TFG - ERP Modular para PYMEs

[TFG] ERP Modular para PYMEs

\*\*Autor:\*\* Javier Hortigüela Valiente

\*\*Tutor:\*\* Jose Vicente Carratalá Sanchis

\*\*Curso:\*\* Desarrollo de Aplicaciones Multiplataforma

# 1. Introducción

El presente Trabajo de Fin de Grado tiene como objetivo el desarrollo de un sistema ERP (Enterprise Resource Planning) modular, escalable y reutilizable, enfocado a cubrir las necesidades de gestión de empresas con un elevado volumen de datos y operaciones. Este sistema ha sido concebido desde un enfoque profesional, con una arquitectura MVC (Modelo–Vista–Controlador) implementada en PHP 8.3, y está diseñado para facilitar la integración de nuevos módulos de forma dinámica y segura, permitiendo su adaptación a múltiples tipos de empresas.

El proyecto surge tras un análisis detallado de los problemas comunes que enfrentan muchas pymes y grandes empresas: desorganización de datos, procesos manuales, falta de trazabilidad de operaciones y ausencia de herramientas de control en tiempo real. Frente a ello, se plantea como solución la creación de un ERP desde cero, con los principios de modularidad, dinamismo, seguridad y rendimiento como pilares esenciales.

Durante su desarrollo, se han implementado diferentes módulos: gestión de usuarios, empresas, pedidos y facturación, con edición inline, filtros dinámicos, búsqueda en vivo, paginación AJAX y exportación a Excel. Además, se ha desarrollado un sistema avanzado de roles y permisos para controlar el acceso a las distintas funcionalidades, lo que garantiza un alto grado de seguridad y control.

El ERP ha sido diseñado para funcionar en entornos productivos reales y escalables, con soporte para miles de registros. El presente documento describe en profundidad todas las fases del proyecto: análisis de necesidades, diseño técnico, planificación, ejecución, documentación, gestión del cliente, seguimiento, control de calidad y reflexión final. El tutor del proyecto es Jose Vicente Carratalá Sanchis, del ciclo de Desarrollo de Aplicaciones Multiplataforma.

# 2. Análisis

## 2.1 Recopilación de información

### 2.1.1 Empresas del sector por sus características organizativas y el tipo de producto o servicio que ofrecen

En el sector del software de gestión empresarial destacan empresas como Odoo, SAP, Holded o Zoho. Todas ellas ofrecen sistemas ERP orientados a diferentes tipos de negocios, desde pequeñas empresas hasta multinacionales. Estas plataformas se caracterizan por ofrecer funcionalidades como contabilidad, facturación, CRM, gestión de recursos humanos, almacén, etc.

Estas empresas tienen modelos organizativos complejos, con equipos dedicados a desarrollo, soporte técnico, marketing, atención al cliente, y ventas. Además, suelen ofrecer su producto como servicio (SaaS), con cuotas mensuales y escalado por usuario o funcionalidad.

### 2.1.2 Empresas tipo indicando la estructura organizativa y las funciones de cada departamento

Las empresas de software tipo se organizan generalmente en:

* **Departamento de desarrollo:** encargado de la creación, mantenimiento y mejora del software.
* **Departamento de soporte:** responsable de resolver incidencias y ayudar al cliente final.
* **Departamento comercial y marketing:** enfocado en la captación de clientes y posicionamiento del producto.
* **Departamento de administración:** gestiona contratos, facturación y recursos humanos.

Esta estructura es un modelo de referencia para plantear el proyecto ERP, que si bien es desarrollado por una sola persona, está pensado para adaptarse a este tipo de organizaciones.

## 2.2 Identificación y priorización de necesidades

La necesidad principal detectada es la carencia de una herramienta accesible y escalable para la gestión diaria de operaciones empresariales. Muchas empresas aún operan con Excel o herramientas fragmentadas, lo que genera errores, duplicidad de datos y pérdida de información.

