



Universidad
Isabel I



Memoria Prácticas Externas

Curso 2024-2025

Alumno Pablo Enrique Guntín Garrido

Asignatura Prácticas Externas. Cuarto Curso (trimestres 11-12)

Grado en Ingeniería Informática

curso 2024-2025



Índice

1. Introducción	3
2. Descripción del centro colaborador	4
3. Descripción y desarrollo de las prácticas.....	5
3.1 Descripción detallada de tareas y problemas planteados	5
3.2 Objetivos	6
3.2.1 Objetivo general	6
3.2.2 Objetivos específicos	7
3.3 Recursos empleados	8
3.3.1 Recursos tecnológicos.....	8
3.3.2 Recursos estructurales y arquitectónicos.....	9
3.4 Procedimientos de resolución y dificultades encontradas	10
3.5 Resultados	10
4. Conclusiones.....	11
5. Bibliografía	11

1. Introducción

MEMORIA DE PRÁCTICAS EXTERNAS

CURSO 2024/25

Alumno/a:

PABLO ENRIQUE GUNTÍN GARRIDO

Tutor/a externo:

SERGIO LASTRA POZO

Tutor/a académico:

MARIO SOLANA EZQUERRA

GRADO EN INGENIERÍA INFORMÁTICA

FACULTAD DE CIENCIAS Y TECNOLOGÍA

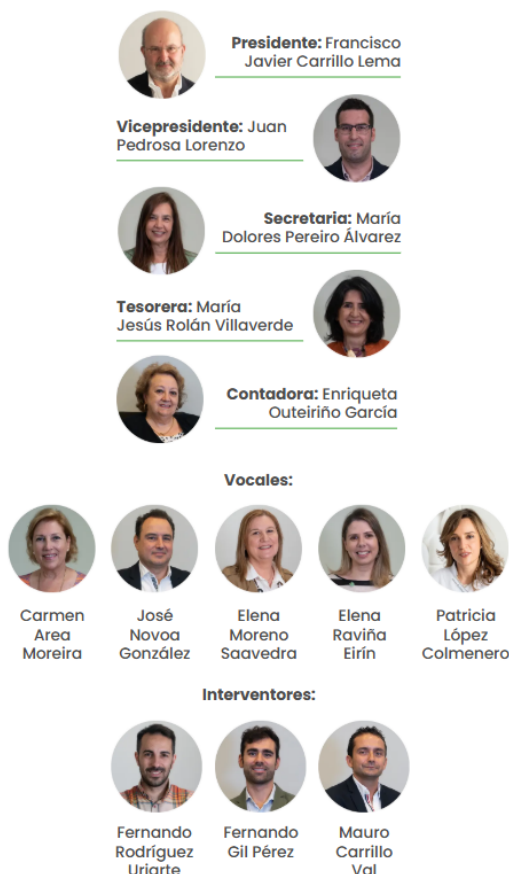
UNIVERSIDAD ISABEL I

2. Descripción del centro colaborador

El centro donde he realizado las prácticas ha sido COFANO. Su sede principal se encuentra en Vigo, pero con almacenes y oficinas en Ourense, Pontevedra y A Coruña. Yo he realizado las prácticas entre Vigo y Ourense, donde yo resido.

El organigrama del centro está dividido en Organigrama Consejo Rector y Organigrama Equipo Directivo.

Organigrama Consejo Rector



Organigrama Equipo Directivo



Imagen 1. Organigrama. Fuente: [COFANO](#)

COFANO es una cooperativa de distribución farmacéutica y de productos sanitarios que opera en Galicia y el noroeste de la península.

Su principal función es la distribución de medicamentos y productos sanitarios a farmacias pertenecientes a la cooperativa, aparte de prestar servicios de gestión y apoyo a las farmacias, como puede ser de consultoría, logística o de software.

La práctica la he realizado entre Vigo y Ourense, especialmente desde las oficinas de Ourense.

El trabajo realizado ha servido para atender a la los empleados que se encargan de suministrar a las farmacias.

3. Descripción y desarrollo de las prácticas

3.1 Descripción detallada de tareas y problemas planteados

Durante el período de prácticas curriculares tuve la oportunidad de desarrollar una aplicación web real destinada a los transportistas de la cooperativa farmacéutica COFANO. El objetivo de dicha aplicación era facilitar la interacción entre los trabajadores de logística y la base de datos del sistema, permitiendo consultar y actualizar información relativa al inventario de productos. La aplicación se diseñó siguiendo el patrón arquitectónico Modelo-Vista-Controlador (MVC), utilizando PHP como lenguaje de desarrollo principal y procedimientos almacenados para realizar las operaciones de lectura y escritura sobre la base de datos.

