



Entorno Servidor
Carlos Tarriño Espinosa

Trabajo de Entorno Servidor (DDP)





1. Control de versiones

El siguiente enlace te redireccionará hacia el control de versiones de mi repositorio de GitHub:

<https://github.com/Tares98/Trabajo-entorno-servidor/commits/main/>

2. Índice

DOCUMENTO DE DEFINICIÓN DE PROYECTO

1. Índice	2
2. Contenido principal	3
Introducción	3
Requerimientos funcionales	3
Diseño de la solución 3 capas	4
Diagrama de la capa front-end	5
Diagrama de los principales flujos de navegación	6
Diagrama de la capa de la lógica de negocio	2
Diseño del diagrama E-R.....	2
3. Datos de ejecución	3



3. Contenido principal

Introducción

Sport's Moment es una empresa dedicada al mundo del fitness. Fue fundada en 2024 y se ha establecido como una tienda de confianza para productos relacionados con el deporte y el ejercicio. Se compromete a ofrecer artículos de alta calidad para ayudar a sus clientes a alcanzar sus objetivos deportivos, sin importar el nivel del atleta. La compañía ofrece una amplia gama de ropa deportiva, accesorios avanzados y equipamiento de alto rendimiento. Sport's Moment concibe el fitness como un estilo de vida y busca ser un aliado en el viaje hacia un cuerpo más fuerte y una mente más saludable, invitando a todos a unirse a su aventura.

Requerimientos funcionales

- **RF01-Navegación por la página web:** implementación de un sistema de navegación para que el usuario pueda desplazarse por las diferentes páginas que componen la página web
- **RF02-Registro de usuario:** elaboración un formulario para que los usuarios puedan registrarse
- **RF03-Acceso de usuarios:** elaboración un formulario para que los usuarios registrados puedan acceder
- **RF04-Control de Acceso de usuarios:** creación de un sistema de control de diferentes perfiles de usuarios
- **RF05-Control de sesiones:** creación de un sistema que controle las acciones de cada usuario
- **RF06-Alta/Baja/Modificación de usuarios:** implementación de un sistema de gestión de usuarios para los administradores
- **RF07-Alta/Baja/Modificación de productos:** implementación de un sistema de gestión de productos para los administradores
- **RF08-Incorporación de productos al carrito de compras:** elaboración del método por el que los clientes añaden los productos que desean comprar a su carrito de compra



- **RF09-Gestión del carrito de compras:** elaboración del método por el que los clientes interactúan con su carrito, pudiendo eliminar productos de este o editar la cantidad seleccionada
- **RF10-Pago del carrito:** creación de confirmación de compra una vez que se realiza el pago del carrito, mostrando un resumen de los productos adquiridos
- **RF11-Consulta de historial de pedidos:** implementación de página que recolecte los pedidos realizados por el usuario y se los muestre

Diseño de la solución 3 capas

1. Capa de Presentación (Front-End):

- **Descripción:** Esta capa es lo que los usuarios ven y con lo que interactúan. Incluye la interfaz de usuario (UI) y la experiencia de usuario (UX). Representaría el diseño de la tienda, los catálogos de productos, las opciones de búsqueda, carrito de compras, etc.
- **Tecnologías Utilizadas:** HTML, CSS y JavaScript.
- **Interacción con Otras Capas:** Esta capa envía y recibe datos de la capa de lógica de negocio, como solicitudes de información de productos, gestión de carritos de compras, procesamiento de pagos, etc.

2. Capa de Lógica de Negocio (Back-End):

- **Descripción:** Esta capa gestiona la lógica de negocio, el control de sesiones, el procesamiento de datos de productos, la lógica del carrito de compras, etc.
- **Tecnologías Utilizadas:** PHP.
- **Interacción con Otras Capas:** Procesa las solicitudes de la capa de presentación e interactúa con la capa de acceso a datos para recuperar o almacenar información.

3. Capa de Acceso a Datos (Base de Datos):

- **Descripción:** Esta capa se encarga del almacenamiento y recuperación de datos, como información de usuarios, datos de productos, datos del carrito y el historial de pedidos.
- **Tecnologías Utilizadas:** MySQL y MySQLi

- **Interacción con Otras Capas:** Esta capa recibe solicitudes de la capa de lógica de negocio para leer o escribir datos en la base de datos y le devuelve estos datos cuando lo solicita.

4. Aspectos Importantes a Destacar:

- **Seguridad:**
 - Para las contraseñas he usado "password_hash"
 - He usado el método POST para todos los formularios
- **Escalabilidad y Mantenimiento:** He implementado un sistema modular separando las funcionalidades en diferentes scripts.

Diagrama de la capa front-end

Debido a la incorporación de un mapa de navegación en la cabecera de cada página se podrá acceder en cualquier momento a todas las secciones de la página web. A excepción del botón de salida, que solo se podrá acceder si estamos logeados, las páginas de control, a las que solo podrán acceder usuarios administradores, y desde donde podremos acceder a las páginas de edición y creación.