### 2.2.1 Necesidades más demandadas a las empresas

* Control centralizado de clientes, pedidos y facturación
* Acceso multiusuario con diferentes permisos
* Visualización clara del estado de las operaciones
* Exportación de datos e integración con otros sistemas
* Seguridad en el acceso y trazabilidad de acciones

### 2.2.2 Oportunidades de negocio previsibles en el sector

Con la creciente digitalización de las empresas (especialmente con ayudas como el Kit Digital en España), se abre una oportunidad para herramientas ERP simples pero robustas, que puedan ser instaladas y gestionadas sin necesidad de licencias propietarias ni altos costes de mantenimiento.

### 2.2.3 Tipo de proyecto requerido para dar respuesta a las demandas previstas

El proyecto es una aplicación web con arquitectura MVC y backend en PHP 8.3. Incluye panel de control, tablas dinámicas con edición inline, sistema de autenticación, control de accesos y generación de documentos. Utiliza AJAX, HTML, JS y MySQL. El sistema está pensado para ser instalado en un servidor Apache/Nginx y accesible desde cualquier navegador moderno.

### 2.2.4 Características específicas del proyecto según los requerimientos

* Aplicación web sin necesidad de instalación local
* Modularidad para integrar fácilmente nuevos componentes (clientes, productos, incidencias, etc.)
* Panel de administración responsive
* Sistema de roles y permisos
* Generación de facturas en PDF con Dompdf
* Exportación de datos filtrados a Excel
* Soft delete y restauración de registros
* Filtros AJAX en tiempo real

## 2.3 Identificación de los aspectos que facilitan o dificultan el desarrollo

**Facilitadores:**

* Lenguajes conocidos (PHP, JS, SQL)
* Entorno local de desarrollo controlado (MAMP/Localhost)
* Disponibilidad de documentación sobre MVC y librerías externas

**Dificultades:**

* Complejidad de los módulos (facturación, permisos)
* Tiempo limitado
* Control de seguridad en sesiones, roles y acceso

### Obligaciones fiscales, laborales y de prevención

El proyecto no tiene obligaciones fiscales ni laborales directas, ya que es una práctica académica. No obstante, se sigue la normativa básica de protección de datos en el diseño de la aplicación.

### Ayudas o subvenciones

Proyectos como este pueden beneficiarse de programas como el Kit Digital, que financian soluciones ERP para pymes si el producto se convierte en una solución comercial SaaS o instalable.

### Guión de trabajo

1. Análisis de necesidades
2. Diseño de arquitectura MVC
3. Desarrollo de modelo base
4. Implementación de módulos (empresas, usuarios, pedidos)
5. Sistema de permisos y roles
6. Generación de informes y facturas
7. Documentación técnica y evaluación

# 3. Diseño

## 3.1 Definición o adaptación de la intervención

El proyecto consiste en desarrollar una solución ERP accesible, modular y escalable que permita gestionar las operaciones clave de cualquier empresa. La intervención resuelve un problema real: la ausencia de una herramienta gratuita, autogestionable y dinámica para empresas que no desean depender de plataformas SaaS externas.

La solución no solo responde a necesidades técnicas, sino que también propone una forma de organizar la información de forma ordenada, segura y reutilizable. El desarrollo ha partido desde cero, sin frameworks externos, para garantizar un mayor control sobre cada componente del sistema.

## 3.1.1 Información relativa a los aspectos que van a ser tratados

Los principales bloques funcionales del proyecto son:

* **Autenticación de usuarios:** login, logout y validación segura
* **Gestión de empresas:** edición inline, filtros dinámicos, eliminación/recuperación de registros
* **Gestión de usuarios:** creación, asignación de roles, control de accesos
* **Pedidos y facturación:** asignación de pedidos a empresas, confirmación de pedidos, generación de factura en PDF
* **Sistema de permisos:** middleware requirePermission(), panel visual de permisos, ocultación de opciones según rol
* **Exportación:** a Excel según filtros activos
* **Documentación:** código comentado, trazabilidad por fases, archivo de cambios (documentación técnica incluida)