La primera fase del desarrollo consistió en la verificación y validación de la conexión entre la aplicación y la base de datos corporativa. Para ello, trabajé con una clase `conexion.php`, la cual implementaba la conexión mediante PDO y ODBC, incluyendo manejo de excepciones (try-catch) y el uso del patrón Singleton, para asegurar que sólo existiera una conexión activa durante la sesión del usuario. Esta parte fue especialmente crítica, ya que errores en la cadena de conexión o en la configuración del driver podían bloquear el acceso al sistema completo.

Una vez lograda y garantizada la conectividad, me centré en implementar el sistema de autenticación. Esta parte incluía la construcción de un formulario de login (`loginView.php`) y su correspondiente lógica en el controlador `LoginController.php`. Para validar las credenciales del usuario, utilicé un procedimiento almacenado llamado `PAVALIDAUSR`, al que se le pasaban parámetros de entrada, usuario y contraseña, y se leía su respuesta mediante parámetros de salida, utilizando `bindParam` con `PDO::PARAM_INPUT_OUTPUT`. Si la autenticación era correcta, el usuario era redirigido a un panel principal (`dashboardView.php`).

Dicho panel principal contaba con un menú desplegable con cuatro funcionalidades previstas y un botón para cerrar sesión. Además, la pantalla incluía un botón de inicio y un panel de control (body) que serviría como contenedor dinámico de los diferentes módulos o vistas que conformaban las funcionalidades de la aplicación. Por cuestiones de tiempo, solo una de esas funcionalidades pudo desarrollarse completamente durante el período de prácticas: la funcionalidad de consulta y modificación de inventario. Aunque para lograrlo tuve que dedicarle más horas de las debidas.

El módulo de "lista de inventario" fue el más extenso y desafiante de toda la aplicación. Su diseño incluía los siguientes pasos:

1. Mostrar un formulario de filtrado con dos selectores (tipo de producto y estado) obtenidos desde la base de datos mediante el procedimiento `PASELTXINV`, con códigos 1 y 3.
2. Enviar la selección al backend para recuperar el inventario filtrado desde la base de datos, mediante `PASELLSTINV`.
3. Mostrar los resultados en una tabla paginada con la ayuda de JavaScript (`paginacion_inventario.js`).
4. Permitir al usuario seleccionar una fila de inventario, lo que redirigía a una nueva vista de detalle (`DetalleInventarioView.php`).

5. En esta vista, se mostraban los productos dentro del inventario seleccionado. Cada producto incluía campos editables como unidades reales (uds) y fecha de caducidad (caducidad).
6. Cuando el usuario modificaba uno de estos campos, se activaba un sistema de semáforo visual que indicaba que la fila estaba pendiente de guardar.
7. Los cambios podían ser guardados de forma individual (botón "Guardar") o de forma masiva (botón "Procesar cambios"), ambos gestionados mediante llamadas AJAX a `DetalleInventarioController.php`, que invocaba el procedimiento `PAUPDINV`.

Los principales problemas planteados durante el desarrollo de esta funcionalidad fueron:

- Dificultades en la interpretación de los procedimientos almacenados existentes: fue necesario comprender bien su lógica, entradas y salidas, especialmente al no disponer de documentación detallada.
- La implementación de la paginación en las tablas HTML con JavaScript, asegurando que el sistema fuera eficiente y fácil de usar.
- Gestión de errores silenciosos al invocar procedimientos almacenados desde PHP: fue necesario depurar paso a paso para detectar qué parámetros eran incorrectos o por qué no se devolvían resultados.
- Diseño e integración del sistema de semáforos con clases dinámicas en JavaScript para detectar y visualizar qué filas habían sido modificadas por el usuario.

A pesar de que solo se desarrollé una de las cuatro funcionalidades previstas, esta abarcó todo el flujo de la arquitectura MVC y permitió trabajar con conexión a base de datos, validación, visualización, edición y sincronización en tiempo real. Para ello, he trabajado varias veces con la arquitectura MVC, ya que antes de desarrollar la funcionalidad, he realizado otras funcionalidades mucho más sencillas a modo prueba para saber cómo realmente se utilizaban los procedimientos, cómo extraer los datos, cómo mostrarlos, cómo seleccionarlos, etc.

