



UNIVERSIDAD TECNOLÓGICA
DE LA ZONA METROPOLITANA DE GUADALAJARA

MEMORIA TÉCNICA REALIZADA EN:

Corporativo Textil JSN, S. de R.L de C.V



PROYECTO: Módulo de Gestión de Reprocesos y Reposiciones

PARA OBTENER EL GRADO DE:

Técnico Superior Universitario (TSU) en:

TECNOLOGÍAS DE LA INFORMACIÓN, ÁREA DESARROLLO DE
SOFTWARE MULTIPLATAFORMA

PRESENTADO POR:

Víctor Manuel Montaña Juanpedro

Eduardo Isaías Vázquez Solís

ASESOR INDUSTRIAL ASESOR ACADÉMICO

Jair Efraín Barragan García Lizbeth Noriega Gutiérrez

COORDINADOR DE CARRERA

Mtra. Lizbeth Noriega Gutierrez

TLAJOMULCO DE ZÚÑIGA, JALISCO, AGOSTO DEL 2025

UNIVERSIDAD TECNOLÓGICA DE LA ZONA
METROPOLITANA DE GUADALAJARA
DIRECCIÓN DE DESARROLLO Y GESTIÓN DE SOFTWARE



MÓDULO DE GESTIÓN DE REPROCESOS Y REPOSICIONES

MEMORIA TÉCNICA REALIZADA EN:

CORPORATIVO TEXTIL JSN, S. DE R.L DE C.V

PARA OBTENER EL GRADO DE:

Técnico Superior Universitario (TSU) en:

TECNOLOGÍAS DE LA INFORMACIÓN

Área, DESARROLLO DE SOFTWARE MULTIPLATAFORMA

PRESENTADO POR:

VÍCTOR MANUEL MONTAÑO JUANPEDRO

EDUARDO ISAÍAS VÁZQUEZ SOLÍS

AGOSTO 2025

Agradezco a todas las personas...

Índice general

Agradecimientos	3
1. Introducción	2
2. Antecedentes y Descripción de la Empresa	4
2.1. Ubicación	5
2.2. Misión	5
2.3. Visión	5
2.4. Organigrama	5
2.5. Historia	6
3. Problemática y Descripción del Proyecto	7
3.1. Problemática	8
3.2. Descripción del Proyecto	9
3.2.1. Objetivo General	9
3.2.2. Objetivos Específicos	9
3.2.3. Planeación	9
4. Marco Teórico	10
4.1. Sistemas ERP	11
4.1.1. Historia y evolución	11
4.1.2. Las Raíces en la Manufactura: MRP y MRP II	11
4.1.3. Los Motores del Cambio en los Años 90	12
4.2. Antecedentes y trabajos previos:	13

	1
4.2.1. Antecedente en la Gestión de Procesos Docentes Universitarios	14
4.2.2. Antecedente en el Desarrollo de Interfaces Modernas en Odoo	14
4.3. Arquitectura y Componentes de Sistemas ERP:	15
4.3.1. Modelo de Negocio y Licenciamiento	15
4.3.2. Arquitectura Tecnológica	16
4.3.3. Modularidad y Cobertura Funcional	17
4.3.4. Flexibilidad y Personalización	17
4.4. Desarrollo de Módulos en Odoo	18
4.4.1. Estructura Fundamental de un Módulo	18
4.4.2. Modelos (Capa de Datos y Lógica)	19
4.4.3. Vistas (Capa de Presentación)	20
4.4.4. Controladores y Workflows	21
4.5. Tecnologías Usadas	21
4.5.1. Backend: Python	22
4.5.2. Base de Datos: PostgreSQL	22
4.5.3. Frontend: XML, JavaScript y OWL	22
4.5.4. Protocolos de Comunicación y Servidor	23
5. Desarrollo del Proyecto	24
5.1. Requerimientos	25
5.2. Análisis y Diseño	25
5.3. Implementación	25
5.4. Pruebas	25
6. Resultados y Conclusiones	26
6.1. Resultados	27
6.2. Conclusiones	27
A. Glosario	29

Capítulo 1

Introducción

Esta es la introducción jshdjhsdjkhfsdj kjsdlkfjskad jdhshdfsdflksk, jdfjsdhjjkdsjkh. Esta es la introducción jshdjhsdjkhfsdj kjsdlkfjskad jdhshdfsdflksk, jdfjsdhjjkdsjkh. Esta es la introducción jshdjhsdjkhfsdj kjsdlkfjskad jdhshdfsdflksk, jdfjsdhjjkdsjkh. Esta es la introducción jshdjhsdjkhfsdj kjsdlkfjskad jdhshdfsdflksk, jdfjsdhjjkdsjkh.

