Proyecto de Software

DOCUMENTO TECNICO DE LA APLICACIÓN INVENTORY SYSTEM

Diego Sebastián Pinzón González

Patricia López Sánchez

Damian Esteban Alarcon Pinzon

Profesora

Tatiana Lizbeth Cabrera Vargas

Corporación Universitaria Iberoamericana

Facultad de Ingeniería

Ingeniería de Software

2025

CONTENIDO

1. INTRODUCCIÓN

Este documento describe el diseño técnico del sistema POS desarrollado para pequeños y medianos negocios. El sistema tiene como objetivo automatizar procesos clave como la gestión de inventario, el registro de ventas, la generación de reportes y la administración de usuarios. Está pensado para su uso en entornos locales, sin necesidad de conexión a internet ni servicios de hosting. Su implementación en ambientes con XAMPP o WAMP facilita la instalación para usuarios sin experiencia técnica avanzada.

2. ARQUITECTURA DEL SISTEMA

Tipo de Arquitectura: Modelo – Vista – Controlador en red local

Tecnologías

- Frontend: HTML5, CSS3, JavaScript (Bootstrap)
- Backend: PHP 8+
- Base de datos: MySQL 8+
- Servidor Web: Apache
- Despliegue: Local (localhost) utilizando XAMPP.

La arquitectura se basa en el modelo MVC (Modelo-Vista-Controlador) para separar la lógica de negocio de la interfaz y el acceso a datos. Esto mejora el mantenimiento y la escalabilidad. Se puede escalar a un entorno web externo sin necesidad de rehacer la estructura del sistema.

3. JUSTIFICACIÓN TECNOLÓGICA

Se eligió PHP por su facilidad de uso, curva de aprendizaje accesible y amplia comunidad. PHP permite el desarrollo rápido de aplicaciones web con buen rendimiento en entornos locales. MySQL se eligió por su estabilidad, eficiencia en consultas y excelente integración con PHP. Ambas herramientas son gratuitas, lo que reduce costos para negocios emergentes.

4. MÓDULOS DEL SISTEMA

- Gestión de Usuarios: Alta, baja, modificación y control de accesos. Controlado desde la tabla "usuarios", que incluye roles (perfil), contraseñas encriptadas y estados.

- Inventario: Registro, edición y control de productos con asociación a categorías. La tabla "productos" está relacionada con "categorias" mediante "id_categoria".
- Ventas: Registro de ventas con cliente, vendedor, productos y totales. Los productos se almacenan como JSON dentro de la tabla "ventas", facilitando el manejo de ventas con múltiples productos.
- Clientes: Gestión de información detallada, historial de compras y seguimiento.
- Reportes: Generación de gráficas y tablas por producto, cliente y fecha. Permite exportar información a formatos como Excel o PDF para facilitar la toma de decisiones.

5. BASE DE DATOS

La base de datos se llama "posuniversidad" y contiene las siguientes relaciones clave:

- "productos" está relacionado con "categorias" por "id_categoria".
- "ventas" tiene claves foráneas "id_cliente" y "id_vendedor", que se vinculan con "clientes" y "usuarios" respectivamente.
- Cada venta incluye un campo "productos" en formato JSON con los productos vendidos, sus cantidades y totales.

Relaciones Principales:

- Un cliente puede tener muchas ventas ("clientes" a "ventas": 1:N)
- Un usuario (vendedor) puede registrar muchas ventas ("usuarios" a "ventas": 1:N)
- Una categoría puede tener muchos productos ("categorias" a "productos": 1:N)

6. CASOS DE USO TÉCNICOS

- CU01: Registrar producto (alta en tabla "productos")
- CU02: Realizar venta y generar comprobante (alta en "ventas", actualización de "stock" en "productos")
- CU03: Consultar productos por stock (consulta filtrada en "productos")
- CU04: Visualizar ventas por rango de fechas (consulta agrupada en "ventas")
- CU05: Administrar usuarios del sistema (alta, baja, cambio de contraseña en "usuarios")