## 3.2 Priorización y secuenciación de las acciones

El desarrollo del ERP se ha planteado como un proceso secuencial dividido en fases. Cada módulo se ha abordado de forma modular e independiente para facilitar su diseño, testeo y posible reutilización. Las fases se han ordenado de acuerdo a su dependencia lógica:

1. Diseño base del sistema (estructura MVC, enrutamiento, configuración).
2. Sistema de autenticación de usuarios y seguridad básica.
3. Módulo de empresas: CRUD completo con edición inline, búsqueda y filtros.
4. Módulo de usuarios: creación de usuarios y asignación de roles.
5. Sistema de logs de acceso.
6. Módulo de pedidos: relación con empresas, edición inline, filtros AJAX.
7. Generación de facturas en PDF.
8. Sistema de roles y permisos con panel visual.
9. Exportación a Excel.
10. Documentación técnica y pruebas finales.

## 3.2.1 Estudio de viabilidad técnica del proyecto

El proyecto es completamente viable técnica y temporalmente. Se ha desarrollado en un entorno local con recursos accesibles: PHP 8.3.14, MySQL, JavaScript y HTML/CSS. La complejidad de cada módulo se ha acotado para ajustarse al plazo de tres meses.

Además, se ha trabajado con principios de reutilización de código, evitando repeticiones y facilitando el mantenimiento. No se requieren conocimientos de tecnologías complejas ni licencias de software de pago, lo que refuerza la viabilidad global del proyecto.

## 3.3 La planificación de la intervención

La planificación del proyecto se ha estructurado siguiendo una metodología iterativa. Cada módulo se desarrolla de forma aislada, pero integrándose en el sistema general con una arquitectura MVC central. La planificación ha priorizado la funcionalidad básica y la estabilidad antes de abordar funciones avanzadas.

## 3.3.1 Fases del proyecto especificando su contenido y plazos de ejecución

| **Fase** | **Contenido** | **Plazo estimado** |
| --- | --- | --- |
| F1 | Estructura base del sistema (MVC, configuración, router, helpers) | Semana 1 |
| F2 | Sistema de login/logout, gestión de sesiones seguras, middleware requireLogin() | Semana 1 |
| F3 | Módulo Empresas: CRUD completo, edición inline, soft delete, paginación AJAX, filtros en vivo | Semana 2 |
| F4 | Módulo Usuarios: creación de usuarios, asignación de roles, validaciones básicas | Semana 2 |
| F5 | Sistema de logs de acceso (registro de login y logout) | Semana 3 |
| F6 | Sistema de roles y permisos: gestión visual de roles, asignación a usuarios, middleware requirePermission() | Semana 3 y 4 |
| F7 | Módulo Pedidos: relación con empresas, edición inline, filtros AJAX, restauración y eliminación lógica | Semana 5 |
| F8 | Generación de facturas en PDF desde pedidos confirmados (con Dompdf) | Semana 6 |
| F9 | Exportación a Excel de pedidos y empresas con filtros activos (PhpSpreadsheet) | Semana 6 |
| F10 | Pruebas finales, validación funcional, documentación técnica, mejoras finales y entrega | Semanas 7 y 8 |

## 3.4 Determinación de recursos

### 3.4.1 Objetivos que se pretenden conseguir identificando su alcance

* Crear un ERP funcional y ampliable
* Implementar módulos reales usados por empresas (empresas, usuarios, pedidos, facturación)
* Garantizar la seguridad mediante roles y permisos
* Preparar el sistema para producción (escalabilidad, mantenimiento, documentación)
* Generar informes y facturas de forma automatizada

### 3.4.2 Actividades necesarias para el desarrollo del proyecto

* Instalación de servidor local (MAMP o similar)
* Configuración de base de datos MySQL
* Creación de estructura de carpetas MVC
* Preparación de documentación técnica (fase a fase)
* Pruebas de funcionamiento en entorno real
* Revisión de código y comentarios

