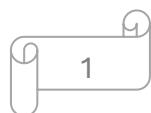


Jonatan Cruz Bueso
José Miguel Sentamans Dasí
IES Eduardo Primo Marqués
20/11/2025

MEMORIA FASE 1 DEL PROYECTO INTERMODULAR

Contenido

Planteamiento Inicial.....	2
Panel de Trello.....	2
Repositorio de GitHub	3
Análisis y diseño de la Base de Datos	4
Diagrama de entidad relación	4
Diagrama de clases	5
Diagrama de casos de uso.....	6
Requisitos funcionales de la app	7
Requisitos no funcionales de la app	9
Requisitos funcionales del ERP	11
Requisitos no funcionales del ERP.....	13



Planteamiento Inicial

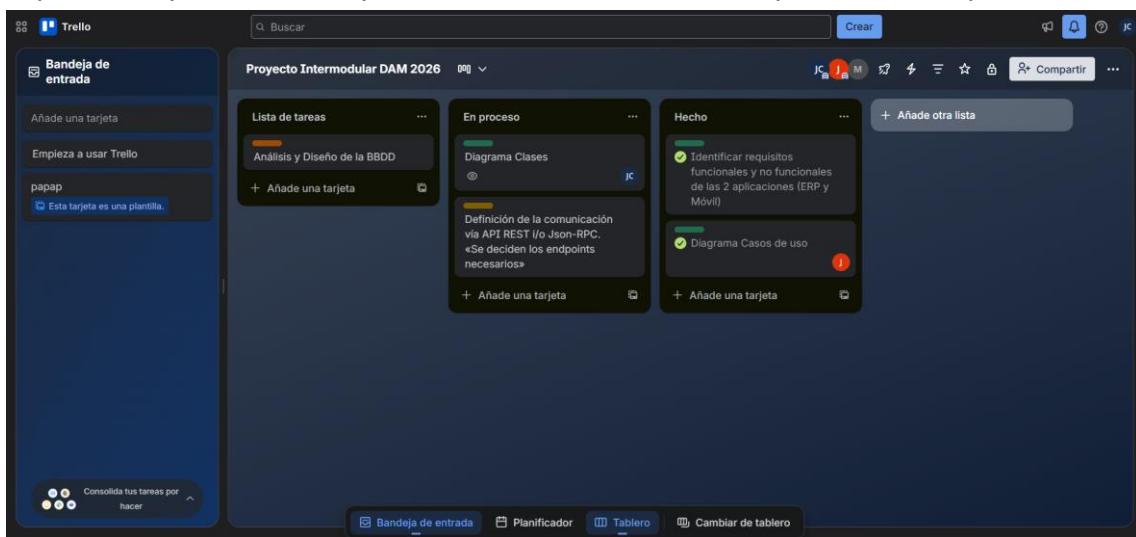
El objetivo de la fase 1 fue reunir información y empezar con el planteamiento lógico de como la aplicación funcionará, eso incluyó muchas tareas de recopilación y análisis que se destacarán más adelante en el documento, pero en resumen todo el trabajo hecho se agrupa en las siguientes tareas:

- Un Panel en Trello
- Un repositorio en GitHub
- Requisitos funcionales y no funcionales de las 2 aplicaciones (ERP y Móvil)
- Análisis y Diseño de la BBDD
- Diagrama de entidad relación
- Diagrama de clases
- Diagrama de casos de uso
- Arquitectura general del sistema (documento de los endpoints)