Esta es la introducción jshdjhsdjkhfsdj kjsdlkfjskad jdhshdfsdflksk, jdfjsdhjjkdsjkh.

Capítulo 2

Antecedentes y Descripción de la Empresa

2.1. Ubicación

La empresa Corporativo Textil JSN, S. de R.L de C.V se encuentra ubicada en C. Flaviano Ramos Sur 37, Centro, 45640 Tlajomulco de Zúñiga, Jal. El mapa puede ser consultado en la Figura 2.1



Figura 2.1: Ubicación de la empresa

2.2. Misión

Proyectar la identidad empresarial de sus clientes a través de la vestimenta de sus colaboradores, considerando su comodidad. Aplica diseños en tendencia, utiliza tecnologías actualizadas y apoya a la fuerza laboral conformada por familias mexicanas.

2.3. Visión

Consolidar una empresa transnacional en la transformación integral de textiles, estandarizando y digitalizando procesos en sus diferentes unidades de negocio para el año 2030.

2.4. Organigrama

A continuación en la figura 2.2 se presenta el organigrama de la empresa.

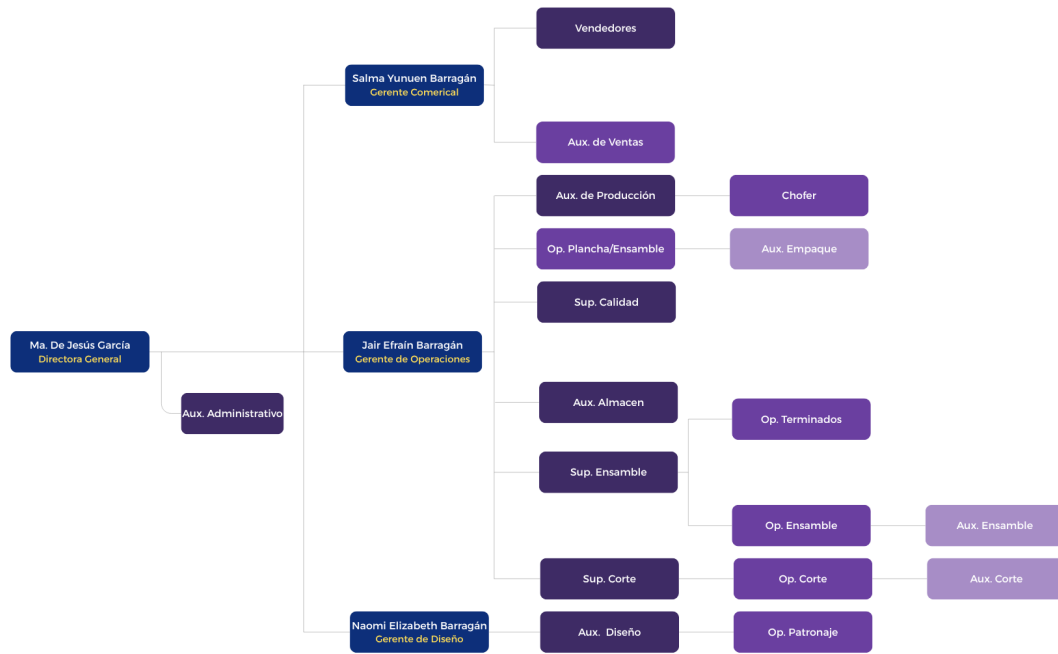


Figura 2.2: Organigrama de Textil JSN

2.5. Historia

En Corporativo Textil JSN, diseñamos y creamos prendas y textiles para hoteles y empresas de hospitalidad. Nos dedicamos a que los colaboradores de nuestros clientes estén presentables y cómodos todos los días, priorizando su seguridad e integridad. Esto lo logramos incorporando tecnologías en nuestras prendas que además aumentan su vida útil.

Capítulo 3

Problemática y Descripción del Proyecto

3.1. Problemática

La empresa **Corporativo Textil JSN**, dedicada a la fabricación de uniformes corporativos, enfrenta actualmente diversas dificultades en el control, trazabilidad y gestión eficiente de los *reprocesos y reposiciones* de piezas defectuosas o dañadas durante la producción. Aunque se utiliza el sistema ERP (Enterprise Resource Planning) Odoo, este proceso específico no se encuentra digitalizado ni debidamente integrado al entorno operativo, generando así múltiples problemáticas en distintas áreas del flujo productivo y administrativo.

