Informe Técnico Detallado del Proyecto “Motorkai”

Materia: Laboratorio de programación
Alumnos: Vega Esquivel, Angeles
Vega Esquivel, Manuel

----1. Introducción y Propósito del Proyecto

Motorkai es una aplicación web desarrollada como proyecto académico integral, cuyo objetivo principal es simular el funcionamiento de un sistema de gestión para un local de venta de motos, repuestos y accesorios.

El sitio ofrece una experiencia fluida tanto para el cliente como para el administrador, incorporando roles, autenticación, operaciones CRUD (crear, leer, actualizar y eliminar), gestión de productos, marcas, categorías, y un panel de control con visualización de ventas mediante gráficos dinámicos.

-----2. Stack Tecnológico y Lenguajes de Programación (LAMP)

El desarrollo de Motorkai se sustentó en un stack clásico LAMP (Linux, Apache, MySQL, PHP), complementado con tecnologías web modernas para mejorar la experiencia de usuario.**2.1. Frontend**

- **HTML5:** Estructura semántica de todas las vistas, garantizando accesibilidad y compatibilidad.
- **CSS3:** Estilización avanzada con transiciones suaves, sombras, gradientes y una paleta cromática corporativa basada en el naranja, negro y gris.
- **JavaScript (Vanilla):** Validaciones dinámicas, efectos interactivos y soporte para operaciones del carrito de compras.
- **Bootstrap 5 (Complementario):** Utilizado en la maquetación del Dashboard y formularios para una disposición responsiva y coherente.

2.2. Backend

- **PHP 8.x:** Lenguaje base para la lógica del servidor. Implementa la gestión de sesiones, control de roles (cliente, vendedor, administrador) y conexión a la base de datos.
- **MySQL:** Base de datos relacional encargada del almacenamiento persistente de

usuarios, productos, ventas, marcas, categorías y registros de autenticación.
- **PDO (PHP Data Objects):** Capa de acceso a datos segura y moderna, utilizada para prevenir inyecciones SQL y mejorar la escalabilidad del código.

2.3. Entorno de Desarrollo

- **Servidor local:** XAMPP, con Apache y MySQL.
- **IDE:** Visual Studio Code.
- **Control de versiones:** Git (estructura modular y separada por componentes: /config, /includes, /assets, /products, /brands, /categories, /dashboard).

-----3. Arquitectura y Organización del Proyecto

Motorkai adopta una arquitectura modular basada en componentes PHP reutilizables. La separación de responsabilidades permite mantener una estructura limpia, escalable y mantenible, siguiendo el principio MVC simplificado (Modelo, Vista y Controlador lógico en PHP).

Módulo	Componente	Descripción
/assets	style.css	estilos personalizados
/assets/js	validations.js	validación de datos
/config	`db.php`	Conexión a la base de datos.
	`auth.php`	Lógica de sesiones y roles.
/bootstrap	`bootstrap.php`	Definición de constantes (BASE_URL, TÍTULO)
/includes	`header.php`	Encabezado general unificado.

	navbar2.php	barra de navegación.
	footer.php	Pie de página unificado.
Gestión CRUD	/products, /brands, /categories	Módulos con list.php, create.php, edit.php, delete.php.
/dashboard	dashboard.php sales_chart.php	Panel de control (KPIs, Gráficos Chart.js) Visualización de ventas (gráfico dinámico).
/index	index.php	landing page/ home publica
/uploads		Fotos de productos subidas dinámicamente
/register	register.php	registro de usuarios
/cart	cart.php	carrito funcional conectado a la bd.
/catalog	index_client.php	catálogo visible para el cliente.

Justificación de la Arquitectura de Archivos

1. **Separación de Preocupaciones:** Los módulos de configuración (config/), interfaz (includes/), y lógica de negocio (carpetas por entidad: products/, sales/) están aislados. Esto facilita la depuración y la extensión.
2. **Centralización de Recursos:** assets/ contiene todos los recursos estáticos (CSS, imágenes) y uploads/ gestiona el contenido dinámico (fotos de productos,

avatares).