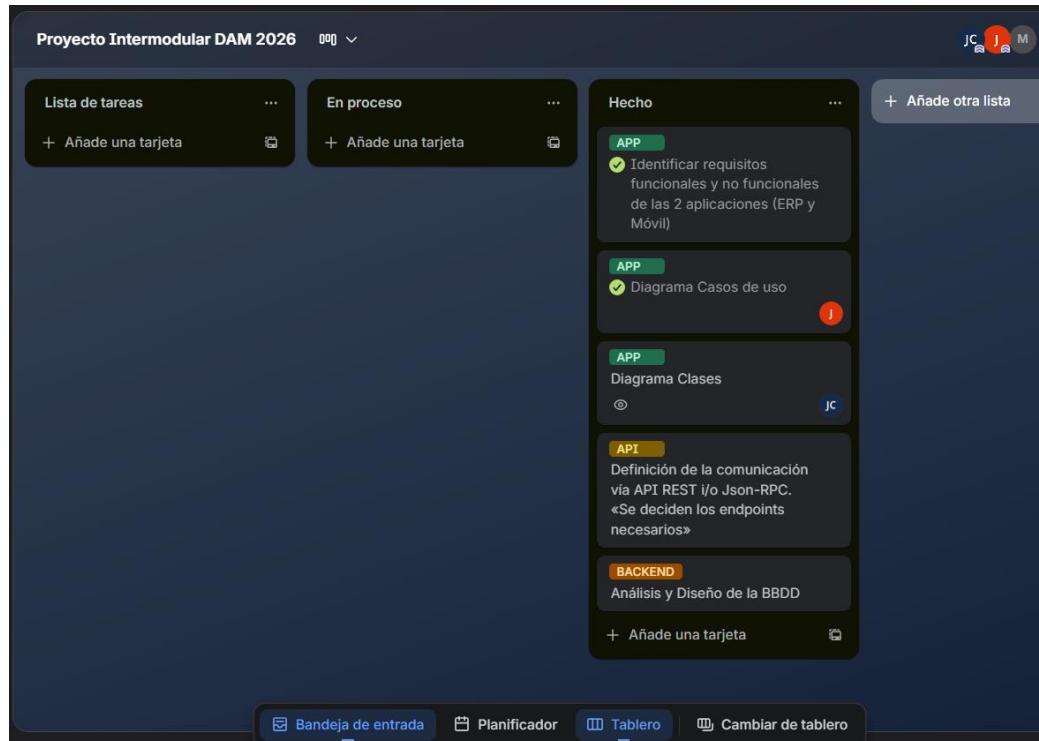
Panel de Trello

Utilizamos trello para organizar las tareas entre nosotros, dividiéndolo en los 3 paneles principales: **Lista de tareas, En proceso y Hecho**

Aquí una captura de los 3 paneles con tareas durante el proceso de la primer fase



Una vez finalizadas todas las tareas estás pasarán al último panel, para saber a qué apartado de la app pertenecía cada tarea las etiquetamos para tener una mejor orientación de lo que ya tenemos hecho y de lo que nos faltaba.



Repositorio de GitHub

Creamos un repositorio de [GitHub](#) el cual enlazamos a nuestros repositorios locales para poder trabajar con él. Su estructura es bastante sencilla, con las siguientes carpetas y archivos en la raíz de este:

A screenshot of a GitHub repository page. The repository name is "jonatan-cruz / proyecto-intermodular-dam-25-26--Jonatan-Jose-". It is a public repository with 1 branch and 0 tags. The commit history shows the following initial commits:

- jonatan-cruz Agregado el documento de requisitos de la app. (1 minute ago)
- documentacion Agregado el documento de requisitos de la app. (1 minute ago)
- presentacion Agregar el apartado de presentación (8 months ago)
- recursos Agregar los recursos del proyecto (8 months ago)
- src Agregar carpeta src del proyecto (8 months ago)
- LICENSE Initial commit (8 months ago)
- README.md Actualización de autores en el README (last month)

- **Documentación:** Aquí subimos toda la documentación de nuestro proyecto.
- **Presentación:** Aquí subimos los archivos que se presentarán en la defensa del proyecto
- **Recursos:** Archivos necesarios para el correcto funcionamiento de la app
- **src:** Todo el código de la app
- **LICENSE:** La licencia elegida para el repositorio
- **README.md:** Readme del proyecto

