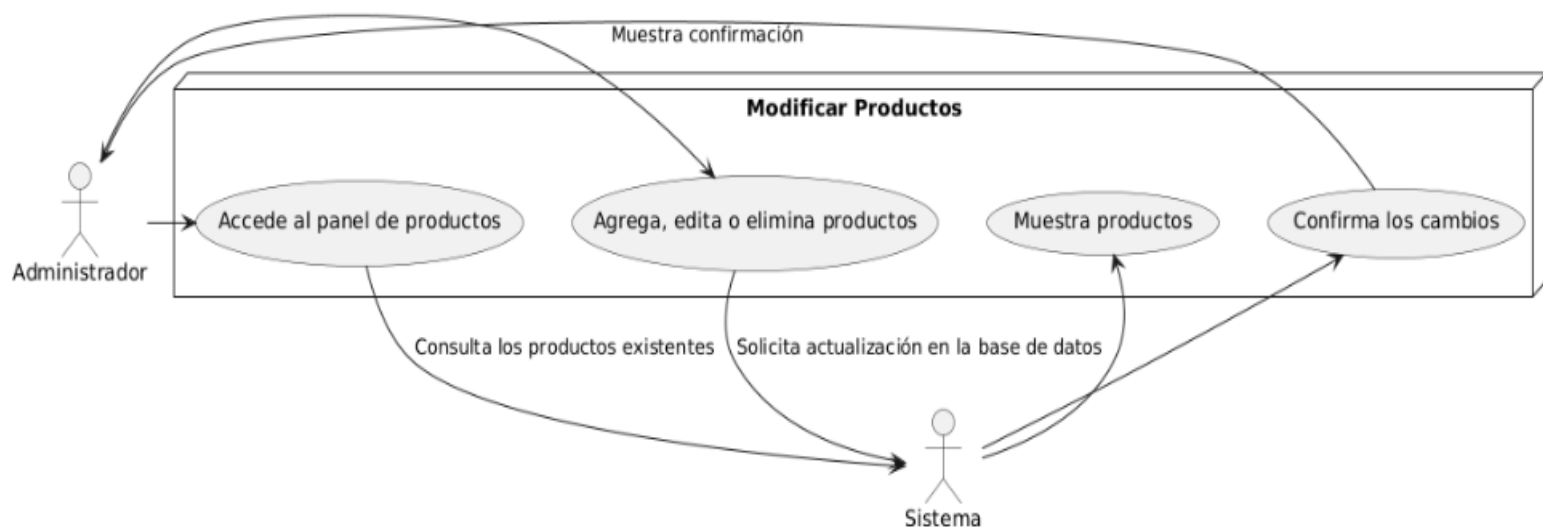
### 3. Diagrama de casos de uso del sistema

El **diagrama de casos de uso** muestra las interacciones del usuario con el sistema y cómo este responde a cada acción.