### 3.4.3 Recursos materiales y personales necesarios

* Ordenador personal con sistema operativo compatible
* Servidor local: MAMP, XAMPP o similar
* Editor de código: VS Code o PHPStorm
* Navegador moderno para pruebas (Chrome, Firefox)
* Dompdf para generación de PDFs
* Librerías PHP para exportar a Excel (PhpSpreadsheet)
* Usuario desarrollador (único programador)

### 3.4.4 Necesidades de financiación

No se requieren fondos externos. Todo el proyecto se ha desarrollado con herramientas de código abierto. La inversión en horas y recursos personales puede estimarse así:

* Licencia hipotética de editor profesional: 100 €
* Coste servidor (si se publica): 60 €/año
* Coste humano estimado (240 h a 15 €/h): 3.600 €

Coste estimado si se desarrollara profesionalmente: 3.760 €

## 3.5 Planificación de la evaluación

La evaluación se realizará al finalizar cada fase funcional. Para ello, se emplean criterios como el cumplimiento de funcionalidades, estabilidad, ausencia de errores y calidad del código. Además, se evalúan los tiempos de respuesta, cumplimiento de filtros, control de acceso y experiencia de usuario.

## 3.6 Diseño de documentación

La documentación del ERP se ha planteado como un documento estructurado en fases, incluyendo:

* Explicación del diseño del sistema (MVC)
* Comentarios línea a línea en el código fuente
* Documentación de los controladores, modelos y vistas
* Listado de endpoints, rutas y acciones
* Manual de administración para uso básico
* Manual técnico para desarrolladores

Se usará formato Markdown y PDF para su difusión. También se planea incluir capturas del funcionamiento y ejemplos prácticos.

## 3.6.1 Documentación necesaria para su diseño

* Reglamento de protección de datos (para evitar vulnerabilidades)
* Documentación de librerías externas como Dompdf y PhpSpreadsheet
* Buenas prácticas PHP/MVC
* Referencias a proyectos similares en GitHub y foros de desarrollo
* Material del ciclo formativo (MVC, seguridad, SQL)

## 3.7 Plan de atención al cliente

El proyecto contempla un sistema de soporte básico post-implementación. Se establece un canal de contacto para la empresa o usuario final, a través del cual podrá:

* Reportar errores
* Solicitar mejoras o ampliaciones
* Solicitar formación en el uso del ERP

La atención sería por email y documentación, con posibilidad de actualizaciones periódicas.

## 3.7.1 Aspectos que se deben controlar para garantizar la calidad del proyecto

* Calidad del código (sintaxis clara, sin duplicaciones)
* Seguridad (escape de entradas, validación de roles)
* Modularidad (código reutilizable y separado por responsabilidad)
* Escalabilidad (soporte para miles de registros)
* Estilo profesional en interfaces y paneles
* Control de errores (mensajes de validación y mensajes flash)

# 4. Organización

## 4.1 Detección de demandas y necesidades

Antes del inicio del desarrollo del ERP, fue necesario detectar y analizar todas las tareas y condiciones necesarias para implementar el sistema con éxito. Este análisis permitió anticiparse a los posibles obstáculos y planificar adecuadamente cada fase del proyecto.

### 4.1.1 Tareas en función de las necesidades de implementación

* Crear base de datos y relaciones entre tablas (empresas, usuarios, pedidos, roles, logs, permisos).
* Desarrollar estructura MVC base (router, controlador principal, modelo base, vistas dinámicas).
* Implementar sistema de login/logout y validación de sesiones.
* Programar módulo de empresas: tabla, filtros, edición inline, alta/baja lógica.
* Programar módulo de usuarios con rol asignable.
* Registrar logs de acceso.
* Desarrollar módulo de pedidos vinculado a empresas.
* Implementar sistema de permisos con controladores protegidos y botones ocultos por rol.
* Generar PDF y Excel exportables.
* Crear formulario de empresa dinámico desde BBDD.
* Documentar todo el sistema técnica y funcionalmente.