3.2 Objetivos

El desarrollo de la aplicación web para los transportistas de COFANO estuvo guiado por una serie de objetivos tanto generales como específicos. Estos objetivos surgieron de la necesidad real de la cooperativa de contar con una herramienta tecnológica que facilitase la gestión y actualización del inventario por parte del personal logístico, especialmente aquellos responsables del control y seguimiento de productos en tránsito.

3.2.1 Objetivo general

Diseñar e implementar una aplicación web funcional, segura y escalable que permita a los transportistas de COFANO consultar y modificar los datos del inventario en tiempo real, a través de una interfaz amigable y conectada directamente a la base de datos corporativa mediante procedimientos almacenados.

Este objetivo no solo responde a una necesidad operativa concreta de la organización, sino que también permite consolidar en un entorno profesional los conocimientos adquiridos a lo largo de la formación académica.

3.2.2 Objetivos específicos

1. Garantizar la conectividad fiable con la base de datos de COFANO mediante tecnologías robustas como PDO (PHP Data Objects) y ODBC (Open Database Connectivity), implementando patrones de diseño que aseguren la eficiencia y estabilidad de la conexión.
2. Diseñar una arquitectura lógica basada en el modelo MVC (Modelo-Vista-Controlador) que permita una separación clara entre las capas de datos, lógica y presentación, facilitando el mantenimiento, la escalabilidad y la reutilización del código fuente.
3. Implementar un sistema de autenticación de usuarios que permita el acceso controlado a la aplicación, validando credenciales mediante procedimientos almacenados preexistentes y gestionando sesiones de forma segura para preservar la privacidad y la integridad de los datos.
4. Desarrollar un panel principal interactivo, accesible tras la autenticación, que sirva como punto de entrada a distintas funcionalidades de la aplicación, organizado mediante un menú desplegable y con estructura modular para permitir la ampliación futura.
5. Crear e integrar la funcionalidad "lista de inventario" como primer módulo operativo de la aplicación, incluyendo:
 - Formulario de filtrado por tipo de producto y estado.
 - Visualización de inventarios en tablas dinámicas y paginadas.
 - Acceso al detalle del inventario seleccionado.
 - Edición de datos sensibles (fecha de caducidad y unidades reales).
 - Confirmación y sincronización de los cambios con la base de datos mediante procedimientos almacenados específicos.
6. Facilitar la edición eficiente y controlada de los datos de inventario, mediante una interfaz que detecte automáticamente los cambios realizados por el usuario, los marque visualmente y permita su envío individual o masivo al servidor, con confirmación inmediata de los resultados.
7. Aplicar buenas prácticas de programación y seguridad, como la validación de entradas del usuario, el control de errores, la separación de responsabilidades entre archivos y la protección de la lógica del servidor frente a accesos indebidos o maliciosos.
8. Permitir la extensibilidad de la aplicación mediante una estructura base capaz de admitir nuevas funcionalidades sin necesidad de rediseñar el sistema completo. Aunque solo se desarrolló una de las cuatro funcionalidades planificadas, el diseño del panel, los controladores y la lógica de vistas permite fácilmente añadir nuevos módulos en el futuro.
9. Contribuir a la mejora de los procesos internos de COFANO, proporcionando una herramienta tecnológica real que optimiza el flujo de trabajo de los transportistas, reduce la necesidad de intervención manual o papel y mejora la precisión de los datos registrados.
10. Consolidar los conocimientos adquiridos durante la formación académica, poniendo en práctica habilidades técnicas (programación, bases de datos, diseño de interfaces, depuración) y habilidades transversales (organización, resolución de problemas, comunicación técnica y trabajo por objetivos) en un entorno de trabajo profesional.

3.3 Recursos empleados

El desarrollo de la aplicación web requirió el uso de una combinación de recursos técnicos, metodológicos y organizativos, orientados a garantizar la funcionalidad, seguridad y mantenibilidad del sistema. A lo largo de las prácticas se emplearon distintas herramientas, tecnologías y estructuras de desarrollo profesional.

3.3.1 Recursos tecnológicos

1. Lenguaje de programación: PHP

PHP fue el lenguaje principal utilizado para desarrollar la lógica del servidor y la comunicación con la base de datos. Se empleó su versión 7, aprovechando características como el manejo de excepciones (try-catch), tipado flexible, clases y estructuras modernas. Además, se siguió una estructura basada en la arquitectura Modelo-Vista-Controlador, lo que permitió separar la lógica de negocio de la presentación y de la manipulación de datos.