Análisis y diseño de la Base de Datos

Diagrama de entidad relación

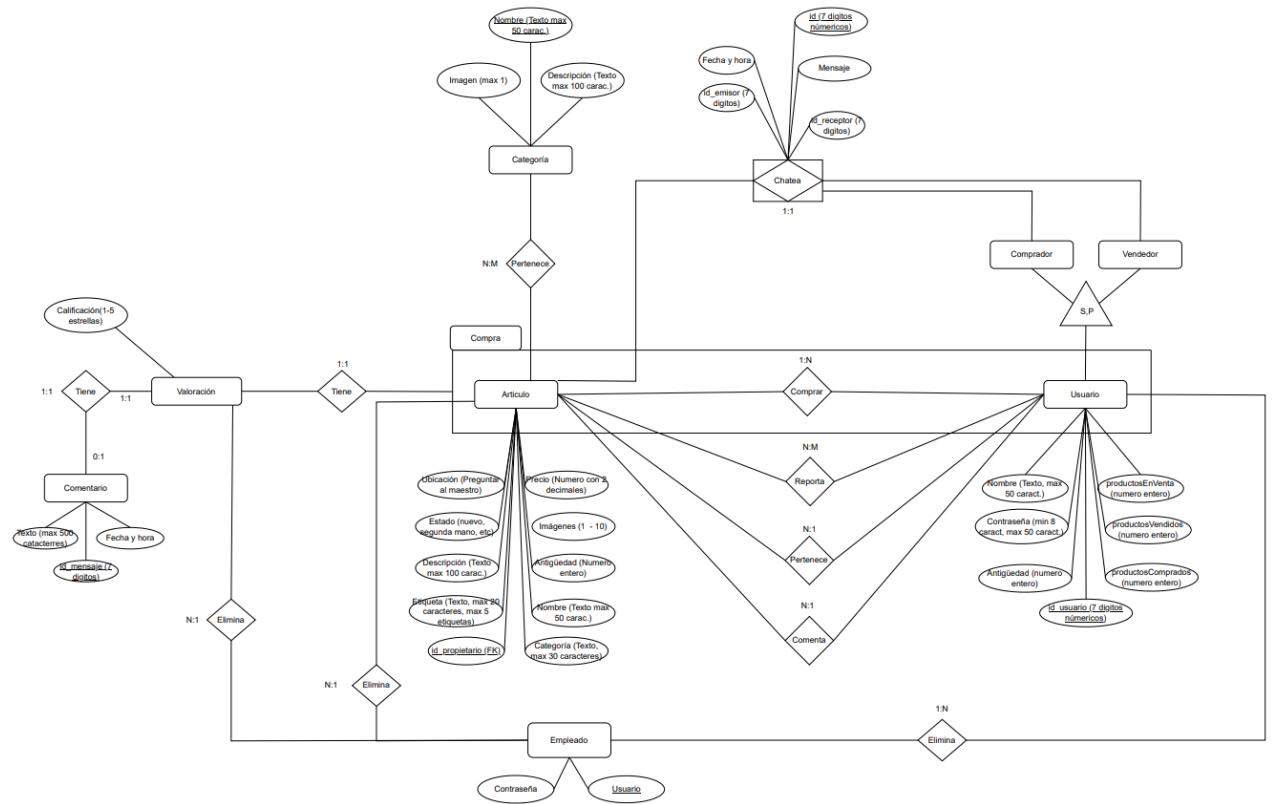


Diagrama de clases