Las principales problemáticas detectadas son las siguientes:

1. **Falta de trazabilidad:** No existe un registro digital estandarizado de qué piezas fueron reprocesadas, cuándo, por quién, ni en qué etapa del proceso ocurrieron los daños.
2. **Procesos manuales:** Actualmente, la información relacionada con daños, responsables y tiempos de corrección se gestiona mediante hojas físicas, lo cual incrementa el riesgo de pérdida de información y errores humanos.
3. **Descontrol en la gestión por áreas:** Las distintas áreas involucradas en la producción (Diseño, Corte, Ensamble, Planchado, entre otras) operan de forma aislada en cuanto al manejo de reprocesos, sin un sistema común que garantice la continuidad y visibilidad del flujo entre ellas.
4. **Dificultad para calcular tiempos y costos:** No se cuenta con una herramienta que permita realizar cálculos automáticos del tiempo invertido en cada reproceso, ni estimaciones precisas del costo de materiales e insumos reinvertidos.

Ante este panorama, se vuelve indispensable el desarrollo de un módulo personalizado en Odoo que permita sistematizar la gestión de reprocesos y reposiciones, garantizando control, transparencia, eficiencia operativa, y toma de decisiones basada en información precisa y actualizada.

3.2. Descripción del Proyecto

El proyecto se basa en el diseño y desarrollo de un módulo personalizado en Odoo que permita gestionar de forma centralizada, digital y trazable los procesos de reposición y reproceso de piezas defectuosas en la cadena de producción de la empresa Corporativo Textil JSN. El modulo heredará las funcionalidades básicas de Odoo, pero se adaptará específicamente a las necesidades de la empresa para optimizar el flujo de trabajo y mejorar la eficiencia operativa. Tomando como base los módulos de Manofactura y Reparaciones, se implementarán las siguientes características: Lleve el registro de solicitudes de reproceso y reposición, asignación de responsabilidades por área, cálculo automático de tiempos y costos, y generación de reportes históricos.

3.2.1. Objetivo General

- Desarrollar e implementar un módulo en el ERP Odoo que permita gestionar digitalmente las solicitudes de reproceso y reposición, asegurando trazabilidad, control por área y generación de reportes históricos.

3.2.2. Objetivos Específicos

- Permitir el registro eficiente de solicitudes con folios y datos clave de producción.
- Asignar y seguir responsabilidades por área en el flujo del reproceso.
- Automatizar el cálculo de tiempos de reproceso y sus costos estimados.
- Generar reportes que ayuden en el análisis.

3.2.3. Planeación

Capítulo 4

Marco Teórico

4.1. Sistemas ERP

ERP es una herramienta, un sistema integral que centraliza y gestiona de manera coordinada los distintos procesos y recursos que se manejan en el ámbito empresarial, desde el areas financieras y de recursos humanos hasta areas más especializadas como la logística y la producción, ordenando de esta manera toda la información de la empresa en un mismo lugar.

Durante los últimos años el mercado tecnológico ha avanzado de manera extraordinaria implantándose en otros ámbitos como los negocios, añadiéndole un nuevo valor a la información que se vuelve necesaria para el análisis y la toma de decisiones. Este es el motivo por el cual cada vez más empresas están considerando la implementación de un sistema ERP en su proceso de producción a manera de sistema central para el manejo y gestión de sus recursos, entrando de esta forma a competir en el siempre creciente e innovador mercado tecnológico e informático.

4.1.1. Historia y evolución

En tiempos pasados las empresas se manejaban en areas aisladas que trabajaban con sistemas denominados "silos." "stovepipes" dependiendo de cada area funcional dentro de una misma compañía, es decir, dentro de una empresa se podía tener un sistema propio para area de ventas, producción, administración, etc. con su propia instalación de Hardware, Software y métodos de procesamiento de datos e información. Por esto se les denominaban "silos", pues tal como hacen los lugares utilizados para almacenar granos en las granjas, cada area tenía su propio espacio controlado.

La transición desde este modelo de silos hacia los sistemas integrados que hoy conocemos como ERP fue una evolución gradual, impulsada tanto por la necesidad del negocio como por avances tecnológicos significativos a lo largo de varias décadas.

4.1.2. Las Raíces en la Manufactura: MRP y MRP II

Los orígenes de los sistemas ERP se encuentran en el sector de la manufactura. Durante los años 60, con el aumento de la capacidad de cómputo, surgieron los primeros sistemas

de Planificación de Requerimientos de Materiales (MRP, por sus siglas en inglés). Estos sistemas permitían a las empresas calcular de manera más eficiente la cantidad de materias primas necesarias y cuándo ordenarlas para cumplir con los planes de producción. El objetivo principal era asegurar la disponibilidad de materiales y reducir los costos de inventario.

En la década de los 80, estos sistemas evolucionaron hacia la Planificación de Recursos de Manufactura (MRP II). A diferencia del MRP original, el MRP II integraba no solo los materiales, sino también los recursos de la planta y la mano de obra, incorporando funcionalidades de planificación de capacidad y finanzas. MRP II fue un paso crucial hacia la integración, ya que comenzaba a coordinar diferentes áreas del ciclo de producción dentro de un marco unificado.