7. PRUEBAS

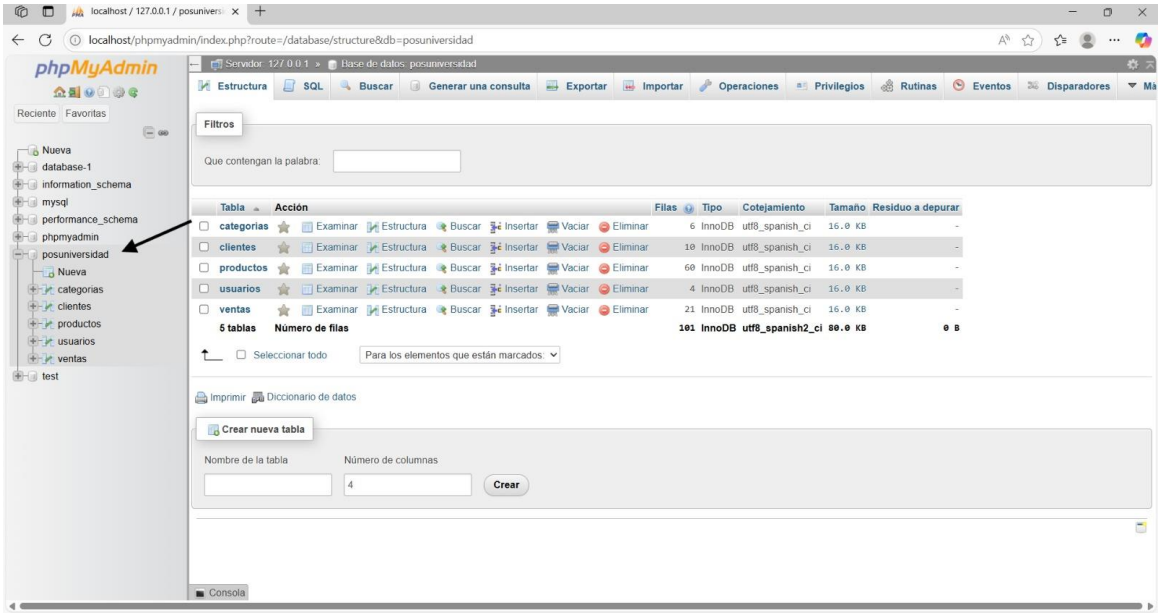
- Pruebas Unitarias: Validación de funciones en PHP como cálculo de totales, actualización de stock.
- Pruebas Funcionales: Flujo completo desde agregar producto al carrito hasta registrar venta.

- Pruebas de Seguridad: Inyecciones SQL, validación de sesiones, acceso según perfil.
- Pruebas de Interfaz: Comprobación de formularios, botones, alertas y gráficas.

8. DESPLIEGUE LOCAL

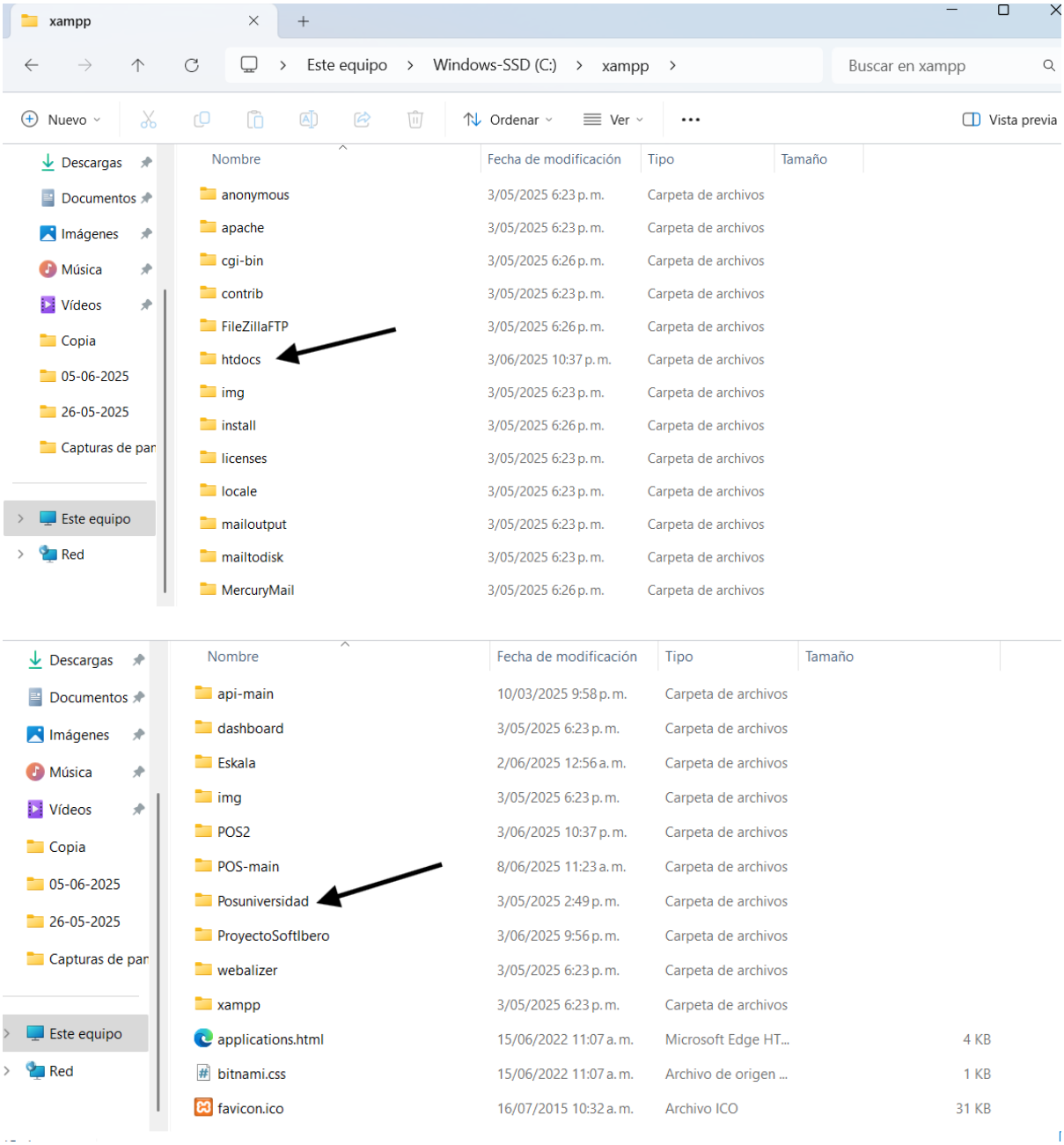
Requisitos:

