

Universidad de San Carlos de Guatemala
Centro Universitario de Occidente
División de Ciencias de la Ingeniería

Teoría de Sistemas
Laboratorio



Práctica 1
Hola Mundo Web
Manual Técnico

Jorge Anibal Bravo Rodríguez
202131782
#3

HERRAMIENTAS USADAS

XAMPP

XAMPP es un paquete de software libre que integra varias herramientas esenciales para el desarrollo web en un único instalador. Incluye el servidor web Apache, el motor de bases de datos MariaDB/MySQL, el lenguaje de programación PHP y otros componentes como Perl y phpMyAdmin. Su objetivo principal es simplificar la instalación de un entorno de desarrollo local (en Windows, Linux y macOS), evitando la configuración manual de cada servicio. Se utiliza ampliamente para pruebas, prototipos y aprendizaje antes de desplegar aplicaciones en servidores de producción.

MySQL/MariaDB

MySQL y MariaDB son sistemas de gestión de bases de datos relacionales (RDBMS) que permiten organizar, almacenar y consultar información estructurada mediante el lenguaje SQL (Structured Query Language).

- MySQL es uno de los motores más usados en aplicaciones web, especialmente en conjunto con PHP.
- MariaDB es un fork desarrollado por la comunidad tras la compra de MySQL por Oracle. Mantiene compatibilidad con MySQL, pero mejora el rendimiento, añade nuevas funcionalidades y garantiza que siga siendo totalmente libre.

Ambos se utilizan en proyectos de todo tipo, desde pequeños sistemas hasta aplicaciones empresariales de gran escala.

PHP

PHP (Hypertext Preprocessor) es un lenguaje de programación interpretado que se ejecuta en el lado del servidor. Está especialmente orientado al desarrollo de aplicaciones web dinámicas, ya que permite generar contenido HTML en tiempo real, procesar formularios, manejar sesiones, conectarse a bases de datos y construir APIs. Su sintaxis es sencilla y está estrechamente integrada con HTML, lo que lo convierte en una de las tecnologías más utilizadas en entornos como WordPress, Drupal o sistemas hechos a medida.

Visual Studio Code

Es un editor de código fuente multiplataforma desarrollado por Microsoft, diseñado para programadores y desarrolladores de software. Ofrece soporte nativo para múltiples lenguajes de programación y se puede extender mediante un amplio ecosistema de extensiones. Entre sus características principales están: resaltado de sintaxis, autocompletado inteligente, depuración integrada, integración con Git/GitHub, terminal embebida y soporte para entornos de desarrollo remoto. Su diseño ligero, rapidez y flexibilidad lo convierten en uno de los editores más utilizados a nivel mundial.

Solus Linux

Es una distribución independiente de Linux (no basada en Debian, Arch ni Red Hat) enfocada en ofrecer un sistema operativo optimizado para el uso en escritorio. Se caracteriza por su estabilidad, rendimiento y simplicidad de uso, ya que está pensado para usuarios que buscan una experiencia “lista para usar”. Cuenta con su propio gestor de paquetes eopkg y ofrece entornos de escritorio como Budgie (diseñado por el mismo equipo de Solus), GNOME, MATE y KDE Plasma.

Git

Git es un sistema de control de versiones distribuido, diseñado para gestionar proyectos de software y coordinar el trabajo de múltiples desarrolladores. Permite registrar el historial completo de cambios en los archivos, crear ramas para el desarrollo paralelo, fusionar modificaciones y revertir versiones anteriores. Al ser distribuido, cada desarrollador posee una copia completa del repositorio, lo que permite trabajar sin conexión y sincronizar los cambios posteriormente. Es una de las herramientas fundamentales en el desarrollo de software moderno, tanto en proyectos individuales como colaborativos.

Github

GitHub es una plataforma en la nube que proporciona alojamiento de repositorios Git, permitiendo a los desarrolladores trabajar de manera colaborativa en proyectos de software. Además de ofrecer control de versiones, GitHub integra funciones de gestión de proyectos, seguimiento de incidencias (issues), pull requests, wikis, integración continua (CI/CD), despliegue automatizado y seguridad del código. Su entorno facilita la colaboración entre equipos de desarrollo distribuidos y fomenta la comunidad de software libre, siendo actualmente uno de los principales centros de proyectos de código abierto y privados en todo el mundo.