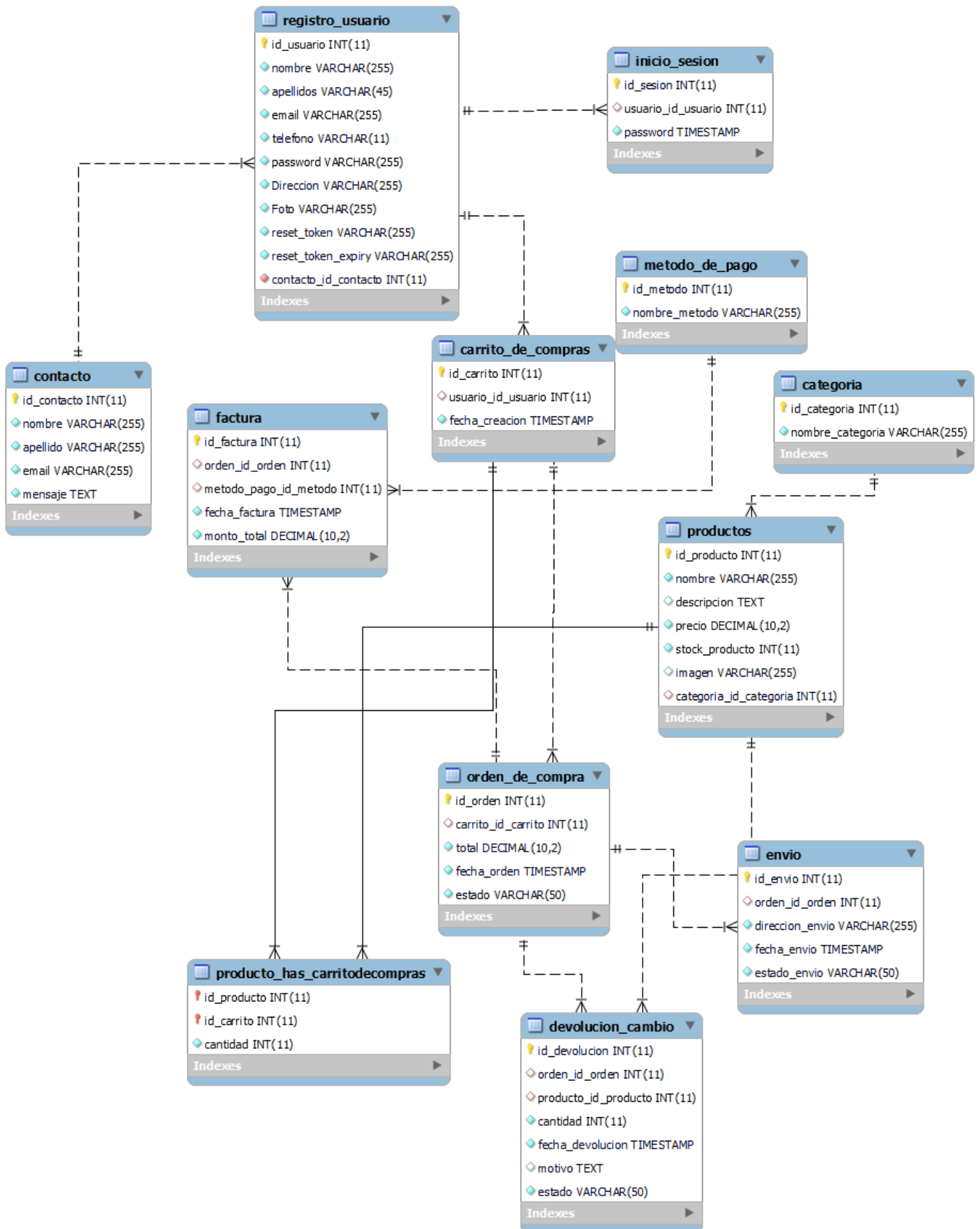




#### 4. Modelo entidad relación de la base de datos

Este **diagrama** muestra cómo las entidades del sistema están relacionadas entre sí, facilitando la comprensión de la estructura de la base de datos.

**Modelo entidad relación:**





En el anterior diagrama de entidad-relación incluye varias entidades no utilizadas en la versión actual del software, pero planificadas para una futura actualización. Estas entidades, como devolución cambio, orden de compra, envío, factura y método de pago, están diseñadas para gestionar operaciones relacionadas con transacciones, logística y pagos, ampliando las funcionalidades del sistema para futuras versiones. Su inclusión anticipa una evolución fluida hacia un software más completo y robusto.

**5. El diccionario de datos;** especifica el contenido de cada tabla en la base de datos, el tipo de dato y las restricciones.

#### market\_online

##### carrito\_de\_compras

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_carrito (Primaria)	int(11)	No		
usuario_id_usuario	int(11)	Sí	NULL	
fecha_creacion	timestamp	No	current_timestamp()	

##### Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_carrito	0	A	No	
usuario_id_usuario	BTREE	No	No	usuario_id_usuario	0	A	Sí	

##### categoria

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_categoria (Primaria)	int(11)	No		
nombre_categoria	varchar(255)	No		

##### Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_categoria	6	A	No	

##### contacto

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_contacto (Primaria)	int(11)	No		
nombre	varchar(255)	No		
apellido	varchar(255)	No		
email	varchar(255)	No		
mensaje	text	No		

##### Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_contacto	2	A	No	

##### devolucion\_cambio

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_devolucion (Primaria)	int(11)	No		
orden_id_orden	int(11)	Sí	NULL	
producto_id_producto	int(11)	Sí	NULL	
cantidad	int(11)	No		
fecha_devolucion	timestamp	No	current_timestamp()	

Columna	Tipo	Nulo	Predeterminado	Comentarios
motivo	text	Sí	NULL	
estado	varchar(50)	No		

## Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_devolucion	0	A	No	
orden_id_orden	BTREE	No	No	orden_id_orden	0	A	Sí	
producto_id_producto	BTREE	No	No	producto_id_producto	0	A	Sí	

