TFG - ERP Modular para PYMEs

[TFG] ERP Modular para PYMEs

\*\*Autor:\*\* [Tu nombre]

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

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

**1. Introducción**

Este Trabajo de Final de Grado (TFG) tiene como objetivo el diseño y desarrollo de un sistema ERP (Enterprise Resource Planning) genérico y modular, orientado a facilitar la gestión interna de pequeñas y medianas empresas. A diferencia de otros sistemas ERP diseñados para un sector específico, esta solución está pensada para ser instalada fácilmente en cualquier tipo de empresa, independientemente de su actividad económica.

El ERP contará con los módulos fundamentales de:

* Gestión de clientes y proveedores
* Gestión de productos y servicios
* Facturación básica
* Gestión de usuarios y roles
* Informes exportables (PDF, Excel)

Se trata de una aplicación web desarrollada con tecnologías estándares del ciclo de DAM: PHP, MySQL, HTML, CSS, JavaScript y Bootstrap. El backend estará estructurado en PHP puro con arquitectura MVC simplificada o usando Laravel, según la complejidad asumible durante el desarrollo.

Este sistema ERP contribuye a la mejora organizacional de empresas que aún no han digitalizado sus procesos internos, promoviendo una gestión más eficiente y ordenada. De este modo, se alinea con la filosofía de "Programas para un mundo mejor" al ofrecer una herramienta de uso libre y socialmente útil.

**2. Análisis**

**2.1. Recopilación de información**

**2.1.1. Empresas del sector por sus características organizativas**

Las empresas que desarrollan ERP suelen dividirse entre grandes proveedores (SAP, Oracle, Microsoft Dynamics) y soluciones más ligeras orientadas a pymes (Odoo, Dolibarr, FacturaScripts, Holded).

Estas soluciones tienen como característica común la integración de varios departamentos en una sola plataforma: compras, ventas, almacén, contabilidad, etc.

**2.1.2. Empresas tipo indicando estructura organizativa**

En general, estas empresas cuentan con los siguientes departamentos:

* Desarrollo: encargado del código y mantenimiento
* Soporte: atención al cliente
* Comercial: captación y retención de clientes
* Dirección técnica y de producto

Nuestro ERP replica las funcionalidades básicas para cubrir las necesidades de estos departamentos desde una instalación simple.

**2.2. Identificación y priorización de necesidades**

**2.2.1. Necesidades más demandadas a las empresas**

* Tener control de facturas, productos y clientes en una sola herramienta
* Reducir el uso de hojas de cálculo dispersas
* Generar documentos legales (presupuestos, facturas)
* Agilizar la atención al cliente

**2.2.2. Oportunidades de negocio previsibles en el sector**

* Digitalización de empresas pequeñas post-pandemia
* Interés creciente en soluciones autoalojadas y gratuitas
* Software como servicio con base en ERP de código abierto

**2.2.3. Tipo de proyecto requerido**

Una aplicación web accesible desde cualquier navegador y alojada en un servidor web. No es necesario instalar apps. Arquitectura cliente-servidor clásica.

**2.2.4. Características específicas del proyecto**

* Interfaz web responsiva
* Permisos por roles (admin, empleado, contable)
* Generación de PDF y exportación de Excel
* Interfaz intuitiva basada en Bootstrap

**2.3. Identificación de los aspectos que facilitan o dificultan el desarrollo**

**Facilitan:**

* Tecnologías conocidas del ciclo (PHP, MySQL)
* Reutilización de plantillas de interfaz (AdminLTE)
* Desarrollo iterativo por módulos

**Dificultan:**

* Complejidad del control de acceso y validaciones cruzadas

**Obligaciones legales:**

* LOPDGDD y RGPD si se trata información personal de clientes/proveedores

**Subvenciones:**

* Puede acogerse a programas de digitalización de pymes

**Guión de trabajo:**

* Fase 1: Backend, base de datos y autenticación
* Fase 2: Módulos funcionales, exportaciones y testeo

(Ahora continuaré con el apartado 3. Diseño, y luego los siguientes hasta completar todo el documento)

**3. Diseño**

**3.1. Definición o adaptación de la intervención**

El proyecto consiste en una intervención tecnológica que pretende solucionar la falta de digitalización en la gestión interna de muchas pequeñas y medianas empresas. Esta falta de herramientas integradas genera problemas de eficiencia, errores administrativos y pérdida de información. El sistema ERP propuesto aborda directamente estos problemas mediante una solución modular, accesible desde cualquier navegador, y adaptable a distintos modelos de negocio.

