



Actividad 2 - Prototipo del Proyecto

Carlos Andrés Obando

Cod. 100177525

Corporación Universitaria Iberoamericana

Analisis y diseño de sistemas – Tatiana Cabrera

Noviembre de 2025

Introducción

Tribu SAS enfrenta la necesidad de modernizar sus procesos financieros, los cuales se realizan de forma manual y generan retrasos, errores humanos y poca visibilidad de la información. Este trabajo propone la formulación de un proyecto orientado al diseño e implementación de un módulo en Odoo 16 que automatice la facturación, integre esquemas de pago flexibles y proporcione dashboards de seguimiento en tiempo real.

Para llevar a cabo el proyecto se adopta la metodología ágil Kanban, la cual permite gestionar de manera visual y flexible el flujo de tareas, facilitando la priorización de actividades y el trabajo colaborativo.

Además de beneficiar directamente a la empresa, este proyecto constituye una oportunidad de aprendizaje personal, ya que fortalece mis competencias en el uso de Odoo, metodologías ágiles, gestión de proyectos de software y automatización de procesos financieros.

1. Contextualización de la necesidad

Tribu SAS enfrenta la necesidad de modernizar su sistema de gestión empresarial (ERP) para ofrecer esquemas de pago flexibles y mejorar la gestión financiera. Actualmente, los procesos de facturación y cobro se realizan de manera manual, lo que genera retrasos, errores humanos y poca visibilidad de la información financiera. La solución propuesta busca implementar en Odoo 16 un módulo que automatice la facturación, los esquemas de pago con cuota inicial y cuotas quincenales, integrando contabilidad y dashboards de seguimiento.

2. Planteamiento del problema

La empresa requiere una solución automatizada que permita gestionar esquemas de pago flexibles (cuota inicial y cuotas quincenales), integrados a los procesos de facturación y contabilidad en Odoo 16.

Actualmente, la gestión manual genera ineficiencias, errores y falta de información en tiempo real, lo cual afecta la toma de decisiones y el cumplimiento financiero.

3. Alcance del proyecto

El proyecto contempla el diseño, desarrollo e implementación de un módulo en Odoo 16 que automatice la facturación, la generación de asientos contables y el seguimiento de pagos. Incluye restricciones como los recursos financieros y técnicos disponibles, así como la aceptación por parte de los stakeholders basada en el cumplimiento de requisitos funcionales y la facilidad de uso.

Criterios de aceptación:

- Reducción de errores manuales,
- Automatización de pagos,
- Dashboard gráfico funcional
- Satisfacción de usuarios clave.

4. Objetivos

Objetivo General:

- Desarrollar un módulo en Odoo 16 que gestione esquemas de pago flexibles (cuota inicial + cuotas quincenales), integrando facturación, contabilidad y dashboards de monitoreo.

Objetivos Específicos:

- Extender los modelos de contacto y plazos de pago.
- Implementar lógica de pagos según niveles de usuario.
- Automatizar la generación de asientos contables.

- Diseñar un dashboard para monitoreo diario del estado de pagos.
- Realizar pruebas y ajustes para asegurar la calidad del sistema.

5. Metodología Ágil seleccionada

Se adopta la metodología ágil Kanban, enfocada en la gestión visual del flujo de trabajo mediante tableros que permiten priorizar tareas y controlar el avance en tiempo real. Esta metodología se eligió porque ofrece flexibilidad en la planificación, facilita la identificación de cuellos de botella y promueve la mejora continua sin necesidad de iteraciones fijas como en Scrum.

El tablero Kanban será implementado en Trello, donde las tareas estarán organizadas en columnas (pendiente, en proceso, y finalizado). Cada tarjeta representará una actividad con sus responsables y fechas de entrega, lo cual permitirá un seguimiento claro del estado del proyecto, mayor transparencia y una mejor colaboración entre los involucrados.