## envio

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_envio (Primaria)	int(11)	No		
orden_id_orden	int(11)	Sí	NULL	
direccion_envio	varchar(255)	No		
fecha_envio	timestamp	No	current_timestamp()	
estado_envio	varchar(50)	No		

## Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_envio	0	A	No	
orden_id_orden	BTREE	No	No	orden_id_orden	0	A	Sí	

## factura

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_factura (Primaria)	int(11)	No		
orden_id_orden	int(11)	Sí	NULL	
metodo_pago_id_metodo	int(11)	Sí	NULL	
fecha_factura	timestamp	No	current_timestamp()	
monto_total	decimal(10,2)	No		

## Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_factura	0	A	No	
orden_id_orden	BTREE	No	No	orden_id_orden	0	A	Sí	
metodo_pago_id_metodo	BTREE	No	No	metodo_pago_id_metodo	0	A	Sí	

## inicio\_sesion

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_sesion (Primaria)	int(11)	No		
usuario_id_usuario	int(11)	Sí	NULL	
password	timestamp	No	current_timestamp()	

## Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_sesion	0	A	No	
usuario_id_usuario	BTREE	No	No	usuario_id_usuario	0	A	Sí	

## metodo\_de\_pago

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_metodo (Primaria)	int(11)	No		
nombre_metodo	varchar(255)	No		

## Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_metodo	0	A	No	

## orden\_de\_compra

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_orden (Primaria)	int(11)	No		
carrito_id_carrito	int(11)	Sí	NULL	
total	decimal(10,2)	No		
fecha_orden	timestamp	No	current_timestamp()	
estado	varchar(50)	No		

## Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_orden	0	A	No	
carrito_id_carrito	BTREE	No	No	carrito_id_carrito	0	A	Sí	

## productos

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_producto (Primaria)	int(11)	No		
nombre	varchar(255)	No		
descripcion	text	Sí	NULL	
precio	decimal(10,2)	No		
stock_producto	int(11)	No		
imagen	varchar(255)	Sí	NULL	
categoria_id_categoria	int(11)	Sí	NULL	

## Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_producto	10	A	No	
categoria_id_categoria	BTREE	No	No	categoria_id_categoria	10	A	Sí	

## producto\_has\_carritodecompras

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_producto (Primaria)	int(11)	No		
id_carrito (Primaria)	int(11)	No		
cantidad	int(11)	No		

### Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_producto	0	A	No	
				id_carrito	0	A	No	
id_carrito	BTREE	No	No	id_carrito	0	A	No	

## registro\_usuario

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_usuario (Primaria)	int(11)	No		
nombre	varchar(255)	No		
apellidos	varchar(45)	No		
email	varchar(255)	No		
telefono	varchar(11)	No		
password	varchar(255)	No		
Direccion	varchar(255)	No		
Foto	varchar(255)	No		
reset_token	varchar(255)	No		
reset_token_expiry	varchar(255)	No		

### Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_usuario	2	A	No	
email	BTREE	Sí	No	email	2	A	No	

## 6. Scripts de Instalación

Los scripts de instalación son conjuntos de comandos predefinidos utilizados para automatizar la configuración de entornos, la instalación de dependencias o la preparación de software en un sistema. En el contexto de proyectos de desarrollo web, un script de instalación puede incluir instrucciones para descargar, configurar y preparar un entorno de trabajo, como instalar módulos o paquetes requeridos para que un proyecto funcione correctamente. Por ejemplo, en un proyecto como este que utiliza ReactJS, se utiliza un script como `yarn install` para descargar todas las dependencias listadas en el archivo `package.json`, configurando el entorno de desarrollo. Estos scripts eliminan la necesidad de realizar manualmente cada paso del proceso de instalación, asegurando que el entorno esté correctamente configurado de manera uniforme y rápida. A continuación, se incluirán los scripts necesarios para instalar el sistema de manera automatizada. Esto incluye la configuración de la base de datos, la instalación de dependencias y la inicialización del entorno.