- Instalación de XAMPP (PHP 8+, MySQL 8+)
- Importar "posuniversidad.sql" desde "Repositorio de github"



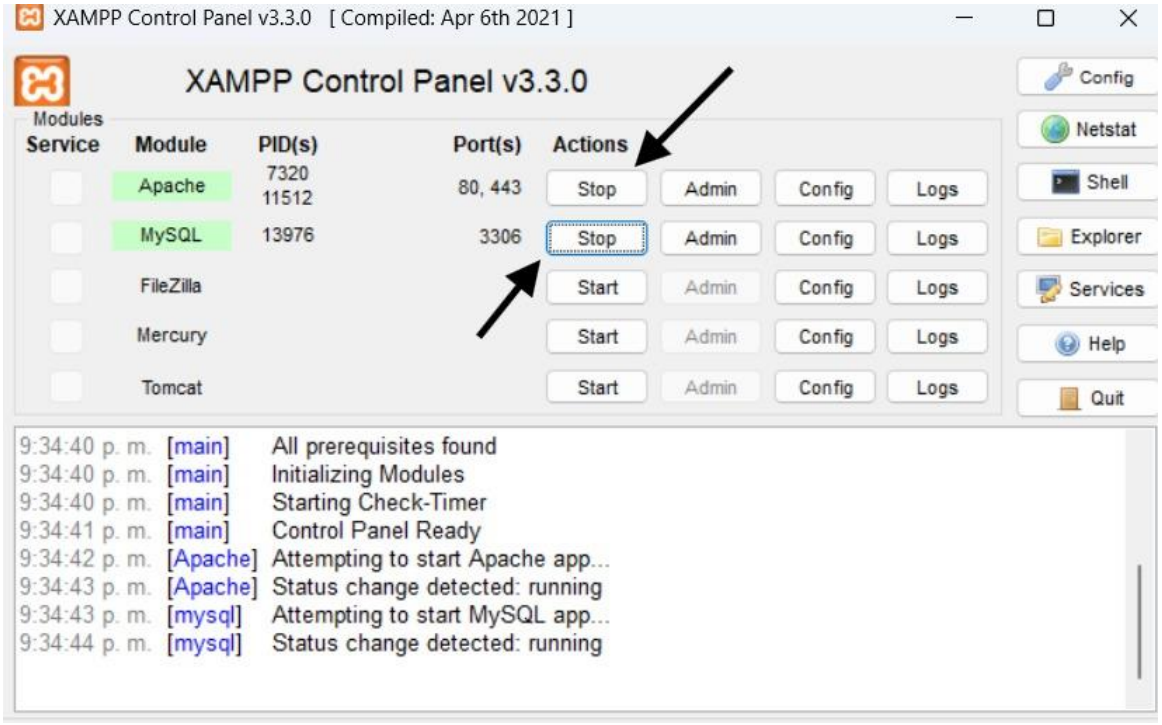
- Colocar los archivos en la carpeta "htdocs"