Link de tablero:

6. Justificación

La implementación de este proyecto permitirá reducir errores manuales, optimizar procesos financieros y mejorar la toma de decisiones mediante dashboards visuales. Además, aportará valor al cliente con esquemas de pago flexibles y aumentará la competitividad de Tribu SAS en el sector HORECA.

A corto plazo, la solución permitirá automatizar la facturación y la generación de asientos contables, disminuyendo la dependencia de procesos manuales y reduciendo el riesgo de errores humanos. Esto generará beneficios inmediatos para las áreas de contabilidad y ventas, al ofrecerles información más confiable y en tiempo real.

A mediano plazo, la empresa podrá gestionar esquemas de pago flexibles (cuota inicial y cuotas quincenales) de manera eficiente, fortaleciendo la relación con los clientes al ofrecerles opciones de financiamiento ajustadas a sus necesidades. Esto impactará directamente en la satisfacción del cliente y en el flujo de caja, ya que permitirá un control más preciso de los ingresos.

A largo plazo, la integración de los procesos en Odoo 16 y la disponibilidad de dashboards de seguimiento facilitarán una toma de decisiones estratégicas basada en datos, apoyando la planeación financiera y el crecimiento empresarial. Además, la empresa estará mejor posicionada frente a la competencia del sector HORECA, consolidando su imagen como organización innovadora que se adapta a las necesidades del mercado.

En cuanto a los stakeholders, los directivos contarán con información clara para la toma de decisiones; el área de contabilidad dispondrá de herramientas automatizadas que optimicen su carga operativa; el equipo de ventas podrá ofrecer mejores condiciones a los clientes; y los usuarios finales obtendrán un proceso de pago más ágil y flexible. Todo ello fortalece la confianza en el sistema, mejora la comunicación interna y asegura la aceptación del proyecto dentro de la organización.

7. Mapa de Stakeholders

- **Directivos de Tribu SAS:** Toman decisiones estratégicas.
- **Área de Ventas:** Responsable de la facturación y comunicación con clientes.
- **Área de Contabilidad:** Encargada de los registros contables y control financiero.
- **Clientes:** Usuarios finales que se benefician de los esquemas de pago.
- **Estudiantes:** Responsables de diseño e implementación.

8. Matriz de Riesgos

Riesgo	Probabilidad	Impacto	Mitigación
Retrasos en la implementación	Medio	alto	Planificación detallada y control de tiempos.
Resistencia al cambio por parte de empleados	Alto	Medio	Capacitación y comunicación.
Fallos técnicos	Medio	Alto	Pruebas unitarias