### 6.1. Script de Instalación de la Base de Datos (MySQL):

Para ejecutar el script SQL en phpMyAdmin, primero abre tu navegador y dirígete a <http://localhost/phpmyadmin/> . Una vez dentro de la interfaz de phpMyAdmin, selecciona la base de datos en la que deseas ejecutar el script desde el panel izquierdo o crea una nueva si es necesario. A continuación, haz clic en la pestaña SQL en la parte superior. Esto abrirá un editor de texto donde podrás pegar el siguiente script SQL. Una vez que hayas pegado el script en el editor, simplemente haz clic en el botón continuar para ejecutar las consultas.

```
-- Base de datos: `market_online`
```

```
-- -----  
-- Tabla `carrito_de_compras`  
CREATE TABLE `carrito_de_compras` (  
  `id_carrito` int(11) NOT NULL AUTO_INCREMENT,  
  `usuario_id_usuario` int(11) DEFAULT NULL,  
  `fecha_creacion` timestamp NOT NULL DEFAULT current_timestamp(),  
  PRIMARY KEY (`id_carrito`),  
  KEY `usuario_id_usuario` (`usuario_id_usuario`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- -----  
-- Tabla `categoria`  
CREATE TABLE `categoria` (  
  `id_categoria` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre_categoria` varchar(255) NOT NULL,  
  PRIMARY KEY (`id_categoria`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```

-----
-- Tabla `contacto`
CREATE TABLE `contacto` (
  `id_contacto` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(255) NOT NULL,
  `apellido` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `mensaje` text NOT NULL,
  PRIMARY KEY (`id_contacto`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-----
-- Tabla `devolucion_cambio`
CREATE TABLE `devolucion_cambio` (
  `id_devolucion` int(11) NOT NULL AUTO_INCREMENT,
  `orden_id_orden` int(11) DEFAULT NULL,
  `producto_id_producto` int(11) DEFAULT NULL,
  `cantidad` int(11) NOT NULL,
  `fecha_devolucion` timestamp NOT NULL DEFAULT current_timestamp(),
  `motivo` text DEFAULT NULL,
  `estado` varchar(50) NOT NULL,
  PRIMARY KEY (`id_devolucion`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-----
-- Tabla `envio`
CREATE TABLE `envio` (
  `id_envio` int(11) NOT NULL AUTO_INCREMENT,
  `orden_id_orden` int(11) DEFAULT NULL,
  `direccion_envio` varchar(255) NOT NULL,
  `fecha_envio` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
  `estado_envio` varchar(50) NOT NULL,
  PRIMARY KEY (`id_envio`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-----
-- Tabla `factura`
CREATE TABLE `factura` (
  `id_factura` int(11) NOT NULL AUTO_INCREMENT,
  `orden_id_orden` int(11) DEFAULT NULL,
  `metodo_pago_id_metodo` int(11) DEFAULT NULL,
  `fecha_factura` timestamp NOT NULL DEFAULT current_timestamp(),
  `monto_total` decimal(10,2) NOT NULL,
  PRIMARY KEY (`id_factura`)
)

```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- Tabla `inicio_sesion`
```

```
CREATE TABLE `inicio_sesion` (  
  `id_sesion` int(11) NOT NULL AUTO_INCREMENT,  
  `usuario_id_usuario` int(11) DEFAULT NULL,  
  `password` timestamp NOT NULL DEFAULT current_timestamp(),  
  PRIMARY KEY (`id_sesion`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- Tabla `metodo_de_pago`
```

```
CREATE TABLE `metodo_de_pago` (  
  `id_metodo` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre_metodo` varchar(255) NOT NULL,  
  PRIMARY KEY (`id_metodo`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- Tabla `orden_de_compra`
```

```
CREATE TABLE `orden_de_compra` (  
  `id_orden` int(11) NOT NULL AUTO_INCREMENT,  
  `carrito_id_carrito` int(11) DEFAULT NULL,  
  `total` decimal(10,2) NOT NULL,  
  `fecha_orden` timestamp NOT NULL DEFAULT current_timestamp(),  
  `estado` varchar(50) NOT NULL,  
  PRIMARY KEY (`id_orden`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- Tabla `productos`
```

```
CREATE TABLE `productos` (  
  `id_producto` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(255) NOT NULL,  
  `descripcion` text DEFAULT NULL,  
  `precio` decimal(10,2) NOT NULL,  
  `stock_producto` int(11) NOT NULL,  
  `imagen` varchar(255) DEFAULT NULL,  
  `categoria_id_categoria` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id_producto`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- Tabla `producto_has_carritodecompras`
```

```
CREATE TABLE `producto_has_carritodecompras` (  
  `id_producto` int(11) NOT NULL,
```