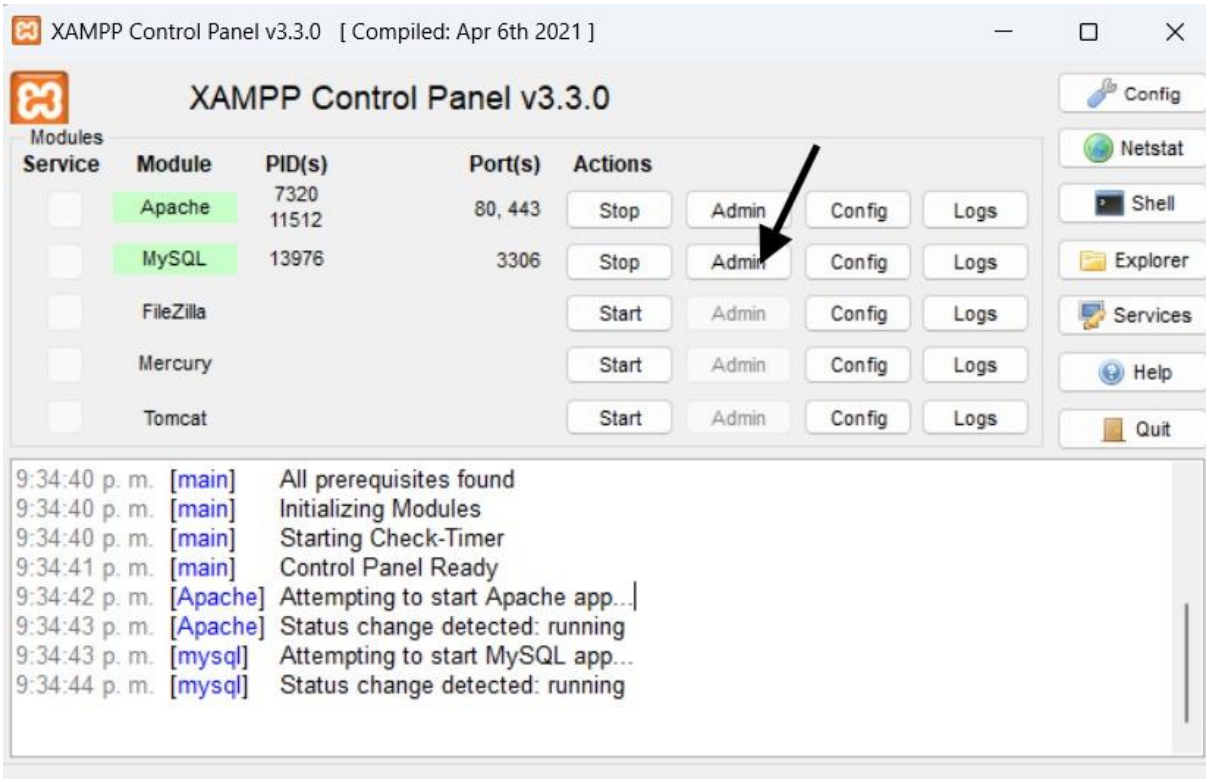
  RUTA LOCAL: C:\xampp\htdocs



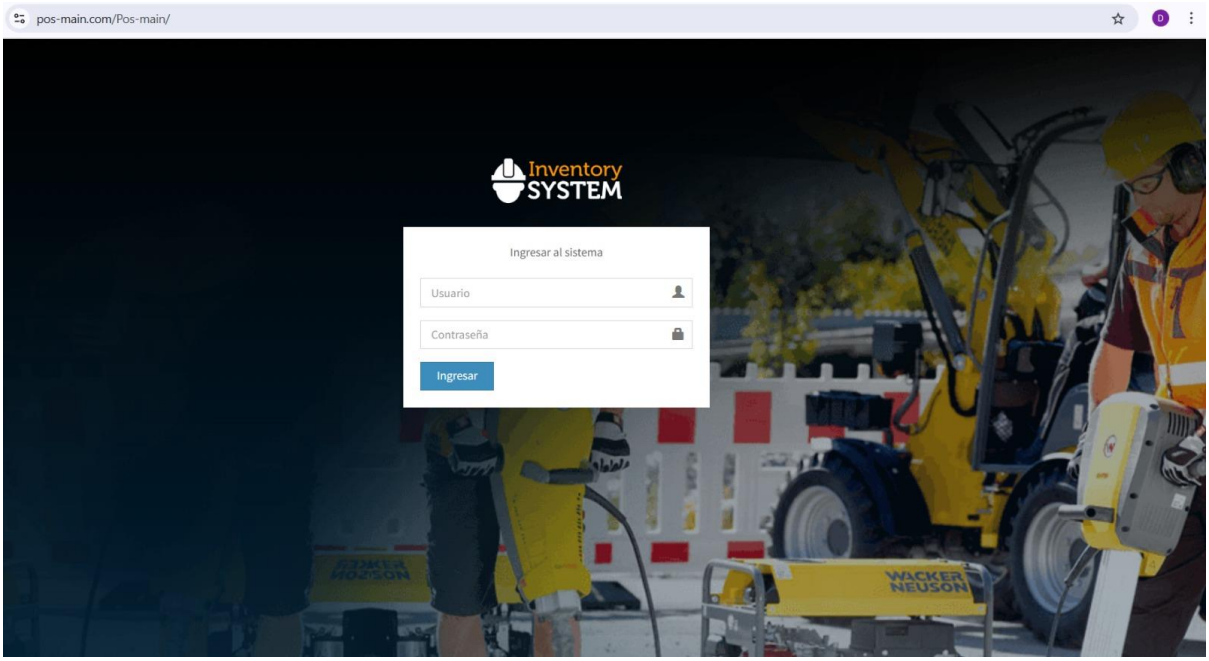
Pasos:

- 1. Iniciar Apache y MySQL desde el panel de XAMPP.





- 2. Navegar a https://pos-main.com/Pos-main/
  NOTA: Se crea un dominio local para correr el proyecto.



- 3. Iniciar sesión con usuario de pruebas: usuario: “admin”, contraseña: “admin”

9. REQUISITOS DEL SISTEMA

- Servidor Local: XAMPP con PHP 8+ y MySQL 8+
- Cliente: Navegador actualizado (Chrome, Firefox)
- Sistema Operativo: Windows 10 o superior, o distribución Linux con soporte PHP

10. CONCLUSIONES TÉCNICAS

El sistema POS desarrollado con PHP y MySQL para entorno local ofrece una solución robusta, segura y de fácil instalación. Su estructura modular y su base de datos relacional están diseñadas para facilitar futuras expansiones como facturación electrónica, backup automatizado, o migración a un entorno web externo. La integración con Apache y el uso de XAMPP permiten que cualquier usuario sin conocimientos avanzados pueda poner en marcha el sistema de manera eficiente.

11. CONTROL DE VERSIONES Y MANTENIMIENTO

- Herramienta de control de versiones: Git
- Repositorio recomendado: GitHub o GitLab