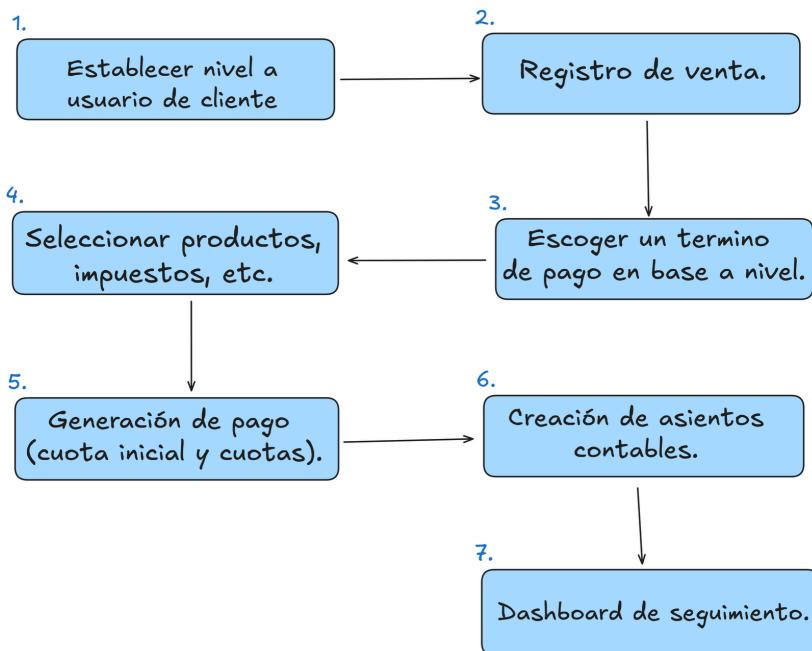
9. Presupuesto

- Infraestructura tecnológica (servidores/hosting): \$2.000.000 COP.
- Capacitación de personal: \$1.500.000 COP.
- Horas de desarrollo (estudiantes): \$3.500.000 COP.
- Total estimado: \$7.000.000 COP.

10. Levantamiento de información

Se realizará mediante entrevistas, observación de procesos actuales y reuniones con personal de ventas y contabilidad, con el fin de identificar requerimientos funcionales y no funcionales.

11. Diagrama de flujo de solución seleccionada



12. Historias de Usuario

HU1 – Reunir información inicial con stakeholders

Como analista, quiero reunir información de directivos, ventas y contabilidad, para definir los requerimientos del sistema.

HU2 – Documentar procesos actuales de facturación y pagos

Como contador, quiero documentar los procesos actuales de facturación y pagos, para tener claridad sobre qué automatizar en Odoo.

HU3 – Diseñar estructura del módulo en Odoo 16

Como desarrollador, quiero diseñar la estructura base del módulo en Odoo 16, para garantizar escalabilidad y mantenibilidad.

HU4 – Definir roles y permisos de usuarios en el sistema

Como administrador, quiero definir roles y permisos en el sistema, para garantizar seguridad y control de accesos.

HU5 – Extender modelos de Odoo para esquemas de pago

Como desarrollador, quiero extender los modelos de Odoo, para gestionar esquemas de pago flexibles (cuota inicial + cuotas quincenales).

HU6 – Configurar seguridad y accesos por roles

Como administrador, quiero configurar accesos por roles, para proteger la información sensible.

HU7 – Implementar lógica de negocio para facturación automática

Como vendedor, quiero que la facturación sea automática, para reducir tiempos y errores en los procesos de cobro.

HU8 – Automatizar generación de asientos contables

Como contador, quiero que los asientos contables se generen automáticamente, para evitar errores manuales y ahorrar tiempo.

HU9 – Diseñar mockup del dashboard financiero

Como gerente, quiero un mockup del dashboard, para validar la visualización de pagos antes de su desarrollo.

HU10 – Desarrollar dashboard gráfico en Odoo

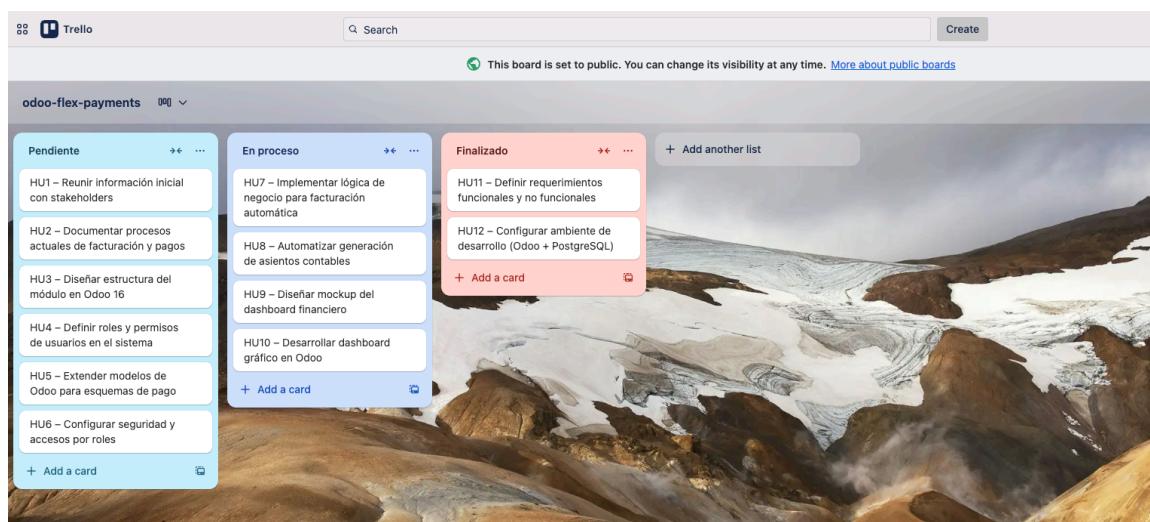
Como gerente, quiero un dashboard gráfico en Odoo, para monitorear en tiempo real el estado de pagos.