4.1.3. Los Motores del Cambio en los Años 90

El verdadero nacimiento de los sistemas ERP tal como los conocemos hoy ocurrió a principios de los años 90. Esta transición fue posible gracias a la confluencia de tres factores clave:

- **Avances Tecnológicos:** La Ley de Moore describía el crecimiento exponencial del poder de cómputo, lo que hacía que el hardware fuera cada vez más potente y asequible. Crucialmente, la industria se movió de la costosa arquitectura de mainframe a la más flexible y distribuida arquitectura cliente-servidor. Al mismo tiempo, el desarrollo y madurez de los sistemas de gestión de bases de datos relacionales (RDBMS) proporcionaron la tecnología necesaria para construir una base de datos centralizada y compartida.
- **La Visión de la Integración:** En el ámbito empresarial, las duras condiciones económicas de finales de los 80 y principios de los 90 impulsaron a las empresas a buscar una mayor eficiencia. Conceptos como la Reingeniería de Procesos de Negocio (BPR), popularizados por autores como Michael Hammer, promovieron la idea de rediseñar radicalmente los procesos de negocio en lugar de simplemente automatizar las tareas existentes. Esto creó el ambiente perfecto para un software que pudiera soportar procesos de negocio integrados de extremo a extremo.

- El Surgimiento de SAP: La compañía alemana SAP (Sistemas, Aplicaciones y Productos en Procesamiento de Datos), fundada en 1972 por ex-empleados de IBM, fue la pionera en materializar esta visión. En 1992, SAP lanzó su revolucionario software R/3, diseñado específicamente para la arquitectura cliente-servidor. R/3 permitía a las empresas conectar sus diferentes áreas funcionales a una única base de datos subyacente, ofreciendo una solución integrada en tiempo real. Esto sentó las bases para el mercado ERP moderno.
- El Catalizador del Y2K y la Expansión del Mercado: Un evento inesperado aceleró masivamente la adopción de los sistemas ERP a finales de los 90: el problema del año 2000 (Y2K). Muchas empresas utilizaban sistemas heredados ("legacy systems") que almacenaban el año con solo dos dígitos (ej. "98" para 1998). La perspectiva de un fallo masivo de sistemas al llegar el año 2000 obligó a las organizaciones a tomar una decisión: invertir enormes sumas en reparar estos sistemas obsoletos o reemplazarlos por completo con una solución moderna e integrada. Muchas eligieron la segunda opción, lo que provocó un auge sin precedentes en las ventas de ERP. Durante este periodo, competidores como Oracle y PeopleSoft (posteriormente adquirido por Oracle) también ganaron una cuota de mercado significativa.

A partir del nuevo milenio, los sistemas ERP continuaron evolucionando, incorporando funcionalidades de CRM (Customer Relationship Management) y SCM (Supply Chain Management), y adaptándose a nuevas tecnologías como el cómputo en la nube, el software como servicio (SaaS) y la movilidad, dando paso a la era de los ERP post-modernos, más flexibles, accesibles y modulares.

4.2. Antecedentes y trabajos previos:

Para contextualizar la presente investigación y fundamentar su relevancia, es imprescindible revisar trabajos anteriores que aborden problemáticas similares o que exploren el uso de las tecnologías propuestas. A continuación, se analizan dos trabajos de investigación recientes que sirven como antecedentes directos y técnicos para el desarrollo de esta propuesta:

un proyecto de mejora de la gestión docente en una universidad pública utilizando Odoo y un desarrollo técnico de una aplicación específica sobre la misma plataforma.

4.2.1. Antecedente en la Gestión de Procesos Docentes Universitarios

Un antecedente de alta relevancia es el trabajo de investigación de maestría titulado *"Propuesta de mejora para el proceso de gestión docente de la Universidad Nacional Autónoma de Chota, 2024"* presentado por Pretell Cruzado (2024) en la Escuela de Posgrado Newman, Perú. Este estudio aborda una problemática directamente análoga a la de la presente tesis, explorando la situación de una universidad pública con el objetivo de proponer una solución tecnológica que mejore sus procesos de gestión docente.

El autor identifica que, en el contexto de las universidades públicas, existen carencias y limitaciones en los procesos académicos, especialmente los referidos a la gestión docente. Estos procesos incluyen el reclutamiento y selección, la inducción, la formación continua, la evaluación del desempeño y el reconocimiento docente. La investigación de Pretell Cruzado tuvo como resultado el diseño y la validación del software ERP Odoo v.17 - edición Community como la solución tecnológica a implementar para sistematizar dichos procesos. La conclusión de su trabajo afirma que una solución de este tipo ofrece beneficios tangibles en el marco de la transformación digital, como un mejor acceso en línea a la información, mayor transparencia y la optimización general del proceso de gestión docente.

