



Universidad de Guadalajara

**Centro Universitario de Ciencias
Exactas e Ingenierías**

**División de Tecnologías para la
Integración Ciber-Humana**

Departamento de Ciencias Computacionales

Tarea No.12 Manual Técnico

**Seminario de Solución de Problemas de
Ingeniería en Software**

Profesor: Héctor Nomar González Flores

Sección: D06

Equipo: Neurodivergentes

Fecha de entrega: 21 de mayo del 2025

Índice

1. Objetivos	3
1.1. Objetivo general:	3
1.2. Objetivos particulares:	3
2. Alcance	4
3. Requerimientos técnicos	5
3.1. Requisitos de hardware:	5
3.2. Requisitos de software:	5
4. Herramientas usadas para el desarrollo	6
5. Instalación	7
6. Configuración	8
7. Arquitectura del sistema	9
8. Usuarios	10
9. Contingencias y soluciones	11
10. Conclusión general	12

1. Objetivos

1.1. Objetivo general:

Desarrollar un sistema de punto de venta (POS) especializado para la empresa "Shark Energy", con el fin de mejorar y automatizar los procesos de venta, gestión de inventario, proveedores, mantenimiento, garantías y atención al cliente. Este sistema tiene como finalidad ofrecer una solución robusta, escalable y funcional que funcione de manera autónoma respecto a la tienda en línea de la empresa.

1.2. Objetivos particulares:

- Diseñar una estructura de base de datos normalizada que permita una organización eficiente de la información.
- Permitir a los empleados registrar ventas, compras y mantenimientos de manera ágil.
- Desarrollar una interfaz amigable que facilite el uso del sistema a usuarios con distintos niveles técnicos.
- Proveer funcionalidades para generar reportes de ventas e inventario.
- Registrar y mantener actualizado el estado del inventario y productos.
- Mantener un historial detallado de los servicios y garantías brindadas a los clientes.
- Integrar funcionalidades para gestión de proveedores, compras y entradas al inventario.
- Documentar el sistema adecuadamente para facilitar su mantenimiento y futura extensión.

2. Alcance

El sistema Shark Energy POS está diseñado para cubrir las necesidades de venta, inventario y atención al cliente de una tienda física. No depende de la página web de la empresa, permitiendo un funcionamiento independiente que centraliza la información de manera local.

Funciones incluidas en el sistema:

- Registro de ventas a clientes.
- Gestión de inventario (entrada, salida, consulta de stock).
- Registro de compras a proveedores.
- Control de garantías y mantenimientos realizados.
- Generación de reportes automáticos.
- Registro y administración de clientes y proveedores.
- Visualización del historial de servicios.

Limitaciones del sistema:

- No opera de forma remota ni en la nube.
- No integra pasarelas de pago digitales.
- No incluye funcionalidades contables avanzadas.
- No contempla la sincronización en tiempo real con la tienda en línea.
- Maneja un solo rol de usuario administrativo.

Este alcance permite que la empresa tenga un mayor control interno de sus procesos sin depender de terceros, optimizando sus recursos tecnológicos y humanos.

3. Requerimientos técnicos

3.1. Requisitos de hardware:

- Procesador: Intel Core i3 de 8ª generación o equivalente.
- Memoria RAM: Mínimo 4 GB (8 GB recomendados).
- Almacenamiento disponible: Al menos 1 GB libre para instalación completa.
- Resolución de pantalla: Mínimo 1280x720 px.
- Accesorios: Impresora de tickets (opcional), lector de código de barras (USB), mouse y teclado.

3.2. Requisitos de software:

- Sistema Operativo: Windows 10/11 o Linux Ubuntu 20.04 o superior.
- Servidor local: XAMPP, Laragon o Docker (Apache + MySQL + PHP).
- PHP 8.1 o superior.
- Composer (gestor de dependencias de PHP).
- MySQL o MariaDB (motor de base de datos).
- Navegador web actualizado (Google Chrome, Firefox, etc.).

Es fundamental asegurar que el sistema se instale en un entorno estable, con los componentes necesarios correctamente configurados para garantizar su rendimiento óptimo.

4. Herramientas usadas para el desarrollo

Para la construcción del sistema Shark Energy POS, se utilizaron herramientas modernas que garantizan robustez, escalabilidad y facilidad de mantenimiento:

- Lenguaje de programación: PHP 8.1
- Framework principal: Laravel (versión 10), por su estructura MVC y seguridad integrada.
- Base de datos: MySQL 8, por su estabilidad, rendimiento y soporte comunitario.
- Frontend: Blade templating engine, HTML5, CSS3.
- Control de versiones: Git para control y GitHub como repositorio remoto.
- IDE: Visual Studio Code, por su extensibilidad y soporte de herramientas.
- Diseño de diagramas: Draw.io para casos de uso, clases y arquitectura.
- Documentación: Microsoft Word y PDF.
- Planificación y metodología: Scrum como enfoque ágil para división de tareas y entrega incremental.

Estas herramientas permitieron al equipo trabajar de forma colaborativa, con ciclos iterativos y entregables funcionales de forma progresiva.

5. Instalación

El proceso de instalación del sistema está pensado para ser sencillo, siempre que se cuente con el entorno adecuado configurado previamente. Se recomienda usar un entorno de desarrollo local como XAMPP o Laragon.

Pasos para la instalación:

1. Clonar el repositorio:

```
git clone https://github.com/usuario/shark-energy-pos.git
cd shark-energy-pos
```

2. Instalar dependencias:

```
composer install
```

3. Copiar y configurar el archivo .env:

```
cp .env.example .env
```

Luego editar el archivo .env para agregar las credenciales de la base de datos.