CONFIGURACIONES INICIALES DE XAMPP

Por defecto, se expone la ruta /opt/lampp/htdocs/ para linux

En mi caso, utilizaré la ruta /home/jorge/Teoría_Sistemas/

Para cambiarlo:

1. Abrir el archivo de configuracion de XAMPP

```
sudo nano /opt/lampp/etc/httpd.conf
```

2. Reemplazar:

```
DocumentRoot "/opt/lampp/htdocs"
<Directory "/opt/lampp/htdocs">
    ... y el resto hasta la etiqueta
</Directory>
```

Por:

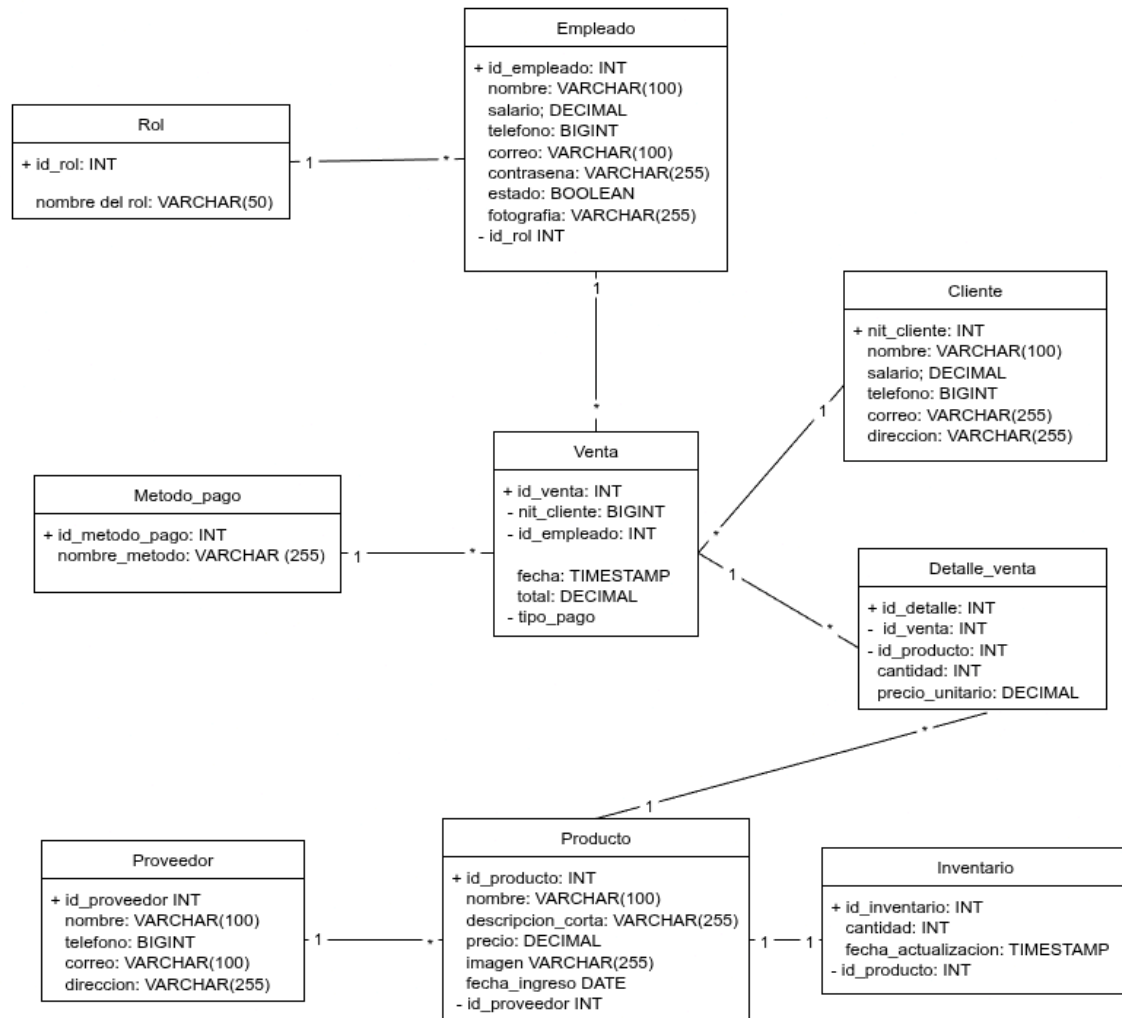
```
DocumentRoot "/home/jorge/Teoría_Sistemas"
<Directory "/home/jorge/Teoría_Sistemas">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

3. Dar permisos a la carpeta:

```
chmod -R 755 "/home/jorge/Teoría_Sistemas"
chown -R jorge:jorge "/home/jorge/Teoría_Sistemas"
```

DESCRIPCIÓN DE LA BASE DE DATOS

Diagrama Entidad - Relación



Descripción:

La base de datos implementa un sistema relacional para gestionar las operaciones principales de un supermercado, incluyendo empleados, clientes, inventario, proveedores, productos y ventas.

Estructura:

1. Gestión de Personal

- **rol**: define los distintos perfiles laborales (cajero, gerente, seguridad, etc.).
- **empleado**: almacena datos de cada trabajador, su información personal, credenciales y referencia a un rol.

2. Gestión de Productos e Inventario

- **proveedor:** contiene información de los proveedores.
- **producto:** define cada artículo con nombre, precio, descripción e imagen. Cada producto se asocia a un proveedor.
- **inventario:** controla el stock disponible y fecha de actualización. Está ligado 1:1 con cada producto.

3. Gestión de Clientes

- **cliente:** almacena información de clientes, identificados principalmente por su NIT (Número de Identificación Tributaria).

4. Gestión de Ventas

- **metodo_pago:** catálogo de formas de pago (efectivo, tarjeta, transferencia).
- **venta:** cabecera de la transacción, con fecha, total, cliente (opcional), empleado responsable y método de pago.
- **detalle_venta:** tabla intermedia que desglosa los productos incluidos en cada venta, con cantidad y precio unitario.

Módulos del Sistema

1. Login

El módulo de login es el encargado de gestionar el acceso de los empleados al sistema, asegurando que sólo usuarios registrados y activos puedan entrar a sus respectivos paneles según su rol.

Archivos principales

1. index.php

- Es la interfaz de login, donde el empleado ingresa su ID y contraseña.
- Se conecta a la base de datos Supermercado usando mysqli.
- Cuando el formulario se envía (POST):
 - Se prepara una consulta SQL con prepare y bind_param para evitar inyección SQL.
 - Se verifica que exista un empleado con el ID y contraseña ingresados y que esté activo (estado_activo = TRUE).
 - Si los datos son correctos, se crean variables de sesión (\$_SESSION) con el ID, nombre y rol del empleado.
 - Si los datos no coinciden, se muestra un mensaje de error.

- **Incluye el formulario HTML con campos:**

- ID empleado (number)
- Contraseña (password)
- Botón de Ingresar

2. panel.php

- Este archivo actúa como enrutador de sesión después del login.
- Verifica que haya una sesión iniciada (isset(\$_SESSION['id_empleado'])).
- Dependiendo del rol del empleado (id_rol):
 - Rol 3 → redirige al panel de administración general (admin_dashboard.php)
 - Rol 2 → redirige al panel de inventario (inventario_dashboard.php)
 - Otros roles → muestran mensaje de acceso no autorizado

Esto garantiza que cada usuario solo pueda acceder a la sección correspondiente a su rol.

Funcionamiento resumido

1. El empleado ingresa ID y contraseña en index.php.
2. El sistema valida credenciales en la base de datos.
3. Si son correctas, se inicializa la sesión y se guarda información básica del empleado.
4. panel.php redirige automáticamente al panel correspondiente según el rol.
5. Si los datos son incorrectos, se muestra un mensaje de error y no se permite el acceso.

Aspectos técnicos importantes

- **Seguridad:**
 - Se usa prepare y bind_param para evitar inyecciones SQL.
 - Solo se permite el acceso a empleados activos.
- **Sesiones:**
 - session_start() es obligatorio para mantener la sesión de usuario.
 - Las variables de sesión (\$_SESSION['id_empleado'], \$_SESSION['nombre'], \$_SESSION['id_rol']) permiten controlar el acceso en otros módulos.

- **Redirección:**
 - Se redirige automáticamente según el rol usando header("Location: ...").
- **Frontend:**
 - Formulario HTML sencillo y limpio, que puede heredar estilos desde estilos.css.

2. Administrador

Este módulo permite a los administradores (rol = 3) gestionar todos los empleados del sistema. Se compone principalmente de admin_dashboard.php y editar_empleado.php.

admin_dashboard.php

Funcionalidad principal: panel de control para crear nuevos empleados y listar los existentes.

- Verificación de acceso:
 - Se inicia sesión con session_start() y se revisa que el usuario tenga id_rol == 3.
 - Si no, se redirige a index.php.
- Conexión a la base de datos:
 - Se conecta a MySQL usando mysqli.
- Crear nuevo empleado:
 - Se captura la información del formulario: nombre, salario, rol, teléfono, correo y contraseña.
 - Si se sube una fotografía, se guarda en ../uploads/Trabajadores/.
 - Se inserta en la tabla empleado usando prepare y bind_param para evitar inyecciones SQL.
 - estado_activo se establece por defecto como TRUE.

- Listar empleados existentes:
 - Se hace una consulta para traer todos los empleados junto con su rol (JOIN rol).
 - Se muestran en una tabla HTML con todas las columnas necesarias, incluyendo miniaturas de la fotografía.