La intervención se justifica por la necesidad de disponer de una herramienta unificada para la gestión empresarial. Mediante este ERP, las empresas podrán gestionar clientes, proveedores, productos, facturas e informes desde un único entorno, con acceso restringido por roles y con interfaz clara y profesional.

**3.1.1. Información relativa a los aspectos que van a ser tratados en el proyecto**

El ERP contará con los siguientes módulos o componentes principales:

* **Módulo de autenticación**: permite iniciar sesión a usuarios registrados, con control de acceso por rol.
* **Módulo de gestión de clientes y proveedores**: alta, modificación, baja y listado de entidades externas.
* **Módulo de productos y servicios**: gestión del catálogo interno de la empresa.
* **Módulo de facturación**: generación de facturas a partir de clientes y productos, con cálculo automático de totales e impuestos.
* **Módulo de usuarios y roles**: creación de cuentas de usuario con permisos diferenciados.
* **Módulo de informes**: exportación de datos a PDF y Excel.

Cada módulo estará estructurado según el patrón MVC, separando claramente la lógica, la presentación y la manipulación de datos. El sistema estará diseñado para ser fácilmente extensible y mantenible.

(Continuará con los apartados 3.2 en adelante)

**3.2. Priorización y secuenciación de las acciones**

El proyecto se desarrollará siguiendo una metodología iterativa por fases, lo cual permite validar cada módulo individualmente y ajustar el rumbo en caso necesario. La secuencia de desarrollo planificada es:

1. **Diseño de la base de datos y arquitectura MVC del backend**
2. **Módulo de autenticación y gestión de usuarios**
3. **Gestión de clientes y proveedores**
4. **Módulo de productos y servicios**
5. **Facturación básica y generación de PDF**
6. **Exportación de informes a Excel**
7. **Estilizado y responsividad del frontend con Bootstrap**
8. **Pruebas, validaciones, optimizaciones**

**3.2.1. Estudio de viabilidad técnica del proyecto**

Desde el punto de vista técnico, el desarrollo del ERP es perfectamente viable utilizando tecnologías que han sido estudiadas durante el ciclo formativo. PHP y MySQL permiten una estructura robusta y probada para sistemas de este tipo.

La elección de una arquitectura MVC simplificada permitirá separar la lógica del negocio, la presentación y el acceso a datos, mejorando la mantenibilidad y la escalabilidad. Además, al no depender de tecnologías móviles o entornos externos, se reduce la complejidad del despliegue y la curva de aprendizaje.

(Continuaré con los puntos 3.3 y siguientes)

**3.3. La planificación de la intervención**

A continuación se describen las fases de ejecución del proyecto, cada una con sus tareas principales y objetivos:

**3.3.1. Fases del proyecto especificando su contenido y plazos de ejecución**

* **Fase 1: Análisis y diseño** Definición del modelo de datos, diagrama entidad-relación, estructura del proyecto, elección de tecnologías.
* **Fase 2: Desarrollo del sistema base** Implementación del sistema de login, roles y gestión de usuarios.
* **Fase 3: Desarrollo de módulos funcionales** Gestión de clientes, proveedores, productos y facturación.
* **Fase 4: Exportación y generación de informes** Generación de documentos PDF y exportación de datos a Excel.
* **Fase 5: Estilizado, pruebas y validaciones** Aplicación de estilos con Bootstrap, pruebas funcionales y validaciones de entrada.
* **Fase 6: Documentación y entrega final** Elaboración del manual de usuario, documentación técnica y preparación de la exposición.

**3.4. Determinación de recursos**

**3.4.1. Objetivos que se pretenden conseguir identificando su alcance**

* Desarrollar una aplicación ERP modular y autoalojable
* Permitir su instalación en distintas empresas sin necesidad de adaptación técnica compleja
* Implementar funcionalidades esenciales de gestión interna empresarial
* Documentar el proyecto de forma profesional para facilitar su comprensión y mantenimiento

**3.4.2. Actividades necesarias para el desarrollo del proyecto**

* Adquisición o uso de equipo informático con entorno de desarrollo local (XAMPP/WAMP)
* Instalación de software de control de versiones (Git)
* Creación de cuenta en GitHub para repositorio privado/público
* Configuración de entorno de pruebas para despliegue web (hosting local o servidor compartido)

**3.4.3. Recursos materiales y personales necesarios para realizar el proyecto**

* Ordenador personal con acceso a internet
* Servidor local (XAMPP, WAMP o similar)
* Software de desarrollo (Visual Studio Code, PHPMyAdmin, navegador web)
* Acceso a herramientas online para pruebas de exportación PDF/Excel

**3.4.4. Necesidades de financiación para la puesta en marcha del proyecto**

No se requieren fondos externos para el desarrollo inicial. Todos los recursos utilizados son gratuitos o de software libre. Se estima un coste nulo para el desarrollo y pruebas locales. En caso de producción, podría requerirse dominio y hosting (~60€/año).