```

`id_carrito` int(11) NOT NULL,
`cantidad` int(11) NOT NULL,
PRIMARY KEY (`id_producto`, `id_carrito`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

-----
-- Tabla `registro_usuario`
CREATE TABLE `registro_usuario` (
  `id_usuario` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(255) NOT NULL,
  `apellidos` varchar(45) NOT NULL,
  `email` varchar(255) NOT NULL,
  `telefono` varchar(11) NOT NULL,
  `password` varchar(255) NOT NULL,
  `Direccion` varchar(255) NOT NULL,
  `Foto` varchar(255) NOT NULL,
  `reset_token` varchar(255) NOT NULL,
  `reset_token_expiry` varchar(255) NOT NULL,
  PRIMARY KEY (`id_usuario`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

## 6.2 Instalación de Dependencias para ReactJS:

Para comenzar, abre una terminal o línea de comandos, como la terminal de Visual Studio Code, y navega al directorio raíz del proyecto React utilizando el comando `cd`. Asegúrate de estar en la carpeta donde se encuentra el archivo `package.json`. Una vez dentro, ejecuta el comando `yarn install`. Este comando descargará e instalará todas las dependencias necesarias para tu proyecto, creando la carpeta `node_modules`. Después de que Yarn complete la instalación, verifica que la carpeta `node_modules` esté presente y que en la terminal aparezca un mensaje confirmando la instalación exitosa de todas las dependencias. Si deseas comprobar que todo funciona correctamente, ejecuta el comando `yarn start`, lo que iniciará el servidor de desarrollo y abrirá tu aplicación React en el navegador en `http://localhost:3000/`. Si todo está en orden, deberías ver tu proyecto React funcionando sin problemas.

## 6.3 Instalación y Configuración de XAMPP:

Visita el sitio oficial de XAMPP en <https://www.apachefriends.org> y selecciona la versión adecuada para tu sistema operativo (Windows, macOS o Linux). Descarga el archivo correspondiente: un archivo `.exe` en el caso de Windows o un archivo `.dmg` para macOS. Durante la instalación, se te dará la opción de seleccionar los componentes a instalar. Asegúrate de incluir al menos los siguientes: Apache (servidor web), MySQL (MariaDB) (base de datos) y PHP (lenguaje de programación backend). Componentes adicionales como phpMyAdmin y Perl son opcionales según las necesidades de tu proyecto.



Una vez completada la instalación, abre el "Panel de Control de XAMPP", que puedes encontrar en la carpeta donde instalaste XAMPP (por ejemplo, C:\xampp en Windows) o en tu lista de aplicaciones. Desde el panel de control, inicia los módulos de Apache y MySQL haciendo clic en el botón Start junto a cada uno. Si los servicios se inician correctamente, verás que el fondo de los módulos se pondrá verde, lo que indica que están activos.

Para verificar que Apache está funcionando, abre un navegador y dirígete a <http://localhost/>. Deberías ver la página de inicio de XAMPP, lo que confirma que el servidor Apache está funcionando correctamente. Para administrar tus bases de datos, abre <http://localhost/phpmyadmin/>, lo que te llevará a phpMyAdmin, una interfaz gráfica donde podrás gestionar tus bases de datos MySQL, crear nuevas bases, y ejecutar consultas.

En XAMPP, todos los archivos que desees que sean accesibles a través del servidor Apache deben ubicarse en la carpeta htdocs, que se encuentra dentro del directorio de instalación de XAMPP. En Windows, la ruta predeterminada es C:\xampp\htdocs, mientras que en macOS podría ser /Applications/XAMPP/htdocs/. Dentro de la carpeta htdocs, crea una carpeta para tu proyecto, donde colocarás todos los archivos PHP y recursos relacionados. Estos serán accesibles desde el servidor web local.