### 4.1.2 Recursos y logística necesaria para cada tarea

* **Servidor local** (MAMP) y base de datos MySQL.
* **Editor de código profesional** (VS Code / PHPStorm).
* **Navegador actualizado** para pruebas visuales.
* **Librerías externas** como Dompdf y PhpSpreadsheet.
* **Entorno controlado y backup regular** para evitar pérdidas de progreso.

### 4.1.3 Necesidades de permisos y autorizaciones

No se requiere ningún permiso externo, ya que el desarrollo ha sido individual y local, sin tratamiento de datos sensibles reales ni integración con sistemas externos de terceros.

## 4.2 Programación

Se ha utilizado una planificación semanal secuencial para el desarrollo de los módulos. Cada fase ha sido planificada de forma independiente, dejando margen para la detección y corrección de errores.

### 4.2.1 Procedimientos para ejecución de las tareas

* Uso de un archivo funcionespendientes.txt para listar tareas diarias
* Revisión manual del funcionamiento de cada módulo antes de pasar al siguiente
* Validación de entradas desde cliente y servidor
* Separación de lógica PHP y JS por módulo
* Creación de backups semanales

## 4.3 Gestión

La gestión del proyecto se ha basado en el cumplimiento de objetivos semanales y validaciones funcionales tras cada entrega parcial. Se ha seguido una metodología propia basada en micro-sprints semanales.

### 4.3.1 Riesgos inherentes y plan de prevención

**Riesgos detectados:**

* Cansancio visual y fatiga postural
* Corrupción de archivos por error del sistema
* Pérdida de datos o progreso por error humano

**Medidas tomadas:**

* Pausas regulares y postura correcta
* Uso de copia de seguridad local y en la nube
* Documentación paso a paso para evitar dependencias mentales

## 4.4 Coordinación y supervisión de la intervención

Al ser un proyecto individual, el mismo desarrollador ha ejercido tanto de programador como de supervisor. No obstante, se ha aplicado una rutina de revisiones internas al finalizar cada módulo.

### 4.4.1 Asignación de recursos materiales y humanos

| **Tarea** | **Recursos materiales** | **Recursos humanos** | **Tiempo estimado** |
| --- | --- | --- | --- |
| Desarrollo MVC | Ordenador + servidor local | 1 programador | 3 semanas |
| Módulo empresas | PHP, JS, MySQL | 1 programador | 4 semanas |
| Roles y permisos | PHP + lógica personalizada | 1 programador | 1 semana |
| Exportar PDF/Excel | Dompdf, PhpSpreadsheet | 1 programador | 2 semanas |

### 4.4.2 Valoración económica

| **Recurso** | **Coste estimado** |
| --- | --- |
| Ordenador (portátil profesional) | 900 € |
| Editor profesional (licencia hipotética) | 100 € |
| Tiempo de desarrollo (240 h a 15 €/h) | 3.600 € |
| Servidor anual básico (hosting real) | 60 € |
| Librerías externas (uso libre) | 0 € |

**Coste estimado total: 4.660 €**

## 4.5 Elaboración de informes

Se han planificado informes semanales internos (resumen de avances por fase), y un informe final con detalles técnicos y resultados obtenidos.

### 4.5.1 Documentación necesaria para la ejecución

* Diagrama de base de datos relacional
* Mapa de navegación de rutas y controladores
* Plantilla de código base reutilizable
* Manual de usuario y documentación de código fuente
* Pruebas de funcionalidad y casos de uso

## 4.6 Seguimiento y control

El seguimiento del proyecto ha incluido validaciones internas continuas, revisión tras cada módulo, y planificación de mejoras continuas.

