



Universidad de la Cuenca del Plata

Autorización Definitiva. Decreto Poder Ejecutivo Nacional N°091/2006

Trabajo Integrador

CARRERA: Ingeniería en Sistemas de Información

MATERIA: Paradigmas y Lenguajes de Programación 3

COMISIÓN: “U” (única) “A”

PROFESOR: Encina Agustín

ESTUDIANTE: Solonezen Brian Ezequiel

FECHA: 18-11-2025

Tabla de contenido

Introducción.....	2
Desarrollo.....	3
1. Análisis	3
Objetivo del sitio.....	3
Requisitos Funcionales.....	3
Requisitos No Funcionales	3
Público objetivo	4
Estructura de páginas	4
Navegación	4
2. Diseño	5
Diseño visual	5
Diseño CSS recomendado	5



Introducción

En este trabajo se hablará del Análisis y Diseño de una página web, aplicando diferentes tecnologías como HTML y CSS. También aplicaremos otras tecnologías en el backend como PHP y MySQL para la base de datos.

En este caso se pide una página principal (index.html), y secciones como listado de productos en tablas, en box, una ficha de un producto en particular y un formulario de compra.

Para este segundo trabajo, agregamos funcionalidades con JavaScript. Incorporamos un sistema de compras, cada compra se guardará en la base de datos. También el registro de usuarios y los productos que agregue el admin.



Desarrollo

1. Análisis

Para el desarrollo de la web apliqué varias animaciones con css para darle un poco más de movimientos y no sea tan estático. En el head agregue html responsive para que no haya problemas cuando la veas en diferentes dispositivos.

Sobre el diseño cada archivo html está enlazado con su respectivo archivo css desde una carpeta llamada hojas, las imágenes tienen su propia carpeta y sobre el video los deje allí. No estoy seguro si dejar ese video o buscar otro más adecuado.

Para un futuro o una entrega desearía poder aplicar o tratar de colocar una api para poder pagar con tarjeta o transferencia, un ejemplo seria Mercado Pago.

Objetivo del sitio

- Vender suplementos nutricionales.
- Mostrar productos en diferentes formatos (tabla y cajas).
- Proveer una ficha individual por producto.
- Permitir completar un formulario de compra sin necesidad de backend.

Requisitos Funcionales

1. Mostrar la portada con presentación de la tienda.
2. Listar los productos en **formato tabla** y **formato box**.
3. Mostrar **ficha detallada de producto** (imagen, nombre, precio, descripción).
4. Formulario de compra con los campos:
 - Nombre del cliente
 - Dirección
 - Teléfono
 - E-mail
 - Medio de pago
 - Listado de productos seleccionados
5. Navegación entre páginas mediante enlaces.

Requisitos No Funcionales

- **Tecnología:** HTML y CSS puros (sin backend ni JavaScript).
- **Compatibilidad:** Visualización correcta en móviles y escritorio.
- **Accesibilidad:** Uso de etiquetas semánticas y atributos alt en imágenes.
- **Estética:** Colores suaves, animaciones sutiles y diseño responsivo.



Público objetivo

- Personas interesadas en salud y fitness.
- Rango de edad: 18 a 50 años.
- Clientes con conocimientos básicos de compra online.

Estructura de páginas

- **index.html:** Portada principal (con logo, slogan y video centrado).
- **listado_tabla.html:** Lista de productos en formato tabla.
- **listado_box.html:** Lista de productos en formato cajas con imágenes.
- **producto.html:** Detalle de un producto.
- **comprar.html:** Formulario de compra.

Navegación

- Menú principal fijo o en la parte superior con enlaces:
 - Inicio
 - Ver Productos (box)
 - Ver Productos en Tabla
 - Comprar



2. Diseño

Para el desarrollo de las páginas o “secciones” decidí poner los mismos colores que el logo de la página o parecidos, el logo fue creado con inteligencia artificial, al igual que las imágenes de los productos. Para una segunda entrega agregaría un <nav> con las secciones. El video decidí dejarlo por ahora en ese lugar pero mi idea es ponerlo al costado de la lista y hacerlo más ordenado, a lo que me refiero es que todos los elementos no estén centrados o uno debajo del otro. Así no se ve tan vacío en los costados. Tal vez agregar una etiqueta <aside> para que sea más semántico. Poder tener un carrito de compras en donde visualizar los productos seleccionados. Sobre el tema del logo todavía no decidí si dejarlo ahí o ponerla en un costado, con un tamaño menor. Agregar un menú desplegable o una barra de búsqueda. La web tiene css, y algunas propiedades de animaciones para que sea mas moderno.