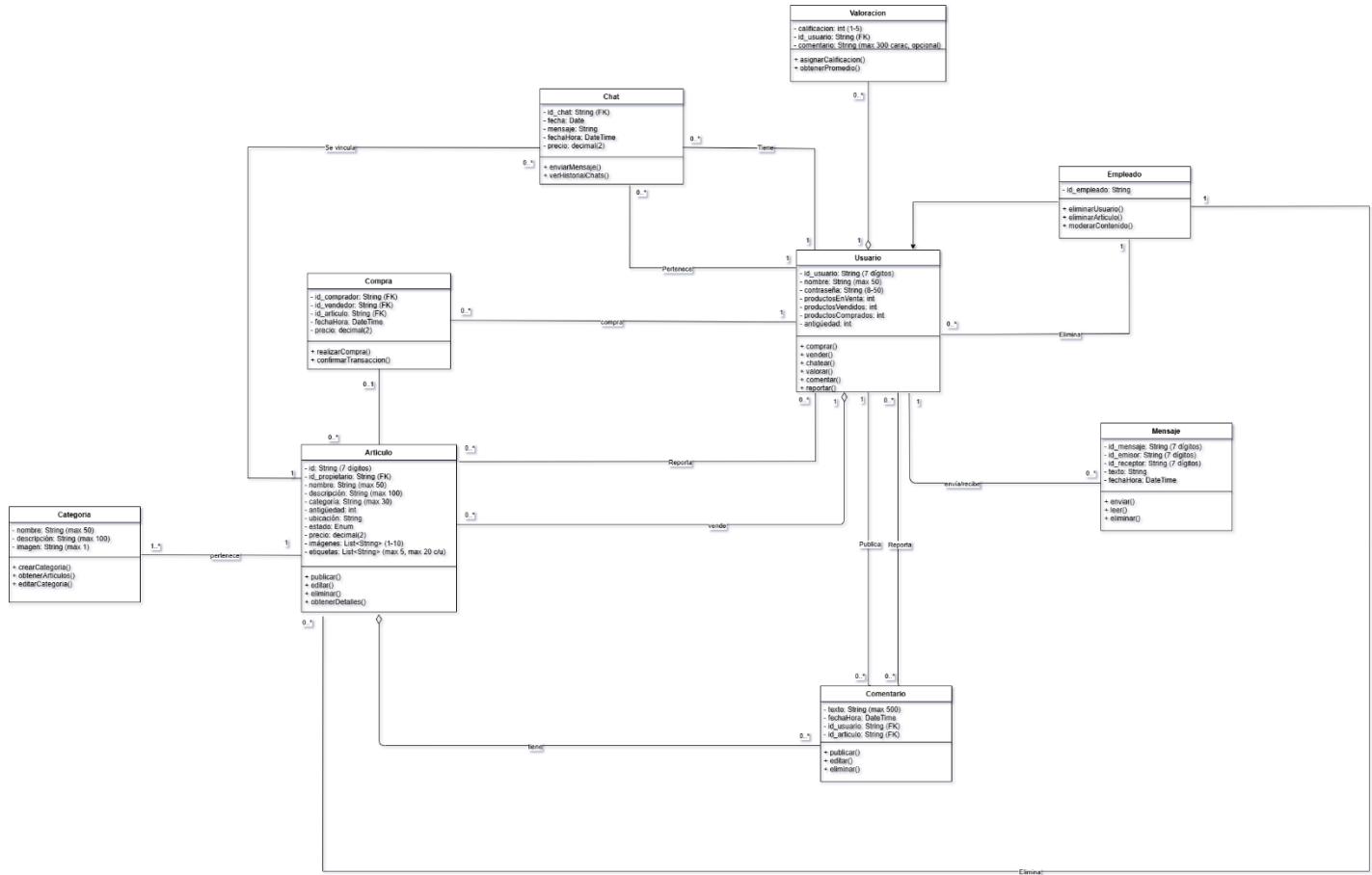
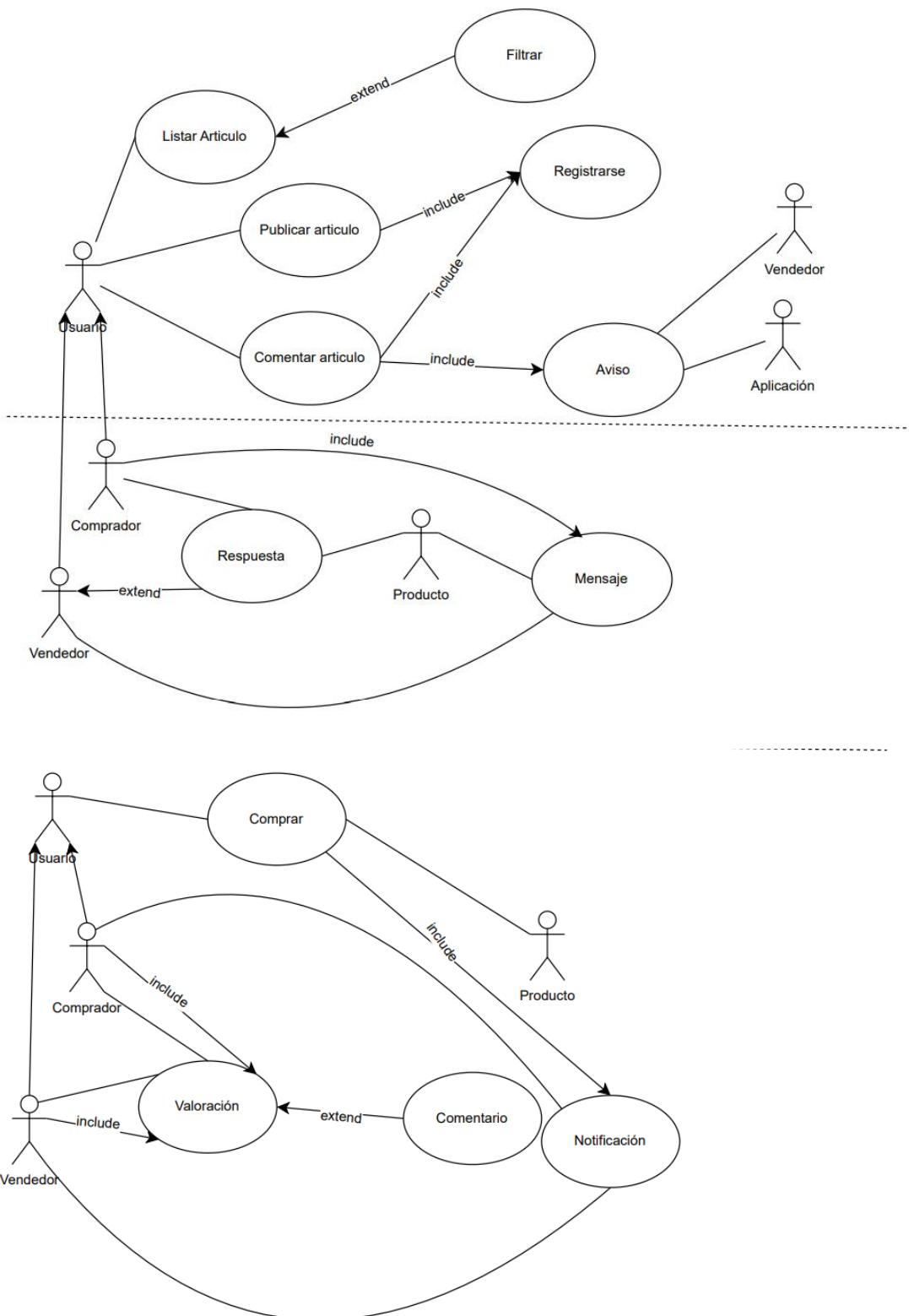
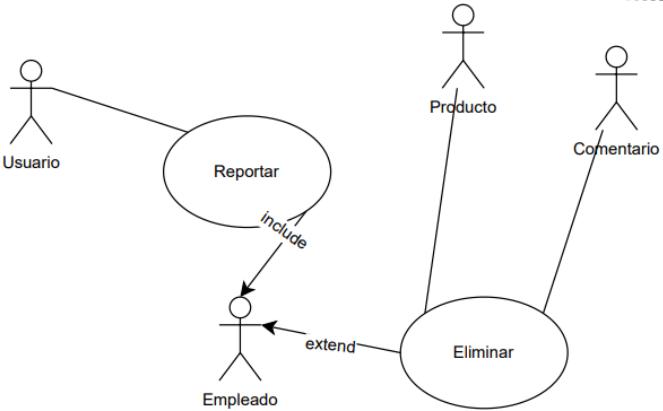


