



Sistemas de gestión empresarial

Piero Zavala Chira | Odoo | 09/02/2025

Contenido

Introducción	2
Tecnologías Utilizadas	2
Estructura del Documento.....	2
Organización de la Estructura.....	2
Diseño de la Solución	3
Diagramas y Mockups	3
Desarrollo Técnico	3
Funcionalidades Avanzadas.....	3
Resultados y Pruebas.....	4
Capturas de Pantalla y Código.....	4
Gestión de Tareas	4
Captura de pantalla - Vista Kanban	4
Captura de pantalla - Vista Calendario	4
Código del Modelo en Python	5
Gestión de Biblioteca.....	5
Captura de pantalla - Formulario de Préstamo	5
Código del Modelo en Python	5
Gestión de Hospital	6
Captura de pantalla - Lista de Pacientes.....	6
Captura de pantalla - Lista de Medico	6
Código del Modelo en Python	6
Gestión Educativa	7
Código del Modelo en Python	7
Conclusión	7
Mejoras Futuras	7
Repositorio y Bibliografía.....	7
Repositorio GitHub	7
Fuentes de Información	7

Introducción

El propósito de este proyecto es desarrollar y extender módulos en Odoo para diferentes dominios de gestión: tareas, biblioteca, hospital e institutos educativos. Se busca mejorar la administración y control de datos mediante vistas personalizadas y validaciones.

TECNOLOGÍAS UTILIZADAS

- Odoo 17
- Python 3
- XML para vistas
- PostgreSQL
- Interfaz web de Odoo

ESTRUCTURA DEL DOCUMENTO

1. Introducción
2. Organización de la estructura
3. Diseño de la solución
4. Desarrollo técnico
5. Funcionalidades avanzadas
6. Resultados y pruebas
7. Conclusión
8. Repositorio y bibliografía

Organización de la Estructura

El proyecto se divide en cuatro módulos clave:

1. **Gestión de tareas** (lista de tareas con vistas Kanban y Calendar)
2. **Gestión de biblioteca** (préstamos de cómics y socios)
3. **Gestión hospitalaria** (pacientes, médicos y consultas)
4. **Gestión educativa** (ciclos formativos, módulos, alumnos y profesores con seguridad implementada)

Cada módulo sigue una estructura lógica que comprende modelos, vistas y seguridad.

Diseño de la Solución

DIAGRAMAS Y MOCKUPS

Se incluyen diagramas de relación entre los modelos y bocetos de las vistas principales en Odoo para asegurar una interfaz clara y funcional.

Ejemplo de estructura de un módulo:

- **Modelos:** Definen la estructura de datos.
- **Vistas:** Permiten la interacción con los usuarios.
- **Seguridad:** Controla el acceso a los datos.
- **Menús y acciones:** Facilitan la navegación dentro del sistema.

Desarrollo Técnico

Cada módulo fue implementado en Python y XML:

- **Python** para la lógica de negocio y validaciones.
- **XML** para definir formularios, listas, vistas Kanban y calendarios.
- **CSV** para configurar la seguridad y permisos de acceso.

Se incluyeron validaciones específicas, como:

- Restricción de fechas en los préstamos de cómics.
- Relación entre médicos y pacientes en consultas hospitalarias.
- Control de acceso para directores y profesores en módulos educativos.

Funcionalidades Avanzadas

Se agregaron características adicionales para mejorar la usabilidad:

- **Validaciones de datos:** Restricciones en fechas y asignaciones.
- **Vistas interactivas:** Uso de Kanban y Calendario para facilitar la gestión.
- **Configuración de seguridad:** Permisos diferenciados para usuarios, administradores y roles específicos.

Resultados y Pruebas

Se realizaron pruebas funcionales para verificar el correcto desempeño de cada módulo:

- **Creación y edición de registros** en cada modelo.
- **Visualización de datos** mediante vistas personalizadas.
- **Validación de restricciones y seguridad** en los accesos y permisos.

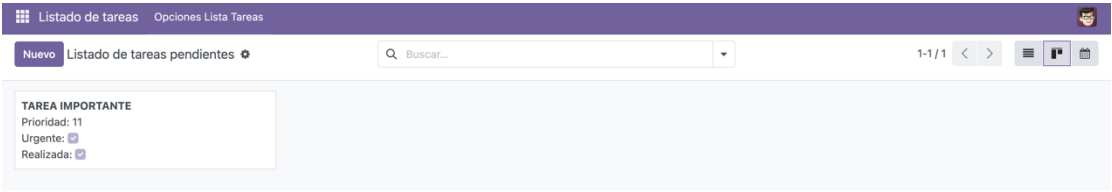
Las pruebas se ejecutaron en entornos locales y en servidores de Odoo.