Para hacerlo más funcional y un poco más ordenado, debería hacer la página más semántico y dejar de usar un poco la etiqueta <div>.

Diseño visual

- **Color principal:** #D2B48C (beige tostado).
- **Color de acento:** blanco para texto sobre fondos oscuros.
- **Tipografía:** Sans-serif limpia (Arial, Helvetica o similar).
- **Estilo de botones:** Bordes redondeados, efecto hover con sombra.
- **Imágenes:** Redondeadas o cuadradas con sombra suave.
- **Video:** Centrado y con borde sutil.

Diseño CSS recomendado

- **Flexbox** para centrar contenido.
- **Grid** para el listado de productos (box).
- **Transiciones suaves** en botones.
- **Responsividad con media queries.**



Implementaciones con JavaScript

En la implementación del formulario de inicio de sesión se utilizó el lenguaje **JavaScript** con el objetivo de realizar **validaciones del lado del cliente** antes de enviar la información al servidor. Esta práctica mejora la **experiencia del usuario y optimiza el rendimiento** de la aplicación al evitar que se envíen datos incorrectos o incompletos.

Mediante un pequeño script en JavaScript se comprueba que los campos obligatorios (correo electrónico y contraseña) cumplan con ciertas condiciones:

- Que el **correo electrónico** tenga un formato válido (por ejemplo, que contenga un "@" y un dominio correcto).
- Que la **contraseña** no esté vacía y tenga una longitud mínima de 6 caracteres.

Estas validaciones se ejecutan en el navegador al momento de presionar el botón "Ingresar". Si alguno de los campos no cumple los requisitos, el formulario no se envía y se muestra un **mensaje de error** debajo de los campos, sin necesidad de recargar la página.

El uso de JavaScript en este contexto permite:

- Detectar errores de forma inmediata.
- Reducir solicitudes innecesarias al servidor.
- Aumentar la seguridad básica del formulario (evitando envíos vacíos o mal formateados).
- Brindar una interacción más dinámica y fluida al usuario.

En conclusión, el uso de JavaScript en el formulario de inicio de sesión cumple un rol esencial en la **validación y control de datos en el lado del cliente**, garantizando que la información enviada al servidor sea consistente y correcta.

Formulario con JavaScript

En el formulario de registro de usuario se implementó **JavaScript** para realizar validaciones dinámicas del lado del cliente, asegurando que la información ingresada por el usuario sea correcta antes de enviarse al servidor. Esta técnica mejora tanto la **usabilidad** como la **seguridad básica** del sistema.

El código de validación en JavaScript verifica distintos aspectos del formulario:

- Que todos los campos obligatorios (nombre, correo, usuario, contraseña y confirmación) estén completos.
- Que el **nombre** tenga al menos tres caracteres.
- Que el **correo electrónico** tenga un formato válido (por ejemplo, "usuario@dominio.com").
- Que el **usuario** y la **contraseña** cumplan una longitud mínima para mayor consistencia.
- Que la **confirmación de contraseña** coincida con la original.

En caso de que algún dato sea incorrecto o falte información, se muestra un **mensaje de error** debajo del formulario, evitando el envío de datos incompletos al servidor.

El uso de JavaScript en este contexto permite:



- **Prevenir errores comunes** antes de enviar la información.
- **Reducir la carga del servidor**, ya que no se procesan formularios inválidos.
- **Brindar retroalimentación inmediata** al usuario, mejorando la experiencia de uso.

conexion.php: es el archivo en el cual hacemos la conexión a la base de datos

La base de datos está compuesta por tres tablas, la tabla usuarios en donde podemos obtener su información importante, la tabla productos y una tabla compras que es en donde se van a guardar las solicitudes de compras.