4. Generar la clave de aplicación:

```
php artisan key:generate
```

5. Crear la base de datos:

- o Ingresar a phpMyAdmin o consola de MySQL.
- o Crear una base de datos llamada shark_energy.

6. Ejecutar migraciones:

```
php artisan migrate
```

7. Iniciar el servidor:

```
php artisan serve
```

Una vez hecho esto, se podrá acceder a la aplicación mediante <http://localhost:8000> desde cualquier navegador.

6. Configuración

Después de instalar el sistema, es necesario realizar ciertas configuraciones para que funcione correctamente y pueda adaptarse a las necesidades específicas del negocio.

Configuraciones clave en el archivo:

- APP_NAME=SharkEnergyPOS
- DB_DATABASE=shark_energy
- DB_USERNAME=root
- DB_PASSWORD=tu_clave

Otras configuraciones:

- Establecer zona horaria (APP_TIMEZONE=America/Mexico_City).
- Definir nivel de log (LOG_LEVEL=debug).
- Configurar idioma (APP_LOCALE=es).

Configuración de roles y usuarios: Aunque la versión actual contempla un solo rol, se pueden preparar configuraciones adicionales para niveles de acceso futuros mediante seeders o migraciones personalizadas.

Configuración de correo (opcional): En caso de habilitar notificaciones por correo:

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=usuario
MAIL_PASSWORD=clave
MAIL_ENCRYPTION=null
```

Estas configuraciones permiten adaptar el sistema a entornos reales de operación y hacer pruebas adecuadas durante el desarrollo.

7. Arquitectura del sistema

El sistema Shark Energy POS está diseñado bajo la arquitectura Modelo-Vista-Controlador (MVC), lo que permite separar claramente la lógica de negocio, la presentación y el acceso a datos. Esta arquitectura es ideal para proyectos escalables, modulares y fáciles de mantener.

Componentes principales:

- **Modelo:** Representa las entidades del sistema (Cliente, Producto, Venta, Empleado, etc.) y se encarga de interactuar con la base de datos.
- **Vista:** Conjunto de archivos Blade que componen la interfaz gráfica del sistema.
- **Controlador:** Gestiona las solicitudes del usuario, procesa la lógica y coordina las respuestas.

Flujo general del sistema:

1. El usuario interactúa con la interfaz.
2. El controlador recibe la solicitud y procesa la lógica.
3. El modelo accede a los datos necesarios.
4. Se actualiza la vista con la información.

Ventajas del enfoque MVC:

- Permite mantener el código organizado y reutilizable.
- Facilita las pruebas unitarias y el mantenimiento.
- Separa la lógica de presentación de la lógica del negocio.

Además, se emplean migraciones y seeders para gestionar la estructura de la base de datos y poblarla con datos de prueba, lo que mejora el control de versiones y la replicabilidad del sistema.

8. Usuarios

El sistema contempla dos tipos de usuarios que interactúan con diferentes niveles:

Usuarios de la base de datos:

- `admin_user`: Tiene permisos totales para creación, modificación y eliminación de datos.
- `readonly_user`: Solo lectura, útil para consultas y reportes sin riesgo de alterar información.

Usuarios de la aplicación:

- En su versión actual, el sistema trabaja con un único rol de usuario: Administrador, quien tiene acceso completo a todas las funcionalidades.

Futuras implementaciones sugeridas:

- Vendedor: Acceso limitado a módulos de ventas y clientes.
- Encargado de inventario: Solo puede modificar productos y entradas al inventario.
- Soporte técnico: Acceso a mantenimientos y garantías.

Esta estructura garantiza una base sólida para escalabilidad en entornos reales con varios niveles de personal.

9. Contingencias y soluciones

En cualquier sistema informático, es fundamental prever posibles fallos y tener una estrategia de contingencia para asegurar la continuidad del servicio. A continuación, se describen algunas contingencias comunes y las soluciones propuestas:

Contingencia	Solución
Fallo en la conexión a la base de datos	Verificar configuración en <code>.env</code> y reiniciar el servidor local.
Eliminación accidental de registros	Implementar borrado lógico (soft delete) y sistema de recuperación.
Falla del equipo físico	Mantener copias de seguridad diarias en disco externo o nube.
Actualización fallida	Usar ramas en Git para pruebas antes de mezclar cambios a producción.
Olvido de credenciales	Implementar recuperación de contraseña vía correo o restablecimiento manual.
Cambios masivos de datos	Crear scripts para edición masiva y respaldo previo a su ejecución.

Estas medidas permiten una respuesta ágil ante problemas operativos, evitando pérdidas de información o interrupciones prolongadas del servicio.

10. Conclusión general

El desarrollo del sistema Shark Energy POS representa un avance significativo en la digitalización y automatización de los procesos internos de la empresa. A través del diseño modular, el uso de tecnologías modernas y la implementación de prácticas de desarrollo ágil, se ha logrado construir una herramienta funcional que responde a las necesidades reales del negocio.

Este manual técnico es una parte clave del proyecto, ya que no solo documenta su estructura y funcionamiento, sino que también proporciona una guía clara para su instalación, mantenimiento y posible evolución. La existencia de esta documentación facilita la colaboración entre equipos, el soporte técnico y la integración de nuevos desarrolladores en futuras etapas.

Finalmente, comprender y aplicar metodologías ágiles, así como generar una documentación clara y completa, son prácticas esenciales para el éxito de cualquier proyecto de ingeniería en software. Este proyecto es testimonio del valor de planear, modelar, codificar y documentar de manera estructurada cada una de las fases de desarrollo.