**3.5. Planificación de la evaluación**

Se evaluará la funcionalidad de cada módulo desarrollado, así como su integración completa. Los criterios incluirán:

* Cumplimiento de requisitos funcionales
* Interfaz clara y uso intuitivo
* Correcto funcionamiento de roles y permisos
* Correcta generación de documentos y exportaciones

**3.6. Diseño de documentación**

La documentación se dividirá en dos tipos:

* **Manual técnico**: arquitectura del sistema, estructura del proyecto, instalación, configuración, y mantenimiento
* **Manual de usuario**: instrucciones paso a paso para el uso de cada módulo

Ambos documentos estarán en formato PDF e incluirán capturas de pantalla, esquemas y ejemplos prácticos.

**3.6.1. Documentación necesaria para su diseño**

* Documentación oficial de PHP, MySQL, Bootstrap
* Normativas legales sobre protección de datos (GDPR, LOPDGDD)
* Ejemplos de manuales técnicos y de usuario de ERPs similares

**3.7. Plan de atención al cliente**

El sistema estará diseñado para ofrecer soporte técnico básico y posibilidad de actualizaciones mediante repositorio GitHub. Se contempla:

* Sistema de contacto dentro de la aplicación (en futuras versiones)
* Inclusión de documentación completa accesible desde el panel principal

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

* Pruebas funcionales de cada módulo
* Validación de formularios y manejo de errores
* Comentado del código fuente con buenas prácticas
* Revisión cruzada del sistema con checklist de control de calidad

(Continuaré con el apartado 4. Organización)

**4. Organización**

**4.1. Detección de demandas y necesidades**

La necesidad principal es la falta de digitalización de procesos administrativos en muchas pymes. Estas empresas requieren una herramienta versátil, fácil de usar y económica para gestionar sus recursos de manera efectiva.

**4.1.1. Tareas en función de las necesidades de implementación**

* Diseño del modelo de datos
* Configuración del entorno de desarrollo
* Desarrollo modular del backend y frontend
* Pruebas de validación y verificación
* Documentación técnica y de usuario

**4.1.2. Recursos y la logística necesaria para cada tarea**

* Ordenador con servidor local configurado (XAMPP/WAMP)
* Editor de código (Visual Studio Code)
* Navegador web para pruebas
* Acceso a GitHub para control de versiones

**4.1.3. Necesidades de permisos y autorizaciones**

No se requiere autorización específica, al tratarse de un proyecto individual y académico. El tratamiento de datos personales se limitará a pruebas ficticias.

**4.2. Programación**

El proyecto se ejecutará siguiendo una programación temporal basada en fases semanales. Se utilizará un diagrama de Gantt para visualizar y controlar el progreso.

**4.2.1. Procedimientos para ejecución de las tareas**

* Uso de ramas en Git para cada módulo
* Validación del funcionamiento antes de integrar a la rama principal
* Backup periódico del proyecto

**4.3. Gestión**

Se establecerá un sistema de control de entregas parciales y revisión periódica.

**4.3.1. Riesgos inherentes a la ejecución del proyecto**

* Errores de programación (debugging intensivo)
* Pérdida de datos (uso de Git y copias de seguridad)
* Fatiga visual o postural (pausas y ergonomía adecuada)

**4.4. Coordinación y supervisión de la intervención**

La supervisión se llevará a cabo mediante revisión continua del código y avance funcional.

**4.4.1. Asignación de recursos materiales y humanos**

* Materiales: software libre y recursos propios
* Humanos: desarrollador único (el alumno)
* Tiempo estimado por módulo: 1-2 días

**4.4.2. Valoración económica**

* Hosting: 60€/año (opcional)
* Desarrollo: estimación profesional sería de 2000€–3000€

**4.5. Elaboración de informes**

Se prevé la redacción de un informe final de proyecto que incluya:

* Metodología utilizada
* Capturas de los módulos
* Problemas encontrados y soluciones

**4.5.1. Documentación necesaria para la ejecución del proyecto**

* Diagrama entidad-relación
* Estructura del repositorio Git
* Manual técnico en borrador desde el inicio del desarrollo

**4.6. Seguimiento y control**

Se aplicarán controles de calidad internos en cada fase.