HU11 – Definir requerimientos funcionales y no funcionales

Como analista, quiero definir RQF y RQNF, para establecer criterios claros de aceptación del proyecto.

HU12 – Configurar ambiente de desarrollo (Python + Odoo + PostgreSQL)

Como desarrollador, quiero configurar el ambiente de desarrollo, para iniciar la implementación del módulo sin retrasos.



13. Definición de RQF-RQNF

Requisitos funcionales (RQF):

- Generación automática de facturas y asientos contables.
- Configuración de esquemas de pago por cliente.
- Dashboard de pagos actualizado en tiempo real.

Requisitos no funcionales (RQNF):

- Usabilidad: interfaz intuitiva y accesible.
- Rendimiento: procesamiento de facturación y pagos en menos de 2 segundos.
- Seguridad: control de acceso por roles y respaldo de información.

Repositorio GitHub:

14. Evidencias de herramientas de Design Thinking para la fases de Empatizar, definir e Idear

Fase 1. Empatizar

Objetivo: comprender a profundidad la necesidad de los usuarios internos (ventas, contabilidad, directivos) y externos (clientes) respecto a los procesos financieros.

Herramientas utilizadas:

- Entrevistas semiestructuradas: realizadas con directivos, área de ventas y contabilidad para conocer los puntos críticos de la facturación manual.
- Observación de procesos actuales: seguimiento de cómo se emiten facturas y se registran pagos en la práctica, identificando cuellos de botella y errores frecuentes.
- Mapa de Stakeholders: se clasificaron actores clave (directivos, contabilidad, ventas, clientes) y se analizaron sus expectativas frente al sistema (evidencia: apartado 7 del documento).

- Empathy Map (Mapa de empatía): se representaron frustraciones, necesidades y deseos de los usuarios (ejemplo: contadores desean reducir errores manuales; clientes desean pagos más ágiles y flexibles).

Fase 2. Definir

Objetivo: delimitar con claridad el problema y priorizar las oportunidades de solución.

Herramientas utilizadas:

- Planteamiento del problema: se formuló como la necesidad de contar con un sistema automatizado de facturación y pagos flexibles en Odoo 16, evitando retrasos y errores (evidencia: sección 2 del documento).
- POV (Point of View Statements): declaración de necesidades específicas, por ejemplo:
 - “Los contadores necesitan automatizar asientos contables porque actualmente dedican mucho tiempo a registros manuales y cometen errores que afectan el cierre financiero.”
 - “Los clientes necesitan esquemas de pago flexibles porque las condiciones actuales no se ajustan a sus posibilidades económicas.”
- Matriz de riesgos: permitió definir qué limitaciones y amenazas debía considerar el proyecto (evidencia: apartado 8 del documento).
- Historias de usuario: utilizadas para definir funcionalidades desde la perspectiva de cada rol (evidencia: apartado 12 del documento).

Fase 3. Idear

Objetivo: generar alternativas y seleccionar las mejores soluciones para el proyecto.

Herramientas utilizadas:

- Miro: sesión en equipo para proponer funcionalidades del módulo (automatización de facturación, esquemas de pagos, dashboards).
- Diagrama de flujo de solución seleccionada: se esquematizó cómo interactuarán los usuarios con el sistema (evidencia: apartado 11 del documento).
- Priorización con criterios de viabilidad y valor: se seleccionaron funcionalidades críticas (automatización, pagos flexibles y dashboard en tiempo real) que entregan mayor impacto con recursos disponibles.