3. **Flujo de autenticación Claro:** La lógica de inicio de sesión y registro se encuentra en la carpeta **`auth/`**, separando el proceso de autenticación del resto de la aplicación.
4. **Usabilidad de Rutas:** La mayoría de los archivos de ejecución clave (`index.php`, `catalog.php`, `cart.php`) se encuentran en la **raíz (`Motorkai_final/`)**, simplificando las URL de acceso para los usuarios y la lógica de enrutamiento del lado del servidor.

-----**4. Lógica de Roles y Autenticación (RBAC)**

El sistema implementa un control de acceso basado en roles (Role-Based Access Control). Cada usuario posee un perfil que define qué secciones puede visualizar o modificar.

Rol	**Permisos principales**
Cliente	Navegar el catálogo, agregar productos al carrito, realizar pedidos.
Administrador	Control total: gestionar usuarios, categorías, marcas y monitorear estadísticas globales. Gestionar productos, ver ventas, actualizar stock.

El control se realiza mediante validación de sesión (`$_SESSION['role']`), con redirección automática al login si no se cumplen los permisos.

-----**5. Funcionalidades Principales Desarrolladas**

Motorkai incluye las siguientes características clave:

- **Autenticación y registro** de usuarios con verificación de credenciales.
- **Panel de administración dinámico**, con acceso restringido según rol.
- **Gestión CRUD completa** de productos, marcas y categorías.
- **Carrito de compras** (frontend y backend funcional).
- **Dashboard visual interactivo:** Implementado con Chart.js, muestra la evolución de ventas y estadísticas generales.
- **Gestión de Medios:** Carga de imágenes y visualización desde el servidor local.
- **Experiencia de Usuario:** Mensajes de estado amigables (creado, actualizado, eliminado) y sistema de redirecciones seguras (evita acciones no autorizadas).

-----**6. Prácticas de Desarrollo Implementadas**

Categoría	Prácticas Adoptadas
Calidad de Código	Código modular y reutilizable. Separación lógica entre presentación (HTML/CSS) y lógica (PHP). Comentarios técnicos y semánticos.
Seguridad y Validación	Validaciones del lado del cliente y del servidor. Uso de *prepared statements* con PDO para prevenir inyecciones SQL. Control de errores con *try/catch* y manejo de excepciones.
Diseño y UX	Diseño *responsive* (adaptado a dispositivos móviles). Integración de *includes* globales (`header.php` y `footer.php`) para uniformidad. Sistema de navegación persistente con estado de usuario ("Hola, Usuario").
Base de Datos	Estructura jerárquica clara: productos vinculados a categorías y marcas mediante claves foráneas.

----**7. Identidad Visual y Psicología del Color**

La identidad de Motorkai fue diseñada bajo una línea moderna, industrial y sofisticada, reflejando fuerza, velocidad y profesionalismo.

Color	Código	Significado psicológico	Aplicación
Naranja	`#FF6600`	Energía, movimiento, entusiasmo. Estimula la acción y refuerza la	Botones, acentos visuales, íconos de acción.

		idea de dinamismo y potencia.	
Negro	#121212	Elegancia, autoridad, tecnología. Crea contraste y profundidad.	Fondo principal del sitio.
Gris oscuro	#1F1F1F	Neutralidad, equilibrio y sofisticación. Suaviza la intensidad del negro.	Formularios, tarjetas y paneles.

El contraste entre naranja vibrante y fondo oscuro refuerza la legibilidad, da protagonismo a las imágenes de productos y aporta una sensación de marca *premium*.

-----**8. Conclusión y Valor del Proyecto**

Motorkai consolida un desarrollo completo y profesional, integrando:

- Lógica de negocio real,
- Diseño de interfaz intuitivo,
- Arquitectura escalable,
- Seguridad de datos,
- Identidad visual coherente con su propósito comercial.

El resultado es un proyecto web funcional, moderno y visualmente atractivo, aplicando los fundamentos del desarrollo *full stack* con criterios de ingeniería de software y diseño

estratégico.

🏁 **Conclusión :** Motorkai es un ejemplo de cómo la programación, la estética y la psicología del color pueden converger para construir una experiencia de marca sólida y memorable. El equilibrio entre código y diseño, entre estructura y emoción, hace que el proyecto logre transmitir no solo funcionalidad, sino carácter e identidad.