2. Base de datos: Microsoft SQL Server

La base de datos empleada durante el proyecto es la que COFANO utiliza en su entorno real de trabajo: SQL Server, un sistema de gestión de bases de datos relacional de alto rendimiento y fiabilidad. El acceso a esta base de datos se realizó desde PHP mediante la interfaz ODBC y gestionada con PDO. Esto facilitó una integración eficiente y segura, y permitió la ejecución de procedimientos almacenados que centralizan la lógica de negocio en el servidor.

3. Procedimientos almacenados

Toda la lógica crítica relacionada con la validación de usuarios, consulta y edición de inventario se delegó a procedimientos almacenados previamente creados por COFANO. Algunos ejemplos relevantes incluyen:

- PAVALIDAUSR: valida usuarios y devuelve información sobre su autenticación.
- PASELTXTINV y PASELLSTINV: obtienen listas de tipos, estados y datos de inventario filtrados.
- PASELDETINV: obtiene el detalle de un inventario específico.
- PAUPDINV: actualiza los datos de inventario modificados por el usuario.

El uso de estos procedimientos permitió una integración con los sistemas corporativos reales y un control total sobre la consistencia de los datos.

4. HTML, CSS y JavaScript

Para el desarrollo del frontend (interfaz de usuario), se utilizó HTML5 para la estructura del contenido, CSS3 para los estilos visuales y JavaScript junto con jQuery para la interacción del usuario. Esto permitió diseñar vistas dinámicas y reactivas, con funcionalidades como:

- Tablas con paginación para grandes volúmenes de datos.
- Formularios con validación de campos.
- Actualización asincrónica de datos mediante AJAX.
- Un sistema visual de "semaforización" que marca los cambios realizados por el usuario y habilita la opción de guardar.

5. AJAX (Asynchronous JavaScript and XML)

Se utilizaron llamadas AJAX para enviar los cambios del inventario desde el cliente al servidor sin necesidad de recargar la página. Esto permitió que las actualizaciones fueran rápidas, discretas y cómodas para el usuario, mejorando la experiencia de uso.

6. Gestión de sesiones en PHP

Se emplearon sesiones PHP para mantener el estado del usuario autenticado durante la navegación en la aplicación. Se controló el acceso a las funcionalidades internas, impidiendo que usuarios no autenticados accedieran a rutas o vistas sin permiso.

7. Entorno de desarrollo local: XAMPP

Para desarrollar y probar la aplicación, se utilizó un entorno de desarrollo local configurado con un servidor web compatible con PHP, como lo es XAMPP.

3.3.2 Recursos estructurales y arquitectónicos

Arquitectura Modelo-Vista-Controlador (MVC)

La arquitectura del proyecto se estructuró bajo el patrón MVC, ampliamente utilizado en el desarrollo de aplicaciones web. Esta separación en tres capas facilitó:

- Una organización clara del código fuente.
- La reutilización de componentes y vistas.
- La posibilidad de mantener y ampliar el proyecto con menor esfuerzo.

Esta arquitectura permitió implementar los controladores que gestionaban las solicitudes, los modelos que se conectaban a la base de datos, y las vistas que presentaban la información al usuario.

Controladores (Controllers)

Los controladores fueron responsables de gestionar la lógica de cada módulo. Se desarrollaron controladores específicos para:

- Inicio y cierre de sesión (LoginController.php, LogoutController.php)
- Visualización y edición del inventario (InventarioController.php, DetalleInventarioController.php)

Modelos (Models)

Los modelos encapsularon el acceso a la base de datos y la lógica relacionada con los procedimientos almacenados. Cada controlador se comunicaba con su respectivo modelo para obtener y actualizar los datos necesarios.

Vistas (Views)

Las vistas presentaban al usuario la información visual. Se diseñaron archivos HTML y PHP específicos para cada vista, organizados de forma modular y reutilizable. Se incluyeron además scripts en JavaScript específicos para mejorar la interacción del usuario con las tablas de inventario.