Prototipo Funcional

Enlace de descarga del módulo:

- Repositorio GitHub: <https://github.com/caoc8507/TRIBU-ODO-CASH>

Instrucciones de instalación:

1. Clonar o copiar el módulo en /odoo/custom/addons
2. Actualizar lista de aplicaciones desde el panel de Odoo
3. Instalar el módulo 'odoo-flex-payments' - Reiniciar el servicio si es necesario

Capturas del funcionamiento:

1. Formulario de cliente con nivel financiero.

Individual Company

Abigail Peterson

Company Name...

Private Address

Street...
Street 2...

City State
United States

Tax ID ? e.g. BE0477472701

Nivel de Usuario

Nivel 1	
Nivel 2	
Nivel 3	

2. Configuración de esquemas de pago.

Payment Terms ? 6 Cuotas

Description on the Payment terms: 15 Days
Invoice ?

Display terms on invoice ?

Niveles Permitidos ? Nivel 1 ✕

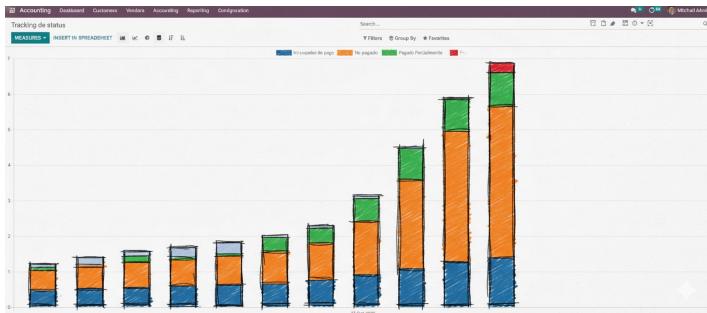
Nivel 2	
Nivel 3	

TERMS

3. Asientos contables automáticos.

Invoice Lines	Journal Items	Other Info	Debit	Credit	Tax Grids	
Account	Label					
400000 Product Sales	[FURN_8999] Three-Seat Sofa Three Seater Sofa with Lounger in Steel Grey Colour		\$ 0.00	\$ 7,500.00		
400000 Product Sales	[FURN_B220] Four Person Desk Four person modern office workstation		\$ 0.00	\$ 11,750.00		
251000 Tax Received	Tax 15%		\$ 0.00	\$ 2,887.50		
121000 Account Receivable	INV/2025/00004		\$ 22,137.50	\$ 0.00		
			\$ 22,137.50	\$ 22,137.50		