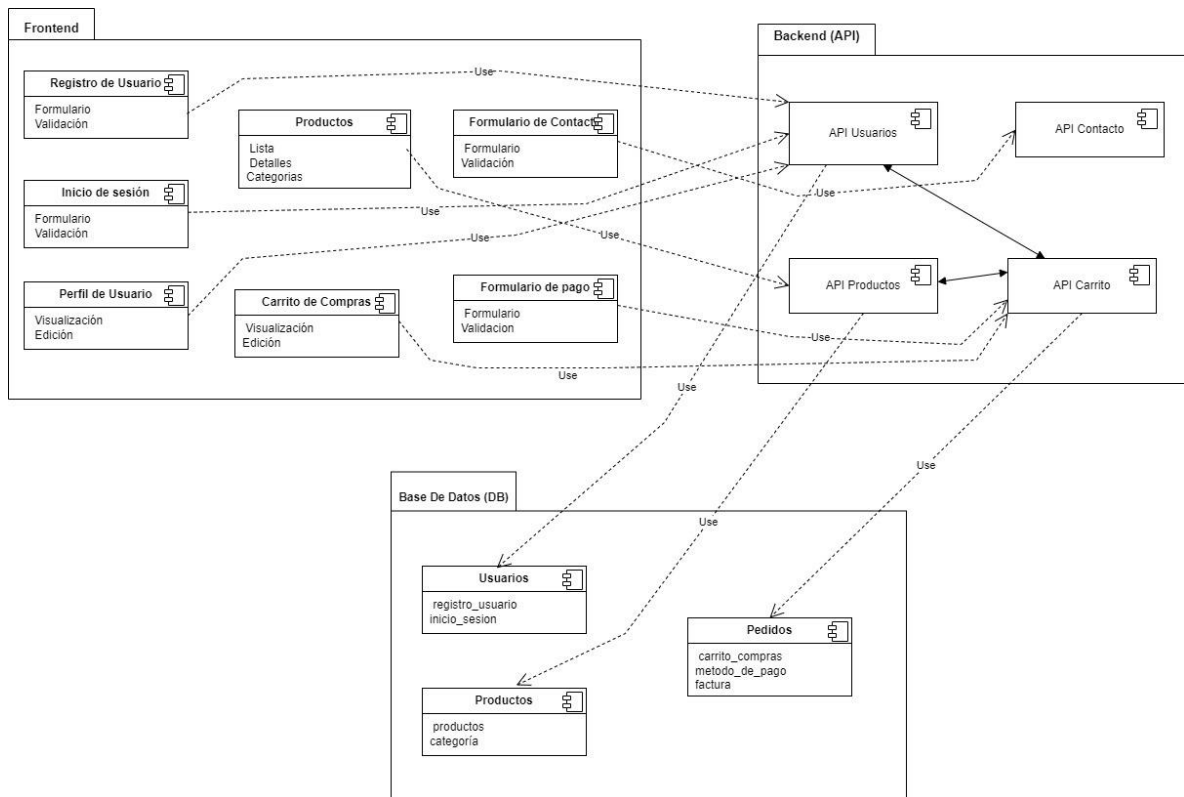
---

## 7. Diagrama de componentes

Este diagrama de componentes ilustra las interacciones clave dentro del sistema, mostrando cómo se comunican los distintos elementos entre sí. En él, se observa cómo el frontend, desarrollado en ReactJS, envía solicitudes al backend a través de APIs RESTful. El backend, implementado en PHP, recibe estas solicitudes, procesa la lógica de negocio y luego interactúa con la base de datos (MySQL o similar) para recuperar o almacenar información. La comunicación entre los componentes se basa en solicitudes HTTP con respuestas en JSON, y el diagrama también destaca las conexiones entre los servidores de frontend y backend, y cómo se gestionan las operaciones de datos en la base de datos. Todo esto asegura un flujo eficiente de datos y una clara separación de responsabilidades dentro del sistema.

**Componentes:**

- **Frontend (ReactJS):** Maneja la interfaz de usuario y las peticiones HTTP (API) hacia el backend.
- **Backend (PHP):** Procesa la lógica de negocio y gestiona la interacción con la base de datos.
- **Base de datos (MySQL):** Almacena los datos persistentes del sistema.



## **Conclusión:**

Este manual técnico proporciona una base sólida para que cualquier desarrollador pueda entender, instalar y mantener el sistema de manera efectiva. Cada sección está diseñada para ofrecer una visión clara de los requisitos, el flujo de trabajo y la arquitectura general de la aplicación, lo que garantiza un proceso de instalación simplificado y estandarizado. Usando tecnologías clave como ReactJS, PHP y MySQL, junto con buenas prácticas de desarrollo y estándares de codificación, el sistema asegura una implementación eficiente, organizada y fácil de mantener.

Además, los diagramas, scripts y herramientas como Yarn y XAMPP facilitan tanto el despliegue como la ejecución de la aplicación en diferentes entornos. Esto contribuye a una experiencia de desarrollo fluida y asegura que cualquier desarrollador pueda continuar con el mantenimiento del sistema sin dificultades.