### 4.6.1 Procedimiento de evaluación de actividades

* Validación funcional tras cada entrega
* Testeo de todas las rutas y formularios
* Validación de seguridad: autenticación, permisos, errores

### 4.6.2 Indicadores de calidad

* Código modular y comentado
* Validación de entradas cliente-servidor
* Funcionalidades escalables
* Uso de AJAX y rendimiento sin recargas
* Interfaz clara y accesible

### 4.6.3 Procedimiento para el registro y evaluación de incidencias

* Registro manual en funcionespendientes.txt
* Clasificación de errores por prioridad
* Análisis de causa y solución aplicada

### 4.6.4 Procedimiento para la solución de incidencias

* Revisión del código implicado
* Refactorización si procede
* Comprobación funcional tras la corrección

### 4.6.5 Procedimiento para gestión de cambios

* Registro de cambios en documentacion/
* Notas de mejora por fase
* Refactorización modular de funciones comunes

### 4.6.6 Procedimiento para evaluación de usuarios

En un futuro entorno productivo, se plantea realizar pruebas con usuarios reales (por ejemplo, personal administrativo) para validar el uso del sistema ERP y detectar mejoras desde la experiencia de usuario.

### 4.6.7 Sistema para garantizar cumplimiento del pliego

Aunque no existe un pliego oficial, se ha seguido una guía de hitos entregables, revisando que cada uno cumpla los objetivos previstos antes de avanzar al siguiente. Se han documentado cada fase, errores encontrados y soluciones implementadas.

# 5. Actividades profesionales

## 5.1 Áreas de sistemas y departamentos de informática en cualquier sector de actividad

El ERP desarrollado está diseñado para ser utilizado principalmente por departamentos de informática en empresas de tamaño medio o grande. Estas áreas pueden:

* Administrar usuarios y roles, asegurando el control del acceso a la información.
* Dar soporte técnico a los usuarios internos que utilizan el ERP.
* Gestionar los módulos activos del sistema, personalizarlos o ampliar sus funcionalidades.
* Velar por el cumplimiento de buenas prácticas de seguridad informática y auditoría interna.

Gracias a su estructura modular y su código comentado, el sistema es mantenible y ampliable por técnicos internos, reduciendo la dependencia de proveedores externos.

## 5.2 Sector de servicios tecnológicos y comunicaciones

Este ERP también puede formar parte del catálogo de servicios ofrecidos por empresas del sector tecnológico. Puede ser integrado como:

* Solución ERP bajo licencia, personalizable por cada cliente.
* Servicio SaaS (Software as a Service) si se despliega en un entorno cloud.
* Plataforma base para desarrolladores de terceros que busquen una solución adaptable.

Su diseño modular permite a empresas de desarrollo usarlo como base para nuevos proyectos o integrarlo con servicios ya existentes como CRMs, plataformas de e-commerce o herramientas contables.

## 5.3 Área comercial con gestión de transacciones por Internet

Desde un enfoque comercial, el ERP es especialmente útil para:

* Empresas con actividad B2B que gestionan relaciones con clientes y pedidos frecuentes.
* Departamentos de administración que necesitan controlar pedidos, generar facturas y exportar datos.
* Personal no técnico que requiere interfaces claras y acciones intuitivas (crear empresa, filtrar pedidos, exportar a Excel, etc.).

En este sentido, el sistema está pensado para dar soporte directo a la operativa diaria, mejorando la eficiencia administrativa y la trazabilidad comercial.

# 6. Líneas de actuación

## 6.1 La ejecución de trabajos en equipo

Aunque el proyecto ha sido desarrollado individualmente durante un periodo de 2 meses, se ha seguido una metodología que permite su futura colaboración en equipo:

* Código modular y bien comentado.
* Separación lógica entre backend (PHP), frontend (HTML/JS) y modelo de datos (MySQL).
* Preparación del sistema para entornos colaborativos con control de versiones (ej. Git).

En un futuro, diferentes desarrolladores podrían trabajar en paralelo sobre nuevos módulos (inventario, incidencias, almacén), sin interferir en el núcleo del sistema.

## 6.2 La autoevaluación del trabajo realizado

Durante el desarrollo, se ha llevado un registro de errores y mejoras pendientes en el archivo funcionespendientes.txt, como herramienta de autoseguimiento.