Diagrama de casos de uso





Requisitos funcionales de la app

Los requisitos funcionales son aquellos que describen qué hace el sistema, es decir, las funcionalidades que debe ofrecer.

Gestión de usuarios

- Los usuarios pueden registrarse y autenticarse en el sistema.
- Solo los usuarios registrados pueden publicar artículos o comentar artículos existentes.
- Los usuarios pueden ver perfiles de otros usuarios, incluyendo:
 - Valoración promedio (1 a 5 estrellas)
 - Comentarios recibidos
 - Número de productos en venta, vendidos y comprados
 - Antigüedad en la aplicación

Gestión de productos

- Los usuarios pueden subir productos completando los siguientes campos obligatorios:
 - Propietario
 - Categoría
 - Nombre
 - Descripción
 - Antigüedad
 - Estado (nuevo, segunda mano, etc.)
 - Etiquetas (máximo 5)
 - Imágenes (mínimo 1, máximo 10)
 - Precio
 - Ubicación
- Los usuarios pueden listar productos y filtrarlos por:
 - Categoría
 - Etiqueta
 - Nombre
 - Precio
 - Distancia
- Los usuarios pueden buscar productos específicos.
- Los usuarios pueden ver detalles de un producto.

Interacción y comunicación

- Los usuarios pueden comentar productos publicados.
- El propietario del producto recibe una notificación (email, WhatsApp, SMS, etc.) cuando recibe un comentario.
- Los usuarios pueden enviar mensajes privados a los propietarios de los productos.
- El propietario puede responder a los mensajes.
- Los mensajes estarán vinculados al producto por el que se abrió el chat.

Compra y valoración

- Los usuarios pueden comprar productos de otros usuarios.
- Cuando se realiza una compra, tanto comprador como vendedor reciben una notificación para valorar la otra parte.
- Los usuarios pueden valorar de 1 a 5 estrellas y añadir un comentario opcional.
- Las valoraciones se agregan al perfil del usuario.