Este trabajo es fundamental como antecedente, ya que valida tanto la problemática abordada como la elección de Odoo ERP como una herramienta tecnológica viable y pertinente para mejorar la gestión docente en el ámbito universitario público.

4.2.2. Antecedente en el Desarrollo de Interfaces Modernas en Odoo

Desde una perspectiva técnica, el trabajo de fin de grado *"Desarrollo de una Aplicación en el sistema de código abierto Odoo para la Gestión de Residencias"* de Tort Carrillo (2024) para la Escola Tècnica Superior d'Enginyeria Industrial de Barcelona, sirve como un excelente

referente. Aunque el dominio de aplicación es distinto (residencias geriátricas), su enfoque se centra en el desarrollo de una interfaz de usuario (UI) moderna sobre la plataforma Odoo.

El objetivo principal del proyecto fue la creación de una interfaz de usuario intuitiva y adaptable, utilizando específicamente el framework de JavaScript **Odoo Web Library (OWL)**. OWL es descrito como una herramienta moderna y eficiente para la creación de aplicaciones web reactivas dentro del ecosistema de Odoo. Este antecedente es técnicamente relevante porque demuestra la capacidad de Odoo para ir más allá de sus interfaces estándar, permitiendo la construcción de soluciones visuales a medida para necesidades específicas.

Asimismo, el trabajo subraya la importancia de la integración del nuevo módulo con otros módulos preexistentes en Odoo para asegurar una gestión correcta del flujo de información. Por lo tanto, este estudio valida el uso de las herramientas de frontend más modernas de Odoo (OWL) para la creación de aplicaciones especializadas y destaca la arquitectura modular e integrada de la plataforma.

4.3. Arquitectura y Componentes de Sistemas ERP:

La arquitectura de un sistema ERP define su estructura técnica, su funcionamiento y su capacidad de adaptación. Para comprender la diversidad de enfoques en el mercado, este capítulo realiza un análisis comparativo entre dos sistemas paradigmáticos: SAP, como representante del modelo de ERP tradicional y líder del mercado, y Odoo, como exponente de los sistemas ERP modernos de código abierto.

4.3.1. Modelo de Negocio y Licenciamiento

El modelo de negocio dicta cómo una empresa adquiere, implementa y mantiene su sistema ERP, lo cual tiene un impacto directo en el costo total de propiedad (TCO).

SAP (Modelo Propietario Tradicional) SAP opera bajo un modelo de software propietario y de código cerrado. Históricamente, esto implica costos significativos asociados a la adquisición de licencias de software, honorarios de consultores certificados para la implementación y contratos de mantenimiento anual para recibir soporte y actualizaciones.

Este modelo, aunque costoso, ofrece a las grandes corporaciones un ecosistema robusto y un soporte centralizado.

Odoo (Modelo *Open Core*) En contraposición, Odoo opera bajo un modelo de negocio conocido como “Open Core”. Este modelo combina lo mejor del software de código abierto con un producto comercial respaldado por la empresa. Se manifiesta en dos versiones principales:

1. **Odoo Community:** Es la versión de código abierto, completamente gratuita para descargar, usar y modificar. No tiene costos de licencia y es mantenida gracias a una comunidad global de desarrolladores.
2. **Odoo Enterprise:** Es la versión comercial que funciona bajo un modelo de suscripción anual por usuario y por aplicación. Esta versión incluye módulos avanzados, herramientas de migración y el soporte directo del fabricante.

Este modelo dual ofrece un punto de entrada de bajo costo para PyMEs y una ruta de escalamiento hacia una solución más robusta y soportada.

4.3.2. Arquitectura Tecnológica

La pila tecnológica (o *stack*) define los componentes de software y lenguajes sobre los que se construye el sistema.

SAP (Arquitectura Cliente-Servidor y Lenguaje Propietario) El éxito de SAP se fundamentó en su arquitectura cliente-servidor de tres capas: presentación (SAP GUI), aplicación y base de datos. El desarrollo y la personalización se realizan utilizando su propio lenguaje de programación patentado, **ABAP (Advanced Business Application Programming)**.

Odoo (Arquitectura Web y Pila de Código Abierto) Odoo está construido sobre una pila tecnológica moderna y de código abierto, diseñada para ser nativa de la web. Su arquitectura también sigue un modelo de tres capas:

1. **Base de Datos:** Utiliza exclusivamente **PostgreSQL**.