4. Dashboard de estado de pagos.



Modelos de Comportamiento y Estructura

Casos de uso

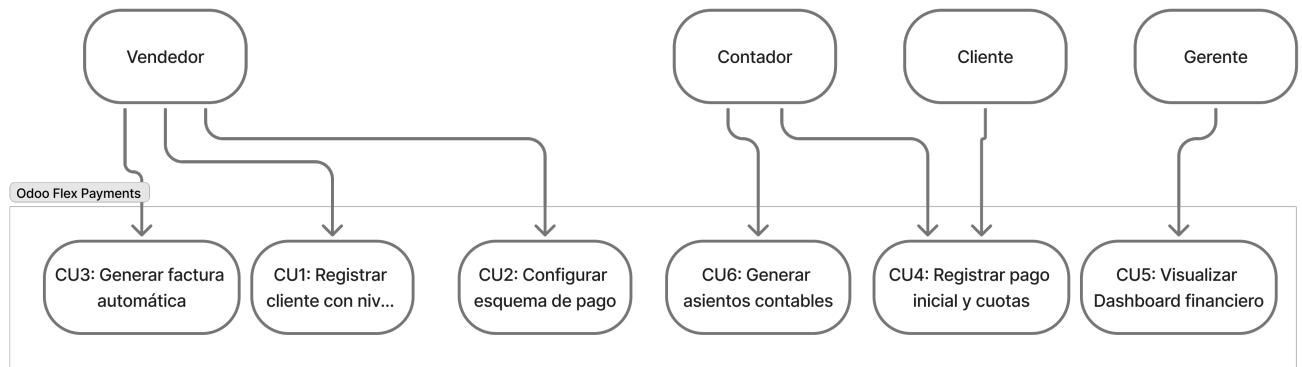