 - Cada fila incluye un botón “Editar” que redirige a `editar_empleado.php?id=ID_EMPLEADO`.

editar_empleado.php

Funcionalidad principal: permitir modificar los datos de un empleado existente.

- Verificación de acceso: igual que `admin_dashboard.php`.
- Obtener empleado a editar:
 - Se recibe el parámetro `id` por GET.
 - Se obtiene toda la información del empleado de la base de datos (`SELECT * FROM empleado WHERE id_empleado = ?`).
- Actualizar empleado:
 - Captura de formulario: nombre, salario, rol, teléfono, correo, contraseña y estado activo.
 - Fotografía:
 - Si se sube nueva foto, reemplaza la anterior en `../uploads/Trabajadores/`.
 - Si no, mantiene la foto existente.
 - Se ejecuta `UPDATE empleado` usando `prepare` y `bind_param` para seguridad.
 - Después de actualizar, se redirige de vuelta a `admin_dashboard.php`.
- Formulario de edición:
 - Campos prellenados con la información actual del empleado.
 - Checkbox para activar/desactivar empleado.
 - Visualización de la fotografía actual en miniatura.

Aspectos técnicos importantes

- Seguridad:
 - Validación de sesión y rol para limitar acceso solo a administradores.
 - Uso de prepare y bind_param en todas las consultas que modifican la DB.
- Manejo de imágenes:
 - Las fotos se guardan en carpetas separadas (uploads/Trabajadores/).
 - Se valida que el archivo exista antes de moverlo al servidor.
- Experiencia de usuario:
 - Formularios tipo grid para mejor organización.
 - Listado de empleados con datos completos y miniaturas.
 - Acciones de edición claras y directas.

3. Inventario

Este módulo está destinado a los empleados con rol de inventario (rol = 2) y permite gestionar productos, proveedores y el inventario de la tienda.

inventari_dashboard.php

Funcionalidad principal: panel de resumen y acceso rápido a las secciones de inventario.

- Verificación de acceso:
 - Se inicia sesión con session_start() y se valida que el usuario tenga id_rol == 2.
 - Redirige a index.php si no cumple.
- Conexión a la base de datos:
 - Se realiza con mysqli.
- Resumen del inventario:
 - Total de proveedores: SELECT COUNT(*) FROM proveedor.
 - Total de productos: SELECT COUNT(*) FROM producto.
 - Total de unidades en inventario: SELECT SUM(cantidad) FROM inventario.
 - Productos con stock bajo: SELECT COUNT(*) FROM inventario WHERE cantidad < 5.
- Tablas recientes:
 - Proveedores recientes: muestra los últimos 5 proveedores registrados.

- Productos recientes: muestra los últimos 5 productos con su proveedor y cantidad.
- Enlaces rápidos:
 - Accesos directos a:
 - proveedores.php
 - productos.php
 - inventario.php
- Interfaz:
 - Uso de cards para resúmenes.
 - Tabla clara para proveedores y productos recientes.
 - Stock bajo destacado con color de alerta.

inventario.php

Funcionalidad principal: gestión directa de cantidades de inventario de cada producto.

- Actualizar cantidad:
 - Se envía un formulario POST con id_producto y cantidad.
 - Se ejecuta UPDATE inventario SET cantidad = ? WHERE id_producto = ?.
- Listado de inventario:
 - Se obtiene la información de la tabla inventario junto con el nombre del producto (JOIN producto).
 - Se muestra en tabla con columnas:
 - ID Producto
 - Nombre
 - Cantidad
 - Última actualización
 - Acciones (formulario para actualizar cantidad)
- Interfaz:
 - Cada fila tiene un pequeño formulario para actualizar la cantidad directamente.
 - Validación mínima: cantidad ≥ 0 .

productos.php

Funcionalidad principal: agregar y editar productos.

- Crear producto:
 - Captura: nombre, descripción, precio y proveedor.
 - Imagen opcional, guardada en ../uploads/Productos/.
 - Inserta en tabla producto.
 - Crea registro inicial en inventario (cantidad = 0).
- Editar producto:
 - Permite modificar todos los campos, incluida la imagen.
 - Si no se sube imagen nueva, mantiene la actual.
 - Actualiza la tabla producto con UPDATE.
- Listar productos:
 - Se obtiene información de todos los productos con su proveedor.
 - Tabla incluye miniatura de imagen si existe.
 - Cada fila tiene un formulario de edición rápido: nombre, precio, descripción, proveedor e imagen.
- Interfaz:
 - Formulario de creación tipo grid.
 - Formularios de edición dentro de la tabla para modificar directamente.

proveedores.php

Funcionalidad principal: agregar y listar proveedores.

- Agregar proveedor:
 - Captura: nombre, teléfono, correo y dirección.
 - Inserta en la tabla proveedor.
- Listar proveedores:
 - Muestra todos los proveedores con columnas:
 - ID
 - Nombre
 - Teléfono
 - Correo
 - Dirección

- Interfaz:
 - Formulario simple para agregar.
 - Tabla clara para visualizar todos los proveedores existentes.

Aspectos técnicos importantes

- Seguridad:
 - Validación de sesión y rol (`id_rol == 2`) para todas las páginas.
 - Uso de `prepare` y `bind_param` para operaciones críticas (INSERT y UPDATE).