2. **Servidor (Lógica):** El núcleo está escrito en **Python** y cuenta con un potente **ORM (Mapeo Objeto-Relacional)** que facilita el desarrollo.
3. **Cliente (Presentación):** La interfaz es un cliente web moderno donde las vistas se definen con **XML** y la interactividad con **JavaScript**.

4.3.3. Modularidad y Cobertura Funcional

SAP (Módulos Funcionales Clásicos) SAP estructura su sistema en un conjunto de módulos funcionales altamente integrados pero bien definidos, que corresponden a las áreas clásicas de una gran empresa: **SD** (Ventas y Distribución), **MM** (Gestión de Materiales), **PP** (Planificación de la Producción), **FI** (Contabilidad Financiera), **CO** (Controlling) y **HR** (Recursos Humanos).

Odoo (Enfoque basado en “Apps”) Odoo adopta un enfoque hiper-modular, organizando su funcionalidad en “Apps” que se pueden instalar o desinstalar fácilmente. Las aplicaciones principales se corresponden directamente con los módulos clásicos de SAP:

- La App **Ventas** de Odoo es el análogo funcional de **SAP SD**.
- Las Apps **Inventario** y **Compras** cubren las funciones de **SAP MM**.
- La App **Fabricación** de Odoo corresponde al módulo **SAP PP**.
- La App **Contabilidad** integra funcionalidades de **SAP FI y CO**.

La gran diferencia radica en su ecosistema, donde existen miles de aplicaciones adicionales para cubrir necesidades muy específicas.

4.3.4. Flexibilidad y Personalización

SAP (Configuración vs. Personalización) SAP distingue claramente entre **configuración** (ajustes mediante herramientas integradas) y **personalización** (modificar el código ABAP), siendo esta última una tarea compleja, costosa y que puede dificultar las actualizaciones del sistema.

Odoo (Flexibilidad a través del Código Abierto) La flexibilidad de Odoo es uno de sus principales argumentos de venta. La práctica estándar no es modificar el código del núcleo, sino **crear módulos personalizados** que se instalan sobre el sistema base. Estos módulos pueden añadir o alterar funcionalidades de forma segura, asegurando que el núcleo del sistema pueda ser actualizado sin afectar las personalizaciones.

4.4. Desarrollo de Módulos en Odoo

Toda la funcionalidad en Odoo está organizada en **módulos**. Un módulo es un directorio que agrupa código de Python, archivos XML, imágenes y otros recursos para añadir o modificar una funcionalidad específica del sistema. El desarrollo en Odoo, por lo tanto, consiste fundamentalmente en la creación o extensión de estos módulos. Esta arquitectura no solo garantiza que las personalizaciones se mantengan organizadas y separadas del código fuente del núcleo, facilitando el mantenimiento y las actualizaciones, sino que también permite que el sistema sea extremadamente escalable, pudiendo activar o desactivar funcionalidades completas según las necesidades del negocio.

4.4.1. Estructura Fundamental de un Módulo

Cada módulo de Odoo es un directorio cuyo nombre técnico es el que se utilizará para referenciarlo. Dentro de él, se encuentran archivos y subcarpetas que organizan sus componentes por convención. Una estructura completa suele incluir:

- **__manifest__.py**: Es el archivo descriptor del módulo y el único estrictamente obligatorio. Contiene un diccionario de Python con metadatos clave como el nombre del módulo (**name**), versión (**version**), autor (**author**), categoría funcional (**category**), un resumen de su propósito (**summary**), y, de forma crucial, sus dependencias (**depends**), que aseguran que otros módulos necesarios se instalen primero. También define los archivos de datos, vistas y seguridad a cargar.
- **models/**: Contiene los archivos de Python (**.py**) que definen los modelos de datos y su lógica de negocio. Es el corazón del módulo.

- **views/**: Contiene los archivos XML (`.xml`) que definen la interfaz de usuario: las vistas, los menús que las invocan y las acciones que las conectan.
- **controllers/**: Alberga los archivos Python que manejan las peticiones web. Son esenciales para crear páginas para el sitio web de Odoo o para definir puntos de acceso (endpoints) para APIs RESTful personalizadas.
- **security/**: Contiene los archivos que definen los permisos de acceso. Principalmente el archivo `ir.model.access.csv`, que especifica qué grupos de usuarios pueden realizar operaciones de lectura, escritura, creación o borrado en cada modelo.
- **data/**: Contiene archivos XML utilizados para cargar datos iniciales o de configuración que el módulo necesita para funcionar, como secuencias, categorías de productos o plantillas de correo.
- **static/**: Contiene los archivos estáticos (CSS, Sass, JavaScript, imágenes, fuentes) que se usarán en la capa de presentación, tanto en el backend como en el frontend del sitio web.