Diagrama de secuencia

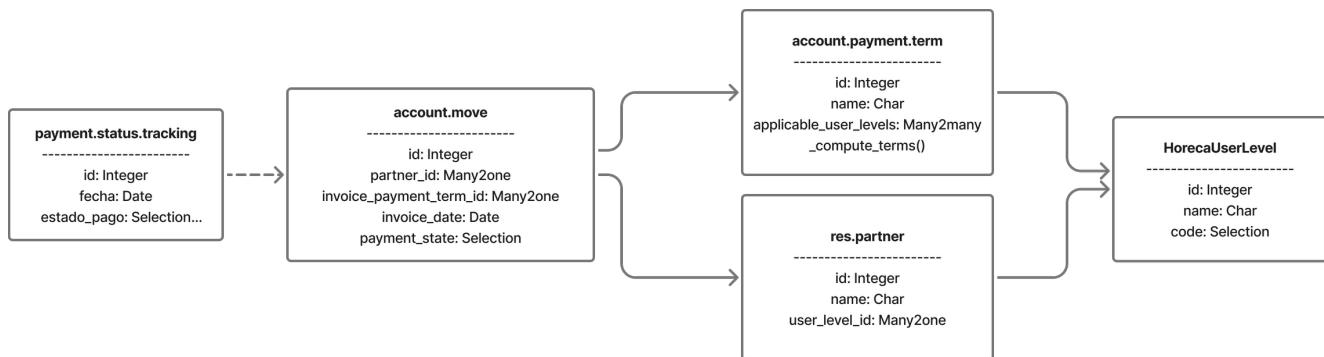
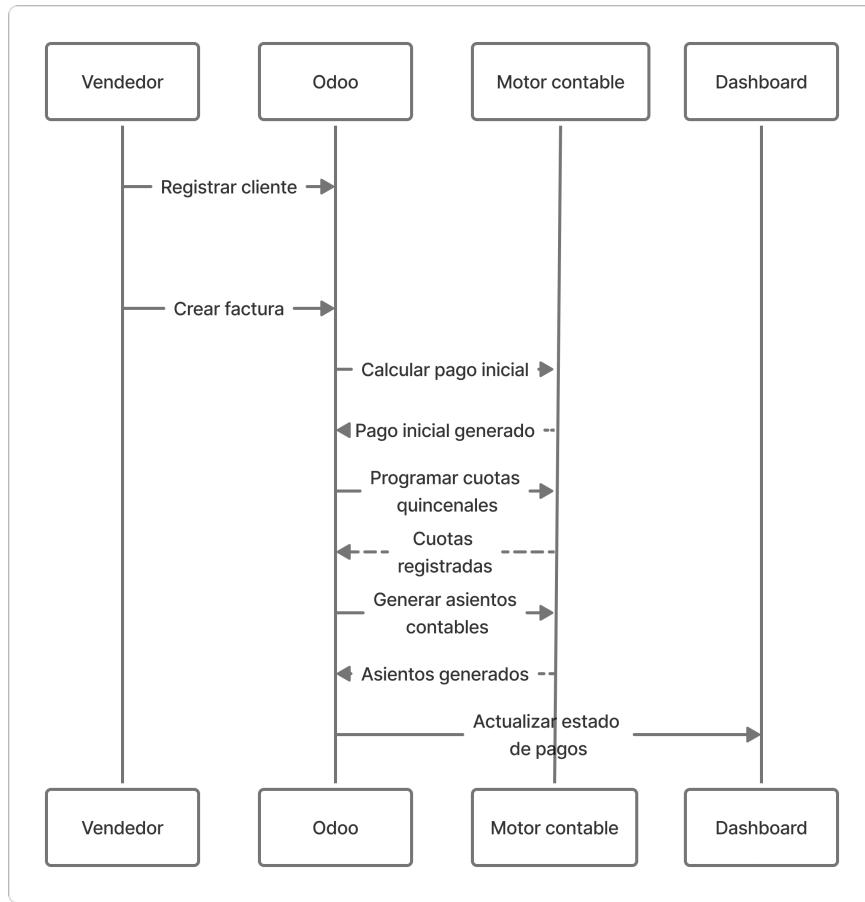
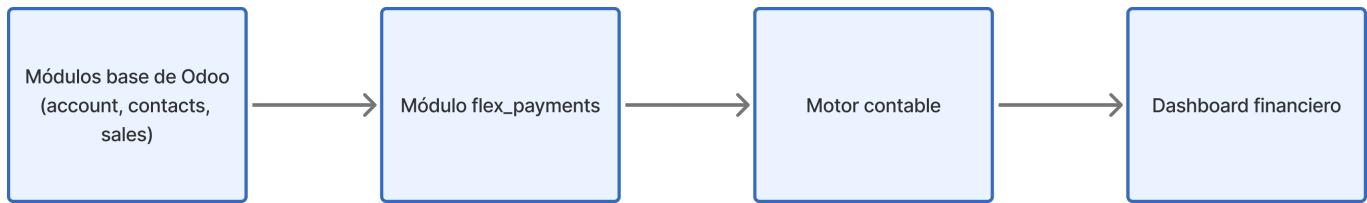