Moderación y administración

- Los usuarios pueden reportar productos o comentarios ofensivos.
- Las denuncias generan una notificación a los empleados para que las gestionen.
- Los empleados pueden eliminar artículos, comentarios o usuarios que incumplan las políticas de la empresa.
- Debe existir una tabla de empleados que administra el backend (panel de administración).

Gestión de categorías

- Las categorías tendrán los siguientes campos:
 - Nombre
 - Descripción
 - Imagen

Requisitos no funcionales de la app

Los requisitos no funcionales describen cómo debe comportarse el sistema, es decir, las características de calidad, rendimiento y seguridad.

Seguridad

- Los datos de usuarios deben almacenarse de forma segura (hashing, cifrado, etc.).
- Los empleados deben tener acceso restringido al panel de administración mediante autenticación especial.
- Se deben validar los datos de entrada.
- Las comunicaciones deben realizarse mediante protocolos seguros (HTTPS).

Rendimiento

- El sistema debe permitir búsquedas y filtrados rápidos incluso con una gran cantidad de productos.
- Las imágenes de los productos deben optimizarse para mejorar el rendimiento de carga.

Fiabilidad y disponibilidad

- Debe existir un sistema de copias de seguridad periódico para proteger la información.

Usabilidad

- La interfaz debe ser intuitiva, fácil de usar y accesible tanto en web como en dispositivos móviles.
- Los procesos de registro, publicación y compra deben ser simples y guiados.

Notificaciones y comunicación

- El sistema debe permitir configurar el canal de notificación (email, WhatsApp, SMS, notificación interna).
- Las notificaciones deben enviarse en tiempo real o con mínima latencia.

Escalabilidad

- El sistema debe ser escalable, permitiendo un incremento de usuarios y productos sin afectar el rendimiento.

Mantenibilidad

- El código debe estar modularizado y documentado para facilitar futuras actualizaciones.
- Debe existir un sistema de logs para registrar errores, acciones de usuarios y empleados.

Requisitos funcionales del ERP

Los requisitos funcionales describen qué funciones debe ofrecer el sistema ERP en Odoo para gestionar las operaciones de la plataforma.

Módulo de Gestión de Usuarios

- Los usuarios pueden registrarse en el portal web del ERP o ser creados manualmente por un empleado desde Odoo.
- Los usuarios deben tener roles diferenciados:
 - **Usuario normal:** puede publicar, comprar y comentar.
 - **Empleado:** puede moderar contenido y gestionar la plataforma.
 - **Administrador:** puede acceder a todos los módulos y configuraciones.
- Los empleados pueden asignar permisos específicos a los usuarios (crear productos, ver reportes, eliminar contenido, etc.).
- El perfil del usuario mostrará:
 - Valoración promedio (1–5 estrellas).
 - Número de productos vendidos, comprados y en venta.
 - Antigüedad como usuario.

Módulo de Gestión de Productos

- Los usuarios (vendedores) pueden crear productos desde el portal o desde el backend, completando:
 - Propietario (usuario vendedor).
 - Categoría (predefinida).
 - Nombre, descripción, antigüedad, estado.
 - Etiquetas (hasta 5).
 - Imágenes (mínimo 1, máximo 10).
 - Precio.
 - Ubicación.
- Los productos se vinculan automáticamente con el usuario propietario en el modelo res.partner.
- Los usuarios pueden listar productos y filtrar por categoría, etiqueta, nombre, precio o distancia.

- Las categorías se gestionan en un modelo propio con:
 - Nombre.
 - Descripción.
 - Imagen asociada.

Módulo de Comunicación y Mensajería

- Los usuarios pueden comentar productos.
- Cuando un producto recibe un comentario, el propietario recibe una notificación (email, SMS, WhatsApp o notificación interna) usando el sistema de mensajería de Odoo.
- Se habilita un chat interno vinculado al producto, para que comprador y vendedor puedan comunicarse.
- Los mensajes quedan registrados en la ficha del producto y asociados al historial de ambos usuarios.

Módulo de Compras y Ventas

- Los usuarios pueden comprar productos desde el portal.
- Al concretar la venta, Odoo genera un pedido de venta y actualiza el inventario.
- Comprador y vendedor reciben una notificación automática para valorar la transacción.
- Las valoraciones (1 a 5 estrellas + comentario opcional) se guardan en un modelo relacionado con res.partner.
- Las valoraciones se reflejan en el perfil del usuario.