4.4.2. Modelos (Capa de Datos y Lógica)

Los modelos son la columna vertebral de Odoo. Son clases de Python que heredan de `models.Model` y utilizan el **ORM (Mapeo Objeto-Relacional)** de Odoo para mapear cada clase a una tabla en la base de datos PostgreSQL y cada instancia de la clase a un registro en esa tabla. El ORM implementa un patrón de tipo *Active Record*, donde los objetos llevan consigo tanto los datos como los métodos para operar sobre ellos. Esta capa es responsable de:

1. **Definición de la estructura de datos:** Se definen los campos como atributos de la clase. Odoo ofrece una gran variedad de tipos de campo, desde los simples como `Char`, `Integer`, `Float`, `Date`, `Datetime`, `Boolean` y `Html`, hasta los relacionales que son clave para un ERP: `Many2one` (para claves foráneas, como vincular una factura a un cliente), `One2many` (la relación inversa, como ver todas las facturas de un cliente) y `Many2many` (para relaciones múltiples, como asignar varias etiquetas a un producto).

2. **Implementación de la lógica de negocio:** Se escriben métodos en Python decorados para interactuar con el ORM. Decoradores como `@api.depends` permiten crear campos cuyos valores se calculan dinámicamente a partir de otros, mientras que `@api.constrains` define reglas de validación de datos a nivel de base de datos para garantizar la consistencia.
3. **Herencia de Modelos:** Una de las características más potentes de Odoo es su mecanismo de herencia, que permite modificar modelos existentes de forma no intrusiva. Se puede extender un modelo existente para añadirle nuevos campos, modificar los atributos de campos existentes o sobrescribir sus métodos para adaptar su comportamiento a nuevas necesidades, todo sin tocar el código del módulo original.

4.4.3. Vistas (Capa de Presentación)

Las vistas en Odoo, definidas en **XML**, son la forma de presentar los datos de los modelos al usuario. El motor de Odoo interpreta estos archivos XML para generar dinámicamente las vistas HTML interactivas, utilizando un motor de plantillas llamado **QWeb**. Las vistas se conectan con los modelos a través de **Acciones** (`ir.actions.act_window`), que son registros en la base de datos que definen qué vistas mostrar para un modelo particular. Los tipos de vista principales son:

- **Vistas de Formulario (form):** La vista detallada para un único registro. Su estructura puede ser altamente personalizada con elementos como pestañas, grupos y botones que ejecutan métodos del modelo.
- **Vistas de Lista (tree):** Muestran múltiples registros en una tabla. Permiten la edición en línea y la visualización resumida de la información.
- **Vistas Kanban (kanban):** Presentan los registros como tarjetas organizadas en columnas, ideal para visualizar flujos de trabajo basados en etapas (por ejemplo, un pipeline de ventas).
- **Vistas de Búsqueda (search):** Definen los filtros, campos de búsqueda y opciones de agrupación (`^agrupar por`) que el usuario puede utilizar para analizar los datos.

4.4.4. Controladores y Workflows

Los **Controladores** son el mecanismo para crear lógica de servidor que responda a peticiones web. Heredan de la clase `http.Controller` y sus métodos se asocian a rutas (URLs) mediante un decorador. Esto permite crear desde simples páginas web con contenido dinámico hasta complejos puntos de acceso (endpoints) de API que devuelven datos en formato JSON.

El concepto de **Workflow** ha evolucionado. En versiones antiguas de Odoo (anteriores a la 8), existía un complejo sistema de workflows basado en XML que definía los flujos de documentos. Este sistema ha sido reemplazado por un enfoque más simple y flexible implementado directamente en Python. Hoy en día, un flujo de trabajo se modela típicamente con un campo de tipo `Selection` llamado `state` en el modelo (ej: 'borrador', 'enviado', 'aprobado', 'hecho'). La transición entre estados es gestionada por métodos del modelo (ej: `def action_confirm()`) que son llamados por botones en las vistas. Estos métodos contienen toda la lógica de negocio: cambiar el estado, crear otros registros, enviar correos, etc.

4.5. Tecnologías Usadas

Un concepto clave en la oferta de los sistemas ERP modernos es el de Software-as-a-Service (SaaS). Este modelo consiste en que un proveedor ofrece acceso a aplicaciones de software a través de internet, eliminando la necesidad de que el usuario descargue, instale o actualice programas en su propio equipo. Odoo ERP es un claro exponente de los programas SaaS, los cuales son particularmente útiles para optimizar la actividad en entidades de investigación y educación, permitiendo una transformación digital acelerada. Las ventajas de este enfoque incluyen ahorros significativos en costos de infraestructura, mantenimiento y seguridad.

Odoo se apoya en una pila tecnológica (*stack*) cohesiva y bien definida, basada casi en su totalidad en tecnologías de código abierto. Esta elección deliberada no solo reduce los costos de licenciamiento, sino que también permite a Odoo beneficiarse de la innovación y la seguridad impulsadas por comunidades globales de desarrolladores.