También se ha hecho una evaluación continua de cada módulo, mediante:

* Pruebas funcionales tras cada implementación
* Validación de requisitos (edición inline, permisos, filtros, etc.)
* Refactorización de código duplicado

Esta reflexión activa ha permitido mejorar la calidad del código y garantizar un desarrollo profesional.

## 6.3 La autonomía y la iniciativa

Este proyecto demuestra una autonomía total en:

* Elección del enfoque técnico (PHP nativo, sin frameworks)
* Diseño de la arquitectura base MVC desde cero
* Desarrollo completo de módulos, sin plantillas preexistentes
* Solución de errores y decisiones de implementación sin supervisión constante

Además, se ha mantenido una motivación constante durante las 8 semanas de trabajo, lo que ha permitido cumplir todos los objetivos y superar problemas inesperados con rapidez.

## 6.4 El uso de las TIC

Las TIC han sido el eje fundamental del proyecto. Se ha hecho uso intensivo de tecnologías y herramientas como:

* **PHP 8.3:** backend del sistema, seguro y escalable
* **MySQL:** base de datos relacional con consultas optimizadas
* **JavaScript y AJAX:** para la edición inline y carga dinámica de datos
* **HTML/CSS:** para la interfaz visual adaptable
* **Dompdf y PhpSpreadsheet:** para la generación de facturas y exportaciones
* **Git:** como sistema de control de versiones local
* **VS Code y DevTools:** como entornos de trabajo profesionales

El uso de las TIC ha permitido crear un sistema realista, avanzado, con una arquitectura escalable y listo para funcionar en entornos empresariales reales.

# 7. Bibliografía

A lo largo del desarrollo de este proyecto se han consultado diversas fuentes técnicas, documentación oficial, recursos académicos y experiencias de desarrolladores que han servido de guía y referencia para construir un sistema ERP profesional, modular y escalable. La siguiente bibliografía recoge aquellas más relevantes:

### Manuales y documentación oficial

* **PHP Documentation.** <https://www.php.net/manual/en/>  
  Referencia oficial del lenguaje PHP, consultada regularmente para el manejo de funciones, seguridad y buenas prácticas en PHP 8.3.
* **MySQL 8.0 Reference Manual.** Oracle Corporation.  
  <https://dev.mysql.com/doc/>  
  Documentación utilizada para el diseño relacional de la base de datos, consultas SQL y optimización de índices.
* **JavaScript MDN Web Docs.** Mozilla Foundation.  
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript>  
  Documentación de referencia para eventos DOM, AJAX y funciones modernas de JS.
* **Dompdf GitHub Repository.**  
  <https://github.com/dompdf/dompdf>  
  Utilizado para la generación de facturas en formato PDF desde los pedidos.
* **PhpSpreadsheet GitHub Repository.**  
  <https://github.com/PHPOffice/PhpSpreadsheet>  
  Librería usada para la exportación de datos de empresas y pedidos a formato Excel (.xlsx).

### Recursos académicos y educativos

* **Carratalá, J. V.** Material docente del módulo de proyecto del ciclo DAM.  
  Referencias y plantillas proporcionadas en clase para estructurar la memoria técnica y la planificación del proyecto.
* **OpenWebinars.** Curso de MVC en PHP sin frameworks.  
  <https://openwebinars.net/>  
  Consultado para afianzar conceptos de arquitectura MVC en PHP puro.
* **W3Schools.** Manuales introductorios a HTML5, CSS y JavaScript.  
  <https://www.w3schools.com/>  
  Apoyos básicos para diseño de interfaz de usuario en los módulos del ERP.

### Buenas prácticas y desarrollo profesional

* **Laracasts Forum.** Casos de uso reales sobre autenticación, roles y permisos.  
  https://laracasts.com/discuss
* **Stack Overflow.** Solución de errores y dudas técnicas comunes en entornos PHP/MySQL.  
  <https://stackoverflow.com/>
* **GitHub – Repositorios de proyectos ERP Open Source.**  
  Consultados para validar estructuras de bases de datos, patrones MVC, o sistemas de permisos.