- Buenas prácticas:
- Uso de ramas por personas (main/damian, patricia/diego1)
- Commits semánticos y documentados
- Versionado por etiquetas (v1.0, v1.1, etc.)

12. SEGURIDAD Y BUENAS PRÁCTICAS

- Contraseñas encriptadas con "crypt()"
- Validación de formularios del lado del cliente y del servidor
- Prevención de inyecciones SQL mediante sentencias preparadas (PDO o MySQLi)
- Control de acceso por roles y validación de sesiones activas
- Escapado de salidas para evitar XSS

13. ESCALABILIDAD Y EVOLUCIÓN

El sistema está diseñado de forma modular, lo que permite:

- Integrar nuevos módulos (por ejemplo: facturación electrónica o gestión de proveedores)
- Migrar fácilmente la base de datos a un servidor web
- Integración con servicios REST en futuras versiones

14. CRÉDITOS DEL DESARROLLADOR

- Nombre: [Patricia López Sánchez]

- Correo: [plopezsa@estudiante.ibero.edu.co]

- Nombre: [Damian Esteban Alarcon Pinzon]

- Correo: [dalarc14@estudiante.ibero.edu.co]

- Nombre: [Diego Sebastián Pinzón González]

- Correo: [dpinzo21@estudiante.ibero.edu.co]

- Tecnologías dominadas: PHP, MySQL, HTML, CSS, JavaScript, (Bootstrap)

- Fecha del desarrollo: [06/2025]