**4.6.1. Procedimiento de evaluación de actividades**

* Checklist por módulo desarrollado
* Pruebas funcionales individuales

**4.6.2. Indicadores de calidad**

* Código limpio y comentado
* Validación de entradas
* Ausencia de errores visibles

**4.6.3. Registro y evaluación de incidencias**

* Documentación en archivo markdown por cada incidencia
* Seguimiento mediante commits y comentarios en GitHub

**4.6.4. Solución de incidencias**

* Debugging guiado por logs de errores
* Uso de foros o documentación oficial si es necesario

**4.6.5. Gestión de cambios**

* Anotación de cada cambio en CHANGELOG.md
* Control de versiones por etiquetas (tags) en GitHub

**4.6.6. Participación de usuarios**

* Evaluación informal por parte de otros compañeros o usuarios ficticios

**4.6.7. Cumplimiento del pliego de condiciones**

* Revisión final mediante checklist de requisitos funcionales acordados

(Continuará con el apartado 5. Actividades profesionales)

**5. Actividades profesionales**

**5.1. Áreas de sistemas y departamentos de informática en cualquier sector de actividad**

Este ERP puede ser implementado por departamentos de informática de cualquier empresa para mejorar la organización de los datos internos, automatizar procesos y centralizar la información administrativa. También puede servir como punto de partida para futuras integraciones con herramientas de análisis o aplicaciones móviles.

**5.2. Sector de servicios tecnológicos y comunicaciones**

El sistema podría formar parte de una oferta comercial como software como servicio (SaaS) para pequeñas empresas que no cuentan con infraestructura ni presupuesto para soluciones de gran escala. También puede ser mantenido y personalizado por empresas del sector TIC como producto de entrada para clientes.

**5.3. Área comercial con gestión de transacciones por Internet**

El ERP se adapta fácilmente a cualquier organización con necesidades de facturación, inventario y gestión de relaciones con clientes, por lo que sería perfectamente viable para su uso en entornos comerciales que operen online o incluso para autónomos que ofrezcan servicios digitales o físicos.

(Continuará con el apartado 6. Líneas de actuación)

**6. Líneas de actuación**

**6.1. La ejecución de trabajos en equipo**

Aunque el desarrollo ha sido individual, se han seguido buenas prácticas que facilitarían el trabajo en equipo: uso de Git, separación por módulos y documentación clara. El sistema está diseñado para que, en un entorno real, varios desarrolladores puedan trabajar de forma colaborativa.

**6.2. La autoevaluación del trabajo realizado**

Durante todo el proceso, se han establecido mecanismos de autoevaluación en cada fase mediante pruebas funcionales, checklist de objetivos y revisiones del código. Esto ha permitido detectar errores tempranos y mejorar la calidad del resultado final.

**6.3. La autonomía y la iniciativa**

El proyecto ha sido concebido, diseñado y desarrollado de forma totalmente autónoma. Desde la elección del tema hasta la implementación, se ha demostrado capacidad de planificación, toma de decisiones y resolución de problemas sin dependencia directa del tutor, más allá de las sesiones de seguimiento.

**6.4. El uso de las TIC**

El proyecto se basa totalmente en el uso de tecnologías de la información y la comunicación. Se ha empleado software libre, sistemas de control de versiones, plataformas de documentación colaborativa y herramientas de diseño web. Estas herramientas han sido fundamentales tanto para el desarrollo técnico como para la organización y documentación del trabajo.

(Continuará con el apartado 7. Bibliografía)

**7. Bibliografía**

* Documentación oficial de PHP: https://www.php.net/manual/es/
* Documentación oficial de MySQL: https://dev.mysql.com/doc/
* Bootstrap Documentation: https://getbootstrap.com/docs/
* AdminLTE Templates: https://adminlte.io/
* GitHub Docs: https://docs.github.com/es
* Reglamento General de Protección de Datos (RGPD): https://www.aepd.es/es
* LOPDGDD (Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales)
* Plantillas de proyectos ERP en Odoo y FacturaScripts para referencias comparativas