Tablas:

index.php: La pagina principal cuenta con un header, en donde estan las opciones para iniciar sesión ó registrarse, el usuario admin al ingresar la interfaz va a ser diferente, ya que en el header van a estar la opcion del panel admin, y además de eso aparece la opcion de cerrar sesion, para el admin o usuario. Dentro del index lo que es el body y la sección **main, hay tres opciones:** comprar(ingresa al formulario de compra), Ver productos(donde están los productos como un listado box) y Ver productos por tabla en donde se puede ver todos los productos sin imagen.

listado_box.php: es el archivo en donde están los productos, estos productos fueron agregados anteriormente por el admin desde su crud. Los productos nos permite agregar a un carrito en donde tiene tecnologías como Javascript, en ese carrito nos podemos permitir eliminar productos que no queramos o poder vaciar el carrito.

comprar.php: es un formulario en donde vamos a completar con nuestros datos y nos va a mostrar, los productos que estaban en el carrito. Al finalizar la compra , nos saldrá un mensaje en la página que la compra fue exitosa (es obligatorio completar todos los campos para completar la compra).

Interfaz del Admin

crear_admin.php: archivo para crear al usuario admin y asi ingresar al panel de administración.

admin.php: El admin al ingresar, automáticamente se le mostrará el panel admin, en donde puede agregar productos con descripción, precio, imagen y su stock, también puede actualizar y eliminar productos.

Todos estos cambios, aparecerán en la tabla productos y se agregarán al listado_box.php sin problema. Todos los cambios como actualizar, agregar y eliminar. Llaman a un archivo php en concreto, en estos casos son tres. agregar_producto.php, eliminar_producto.php y actualizar_producto.php.

Estos cambios se ven reflejados también en el listado de productos por tabla.



Objetivos del Sistema

Los principales objetivos funcionales y técnicos de este desarrollo son:

1. **Conectividad y Persistencia** : Establecer una conexión segura y reutilizable a una base de datos MySQL llamada [nutricion](#)([conexion.php](#)).
2. **Visualización de Productos** : Permitir a los usuarios visualizar el catálogo de productos en dos formatos: una cuadrícula ([listado_box.php](#)) con descripciones y una vista en tabla simple ([listado_tabla.php](#)).
3. **Flujo de Compra No Registrada** : Implementar un **Carrito de Compras** basado en el lado del cliente (utilizando `localStorage` , gestionado por [carrito.js](#)) y un formulario de finalización de compra ([comprar.php](#)).
4. **Gestión de Transacciones** : Procesar la compra, lo que implica registrar la orden en la tabla [compras](#), **actualizar el stock de los productos** y enviar una confirmación por correo electrónico ([comprar.php](#)).
5. **Control de Acceso y Roles** : Autenticar usuarios y restringir el acceso a funcionalidades administrativas, como la gestión de productos, solo a usuarios con el rol "admin" ([index.php](#), [admin.php](#)).

Administración de Contenido (CRUD) : Proveer una interfaz administrativa para **eliminar productos** ([eliminar_producto.php](#)) y editar su precio, stock y descripción ([admin.php](#), [actualizar_producto.php](#)).

En conclusión, el empleo de JavaScript en el formulario de registro asegura que los datos ingresados sean coherentes, completos y válidos, fortaleciendo la calidad de la información almacenada y aportando un comportamiento más interactivo e intuitivo a la aplicación web. Mientras que para las tecnologías del backend nos ayuda a entender como las aplicaciones web de cualquier servicio, están hechas con estas tecnologías.



Conclusión

El uso de javascript nos permite agregar mas funcionalidades a la pagina web, además de eso nos permite tener validaciones en el frontend, esto nos ofrece mejor información y sin perdida de ella. Un buen uso de css para mejorar el diseño de la pagina y aplicar el método responsivo para mantener la web en orden y sin pérdida de espacios.

Repositorio GIT: https://github.com/SalonezenB/Integrador_Paradigma