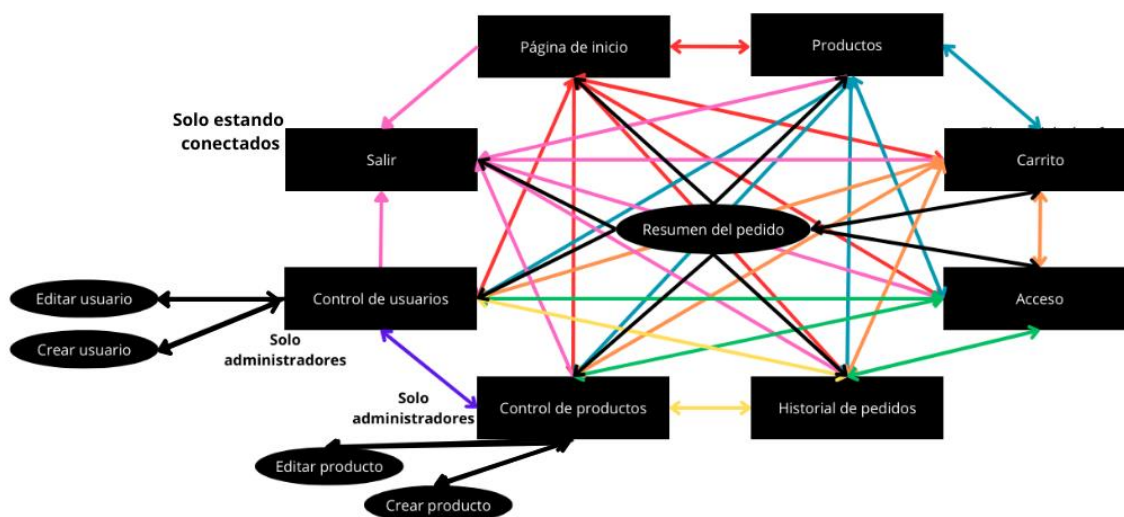
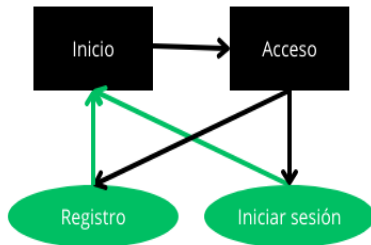


Diagrama de los principales flujos de navegación

**Diagrama para acceso o registro
(Todos los usuarios):**



**Diagrama para compra de
productos y revisión de historial
(Todos los usuarios conectados):**

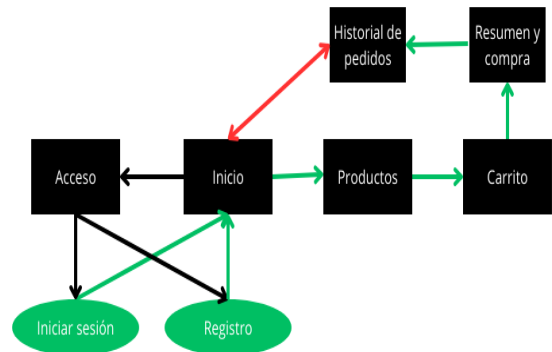


Diagrama para gestión de productos y usuarios (Administradores):