3.4 Procedimientos de resolución y dificultades encontradas

Las principales dificultades encontradas y sus resoluciones fueron:

- **Conexión a la base de datos:** se resolvió configurando adecuadamente el archivo `conexion.php` con manejo de excepciones y testeo de la conexión de forma local y remota.
- **Validación de usuarios:** el procedimiento almacenado `PAVALIDAUSR` requería entender y manipular parámetros de salida, lo cual fue resuelto consultando documentación de PDO y realizando pruebas de conexión controladas.
- **Falta de documentación en procedimientos almacenados:** algunos procedimientos requerían parámetros exactos cuya interpretación no era evidente. Se hizo necesario revisar su comportamiento ejecutándolos directamente desde el entorno SQL y anotando su lógica.
- **Detección de cambios en las tablas:** diseñé una lógica basada en clases dinámicas (rojo/verde) para marcar visualmente los registros que el usuario modificaba, integrando esta lógica con botones habilitados/deshabilitados y llamadas AJAX seguras.
- **Paginación de tablas:** para que el sistema fuera usable incluso con muchos registros, se integró un sistema de paginación en JavaScript que permitía navegar entre páginas sin recargar la vista.

Gracias a un compañero que estaba pendiente de mi todos los días, he conseguido resolver todos los problemas encontrados a lo largo de los días. Estos solo son unos de los problemas con los que me he ido encontrando, en el día a día me encontraba con nuevos problemas, algunos más sencillos de detectar y otros más complicados que me llevaban más tiempo solucionar.

3.5 Resultados

Como resultado de las prácticas curriculares, se logró desarrollar una aplicación web funcional destinada a los transportistas de COFANO. Aunque el proyecto contemplaba inicialmente cuatro funcionalidades, el tiempo disponible permitió completar de forma íntegra una de ellas: la consulta y modificación del inventario.

La aplicación permite a los usuarios:

- Iniciar sesión mediante un sistema de autenticación conectado a la base de datos.
- Filtrar inventarios por tipo y estado.
- Consultar los detalles de un inventario concreto.
- Modificar unidades disponibles y fecha de caducidad.
- Guardar los cambios mediante procedimientos almacenados, con confirmación visual.

Además, se diseñó una estructura de código basada en el modelo MVC, lo que facilita futuras ampliaciones. El trabajo me permitió poner en práctica conocimientos técnicos (PHP, SQL, AJAX, JavaScript), resolver problemas reales y adquirir experiencia en el desarrollo de software con acceso a bases de datos empresariales.

4. Conclusiones

La realización de estas prácticas curriculares ha supuesto una experiencia muy enriquecedora tanto a nivel técnico como personal. Gracias a este proyecto he podido aplicar en un entorno real muchos de los conocimientos adquiridos durante la formación académica: desarrollo en PHP, conexión con bases de datos, uso de procedimientos almacenados, organización de código en arquitectura MVC, validación de usuarios, interacción con el backend, y control de errores, entre otros.

El hecho de tener que afrontar problemas reales, como errores de conexión, lógica de procedimientos almacenados o sincronización de datos modificados por el usuario, me ha obligado a pensar de forma práctica, investigar por mi cuenta y buscar soluciones por ensayo y error. Esto me ha permitido desarrollar mayor autonomía, criterio técnico y confianza en mis propias decisiones como desarrollador.

En este proceso ha sido especialmente importante el apoyo de mi compañero Sergio, quien desde el primer momento supo encontrar el equilibrio perfecto entre ayudarme y dejarme a mi aire. Su actitud me ha permitido aprender de verdad: me orientó en los momentos más complicados, pero me animó a que fuera yo mismo quien resolviera los problemas, buscara documentación, desarrollara métodos y aprendiera a salir adelante con mis propios recursos. Además, de él aprendí una lección fundamental: en el desarrollo de software no solo importa lo que sabes técnicamente, sino también para quién estás trabajando. Entender al cliente, comprender su necesidad real y adaptar el lenguaje técnico a la utilidad práctica del sistema es lo que diferencia a un buen programador de un buen profesional.

Gracias a esta experiencia, hoy me siento más preparado para enfrentarme a nuevos retos en el ámbito profesional, con una visión más clara del mundo real del desarrollo y de lo que supone trabajar en equipo, aportar valor y buscar soluciones útiles para personas concretas.

5. Bibliografía

- [1] «Manual PHP». [En línea]. Disponible en: <https://www.php.net/manual/es/index.php>.
- [2] «Stack overflow». [En línea]. Disponible en: <https://stackoverflow.com/questions>.
- [3] «Resources for Developers by Developers», *MDN*. [En línea]. Disponible en: <https://developer.mozilla.org/es/>.
- [4] «Microsoft Docs», *Microsoft*. [En línea]. Disponible en: <https://learn.microsoft.com/es-es/>.