Módulo de Moderación y Administración

- Los usuarios pueden reportar productos o comentarios ofensivos.
- Las denuncias crean un ticket interno para que los empleados lo revisen.
- Los empleados pueden eliminar artículos, comentarios o usuarios desde el backend de Odoo.
- Los empleados disponen de un panel de control (dashboard) con:
 - Número de reportes abiertos.
 - Productos activos y eliminados.
 - Actividad de usuarios.

Módulo de Inventario y Categorías

- Las categorías de productos tienen:
 - Nombre.
 - Descripción.
 - Imagen.
- Los productos vendidos se marcan como "no disponibles" automáticamente.
- Odoo debe permitir reportes de ventas y actividad por categoría, vendedor y ubicación.

Módulo de Reportes y Estadísticas

- El sistema debe generar informes automáticos sobre:
 - Ventas por mes / categoría / usuario.
 - Productos más buscados.
 - Valoraciones promedio.
 - Actividad de empleados en moderación.
- Los informes pueden exportarse a Excel o PDF.

Requisitos no funcionales del ERP

Estos requisitos definen cómo debe comportarse el sistema en términos de seguridad, rendimiento, mantenimiento y usabilidad.

Seguridad

- Cada usuario accede mediante autenticación segura.
- Los datos sensibles deben almacenarse cifrados en la base de datos.
- Los empleados solo pueden acceder a los módulos según su rol y permisos.
- Todas las transacciones deben realizarse sobre HTTPS.
- Se deben registrar logs de auditoría de todas las acciones administrativas (borrados, modificaciones, etc.).

Rendimiento

- Odoo debe estar optimizado para soportar alta concurrencia de usuarios en el portal.
- Las búsquedas y filtrados de productos deben ejecutarse en menos de 2 segundos.
- Las imágenes se almacenan con optimización automática de tamaño.

Fiabilidad y Disponibilidad

- Se deben realizar copias de seguridad automáticas diarias de la base de datos y los archivos adjuntos.

Usabilidad

- La interfaz del portal debe ser intuitiva, responsive y multilenguaje.
- Las vistas del backend (empleados) deben personalizarse con menús claros y dashboards visuales.

Comunicación y Notificaciones

- El sistema debe permitir configurar canales de notificación (email, WhatsApp, SMS, notificación interna).
- Las notificaciones deben enviarse automáticamente mediante el sistema de colas de Odoo.

Escalabilidad

- El ERP debe poder escalar horizontalmente, permitiendo agregar más nodos Odoo y PostgreSQL según crezca la carga.

Mantenibilidad

- El código personalizado (módulos) debe seguir las buenas prácticas de desarrollo Odoo.
- Debe existir un entorno de pruebas para probar actualizaciones sin afectar el sistema productivo.

API REST End Points

Productos

- GET /api/producto → Devuelve todos los productos de la aplicación.
- GET /api/producto?filtro=x → Devuelve los productos filtrados por un parámetro.
- POST /api/producto → Crea un nuevo producto.
- DELETE /api/producto/{id} → Elimina un producto.

Comentarios

- POST /api/producto/{id}/comentario → Crea un comentario dentro del producto {id}.
- DELETE /api/producto/{id}/comentario{id} → Elimina un comentario.

Usuarios y autenticación

- POST /api/usuario → Registra un nuevo usuario.
- POST /api/login → Inicia sesión de usuario.
- DELETE /api/usuario/{id} → Elimina un usuario.

Valoraciones

- POST /api/usuario/{id}/valoración → Crea una valoración sobre un usuario.
- GET /api/usuario/{id}/valoración → Devuelve todas las valoraciones del usuario {id}.
- GET /api/usuario/{id}/valoración/{id} → Devuelve la valoración específica {id} del usuario {id}.

Conversaciones y mensajes

- POST /api/producto/{id}/conversación → Inicia una conversación entre comprador y vendedor dentro del producto {id}.
- GET /api/producto/{id}/conversación/{id}/mensaje → Devuelve los mensajes de la conversación {id}.

- POST /api/producto/{id}/conversación/{id}/mensaje → Envía un mensaje dentro de la conversación {id}.

Notificaciones

- POST /api/notificación → Envía una notificación.

Reportes

- POST /api/reporte → Reporta un artículo, comentario o usuario.
- GET /api/reporte → Devuelve los reportes.