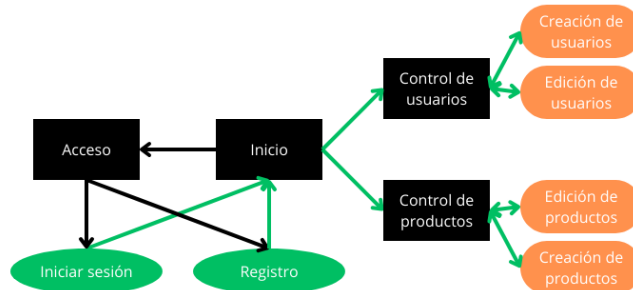
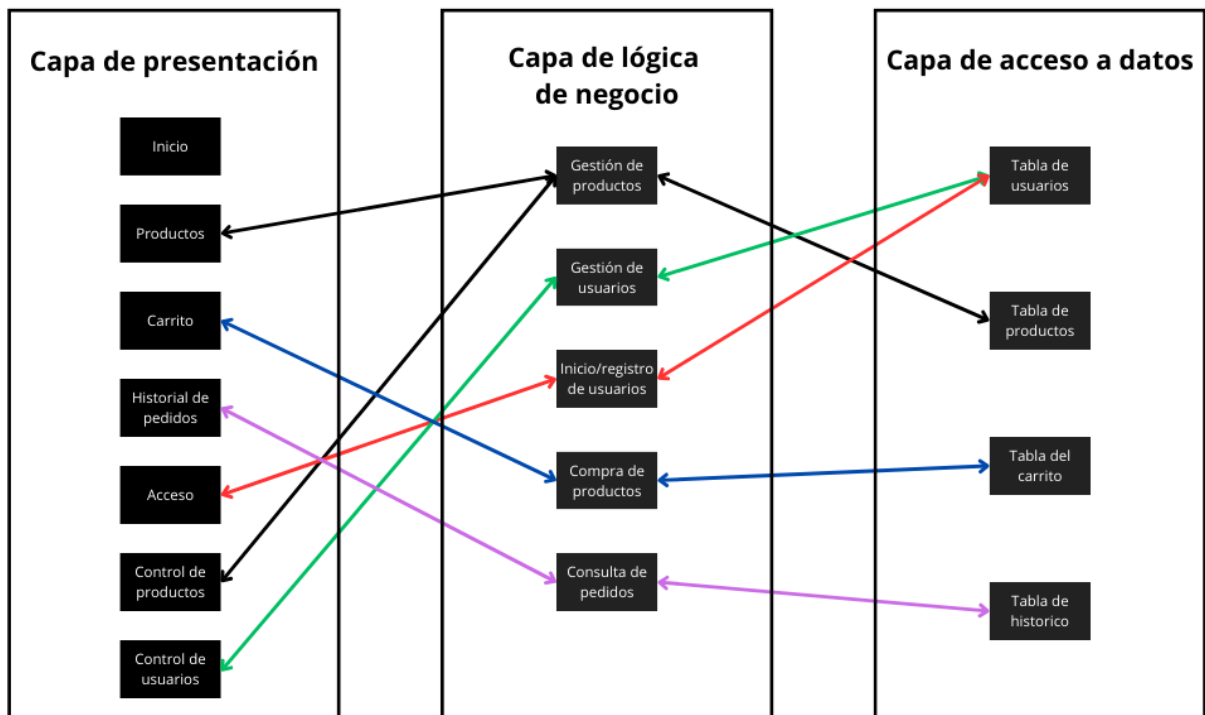
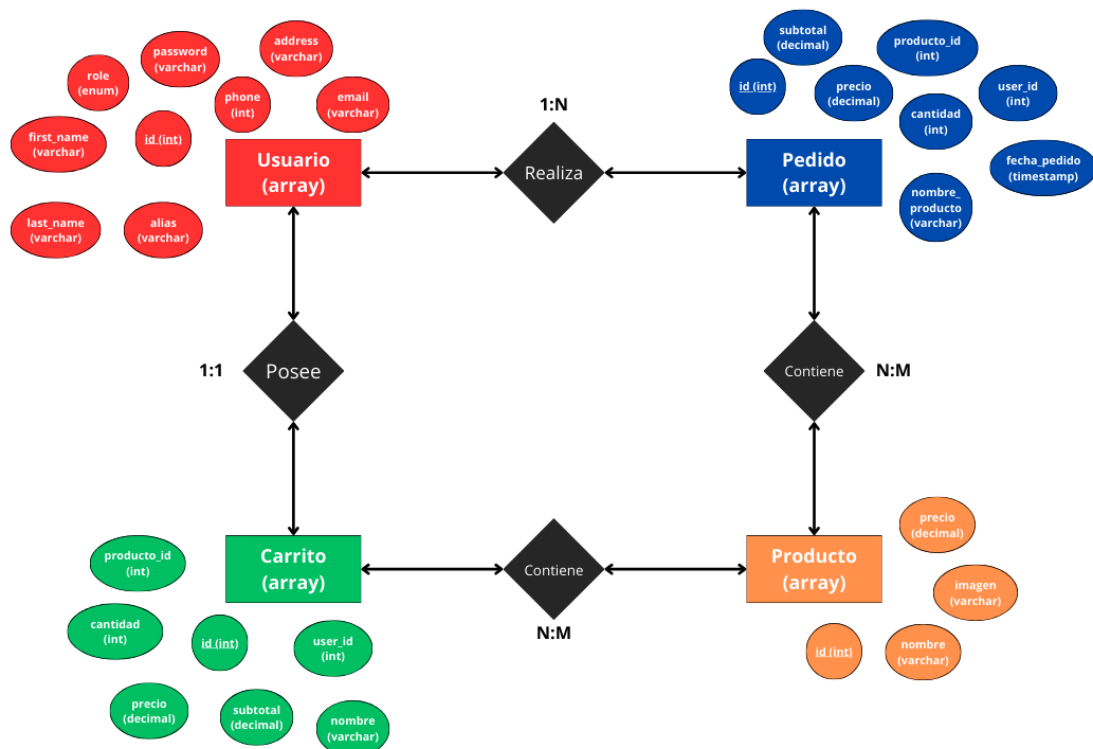




Diagrama de la capa de la lógica de negocio



Diseño del diagrama E-R





4. Problemas

Aunque no ha sido un trabajo fácil, he conseguido solventar todos los baches y contratiempos que han ido apareciendo. Quizás podría haber mejorado un poco más la organización tanto de los archivos .html como de los .php pero cada vez que lo intentaba rompía las rutas estropeando así las funcionalidades de la aplicación por lo que acabé desistiendo.

Mi único gran inconveniente ha sido a la hora de alojar mi aplicación web en un servidor ya que cuando accedí a la página "swhosting.com" no encontré ningún plan totalmente gratuito y como no era obligatorio, opté por no realizarlo.

5. Datos de ejecución

Como comentaba, no lo aloje en ningún servidor web