Capturas de Pantalla y Código

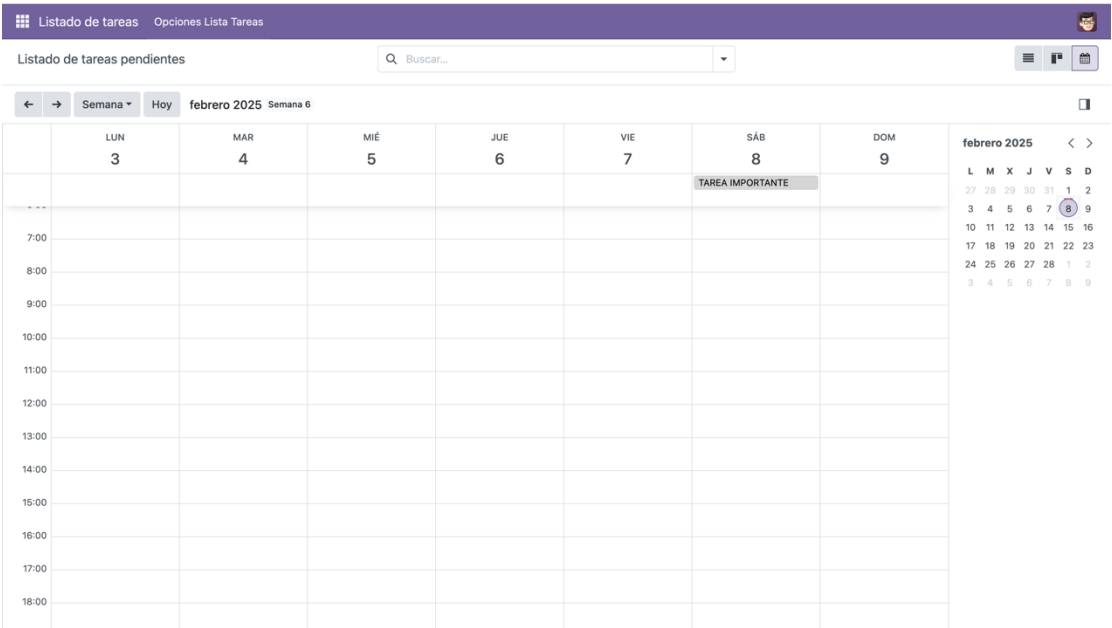
A continuación, se presentan capturas de pantalla y fragmentos de código de cada módulo:

GESTIÓN DE TAREAS

Captura de pantalla - Vista Kanban



Captura de pantalla - Vista Calendario



Código del Modelo en Python

```
class Task(models.Model):
    _name = 'task.management'
    _description = 'Gestión de Tareas'

    name = fields.Char(string='Título', required=True)
    assigned_date = fields.Date(string='Fecha Asignada',
required=True)
    state = fields.Selection([
        ('pending', 'Pendiente'),
        ('in_progress', 'En Progreso'),
        ('done', 'Completado')
    ], default='pending')
```

GESTIÓN DE BIBLIOTECA

Captura de pantalla - Formulario de Préstamo

Préstamos de cómics / biblioteca.prestamo,1

Cómico *	The Hobbit
Socio que recibe el préstamo *	gabriel
Fecha de inicio *	02/02/2024
Fecha prevista de vuelta *	29/02/2024

Código del Modelo en Python

```
class ComicLoan(models.Model):
    _name = 'library.comic.loan'
    _description = 'Préstamos de Cómics'

    comic_id = fields.Many2one('library.comic', string='Cómico', required=True)
    member_id = fields.Many2one('library.member', string='Socio',
required=True)
    date = fields.Date(string='Fecha de Préstamo', required=True)
```

GESTIÓN DE HOSPITAL

Captura de pantalla - Lista de Pacientes

Nom i cognoms	Síntomas
Piero Zavala Chira	Cagadera

Captura de pantalla - Lista de Medico

Pacient	Metge
hospital.paciente,1	hospital.medico,1

Data de consulta: 18/02/2025 19:00:00

Calendar: febrero 2025

Buttons: Cerrar, 19 : 00, Aplicar

Código del Modelo en Python

```
class HospitalPatient(models.Model):
    _name = 'hospital.patient'
    _description = 'Pacientes del Hospital'

    name = fields.Char(string='Nombre', required=True)
    last_name = fields.Char(string='Apellidos',
        required=True)
    symptoms = fields.Text(string='Síntomas')
```

GESTIÓN EDUCATIVA

Código del Modelo en Python

```
class EducationModule(models.Model):
    _name = 'education.module'
    _description = 'Módulo Formativo'

    name = fields.Char(string='Nombre del Módulo',
        required=True)
    cycle_id = fields.Many2one('education.cycle',
        string='Ciclo Formativo', required=True)
```

Conclusión

Este proyecto permitió demostrar la flexibilidad de Odoo en el desarrollo de módulos personalizados. Se implementaron buenas prácticas de estructuración de código, seguridad y validaciones.

MEJORAS FUTURAS

- Integración con API externas para mejorar la funcionalidad.
- Optimización de rendimiento en consultas a la base de datos.
- Mejora en la interfaz de usuario con diseños más dinámicos.

Repositorio y Bibliografía

REPOSITORIO GITHUB

El código fuente del proyecto está disponible en GitHub:

FUENTES DE INFORMACIÓN

- [Documentación oficial de Odoo 17](#)
- [Foros y blogs especializados en Odoo](#)