4.5.1. Backend: Python

El lenguaje de programación principal para todo el desarrollo del backend en Odoo es **Python**. La elección de Python es estratégica debido a su sintaxis limpia y legible, que reduce los costos de mantenimiento, y a su vasto ecosistema de librerías de terceros. Odoo aprovecha varias de estas librerías, como `lxml` para el procesamiento de alta velocidad de los archivos XML de las vistas y `psycopg2` para la comunicación con la base de datos. El propio framework de Odoo provee toda la infraestructura necesaria, incluyendo el ORM, el motor de plantillas y el servidor web.

4.5.2. Base de Datos: PostgreSQL

Odoo utiliza exclusivamente **PostgreSQL** como su sistema de gestión de bases de datos (SGBD). Esta no es una simple preferencia, sino un requisito técnico estricto. La razón es que Odoo está profundamente integrado con PostgreSQL y utiliza muchas de sus características avanzadas que no son estándar en todos los SGBD, como esquemas transaccionales, tipos de datos específicos, funciones y operadores avanzados, y un robusto sistema de bloqueos a nivel de fila (*row-level locking*) para gestionar la concurrencia de operaciones, algo esencial en un sistema ERP.

4.5.3. Frontend: XML, JavaScript y OWL

La capa de presentación que ve el usuario es una amalgama de tecnologías web estándar y un framework propio:

- **XML (Extensible Markup Language):** Su rol es puramente declarativo. Se utiliza para definir la estructura de las vistas, los menús, las acciones y los registros de datos. Odoo lee estos archivos y los almacena en la base de datos para luego renderizarlos dinámicamente.
- **JavaScript:** Es el motor de toda la interactividad en el navegador. Odoo ha utilizado diferentes frameworks a lo largo de su historia, pero su versión más moderna se basa en **OWL (Odoo Web Library)**.

- **OWL:** Es el framework de JavaScript propio de Odoo. Inspirado en frameworks modernos como React y Vue.js, OWL es un sistema de componentes declarativo, reactivo y de alto rendimiento, diseñado específicamente para las necesidades complejas de una interfaz de aplicación de negocio.
- **CSS y Sass:** Para el diseño y la apariencia visual, Odoo utiliza hojas de estilo en cascada (CSS). Se apoya en el preprocesador **Sass** para escribir código CSS de forma más estructurada y mantenible, utilizando variables, mixins y otras funcionalidades avanzadas.

4.5.4. Protocolos de Comunicación y Servidor

La comunicación entre el servidor Odoo y el mundo exterior se realiza a través de protocolos estándar. El servidor web principal que atiende las peticiones está integrado en el propio Odoo y se basa en la librería **Werkzeug** de Python. Para la integración con sistemas de terceros, Odoo expone su API a través de dos protocolos:

- **XML-RPC:** Es el protocolo histórico y más robusto, soportado desde las primeras versiones.
- **JSON-RPC:** Es una alternativa más moderna, ligera y fácil de consumir desde aplicaciones web basadas en JavaScript.

Ambas APIs están securizadas y requieren autenticación, respetando los permisos y las reglas de acceso definidos en la capa de seguridad de Odoo.

Capítulo 5

Desarrollo del Proyecto

5.1. Requerimientos

5.2. Análisis y Diseño

5.3. Implementación

5.4. Pruebas

Capítulo 6

Resultados y Conclusiones

6.1. Resultados

6.2. Conclusiones

Bibliografía

- [1] Dura, C. C., Drigă, I., and Iordache, A. M. M. (2022). Software-as-a-service programs and project management: A case study on odoo erp. 373:00037.
- [2] Monk, E. and Wagner, B. (2008). *Concepts in Enterprise Resource Planning*.
- [3] Odoo S.A. (2024). Odoo 17.0 developer documentation. <https://www.odoo.com/documentation/18.0/index.html>. Consultado el 9 de junio de 2025.
- [4] Pretell Cruzado, R. J. (2024). Propuesta de mejora para el proceso de gestión docente de la universidad nacional autónoma de chota, 2024. Trabajo de investigación de maestría, Escuela de Posgrado Newman, Tacna, Perú.

Apéndice A

Glosario

Asesor Académico Persona encargada de regañar a los alumnos

ERP(Enterprise Resource Planning) Sistema integrado que permite coordinar y optimizar procesos empresariales clave, como finanzas, recursos humanos, logística y producción. Se trata de mucho más que una herramienta tecnológica, ya que busca alinear distintas funciones organizacionales a través de una sola plataforma (O'Brien y Marakas, 2010, como se citó en Hammouch, 2024).