Diagrama de Clases

Diagrama de componentes



Modelo Relacional de base de datos

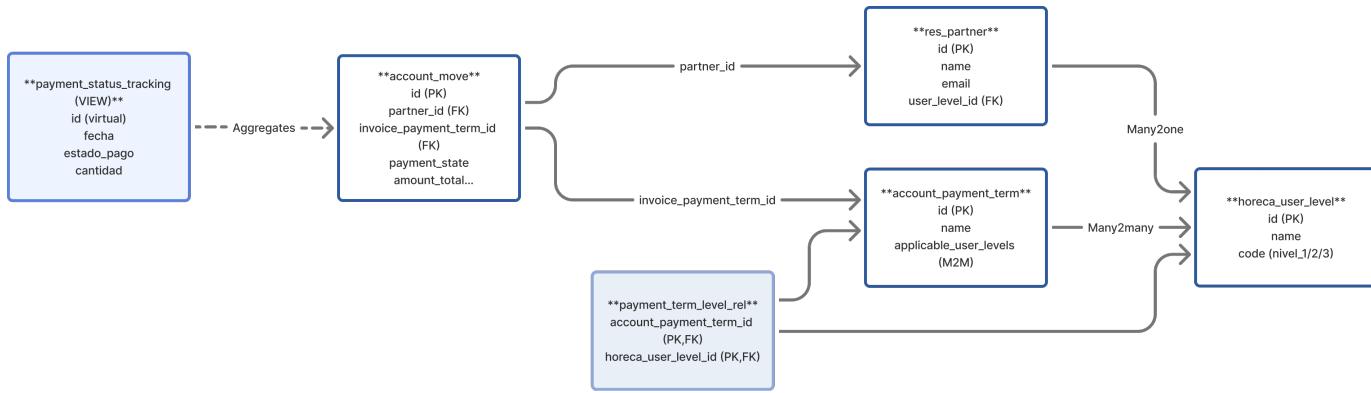
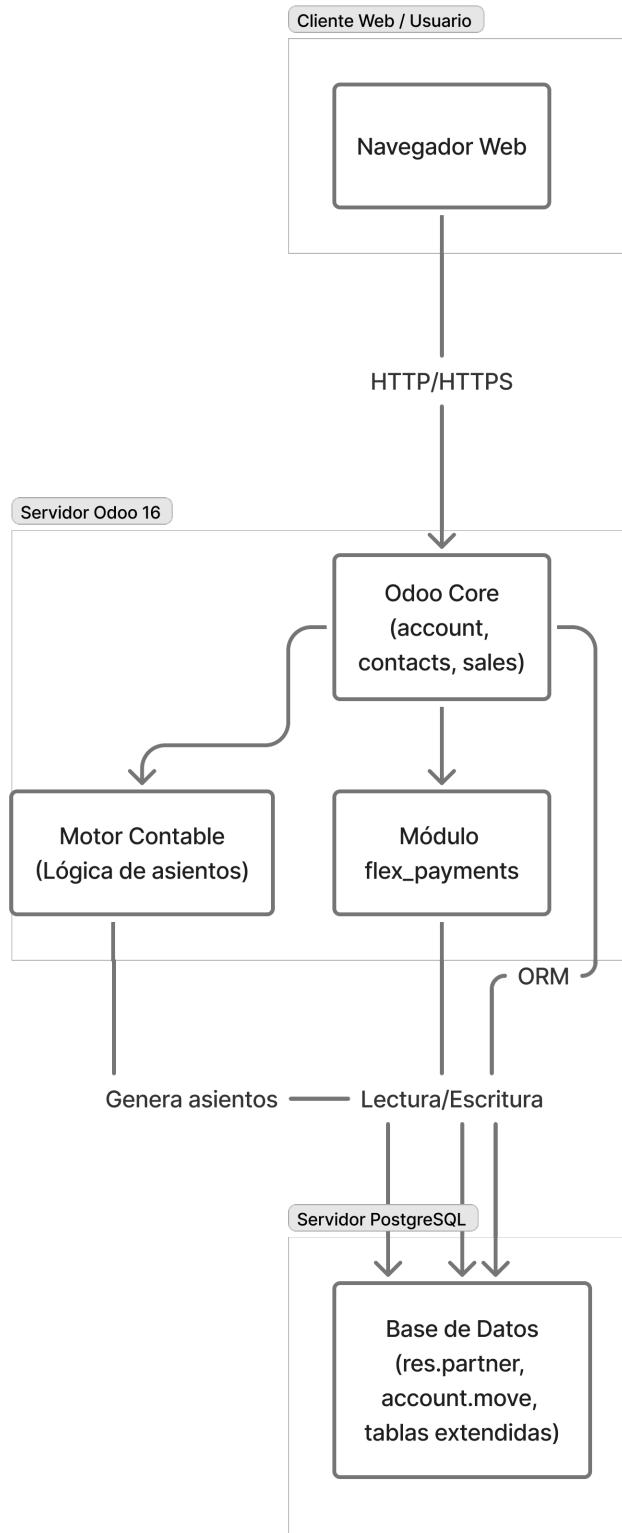


Diagrama de despliegue



Pruebas de Usabilidad

Nombre del proyecto: Odoo-Flex-Payments

Versión: 1.0

Fecha: 30/10/2025

Herramientas: Odoo + observación directa

Participantes: 5 usuarios

Tareas evaluadas

1. Registrar cliente (incluyendo nivel de usuario).
2. Configurar término de pago según nivel.
3. Generar una factura.
4. Verificar y procesar el pago inicial automático.
5. Consultar el dashboard de estado de pago.

Resultados

- Tiempo promedio por tarea: 18 segundos
- Tasa de éxito: 96%
- Errores menores:
 - Breve confusión al ubicar el campo Nivel de Usuario en el formulario (se resolvió rápidamente sin impacto negativo).

Observaciones generales

- El flujo resultó intuitivo y fácil de seguir.
- El cálculo automático del pago inicial fue comprendido sin dificultad.
- El dashboard fue considerado claro y útil para ver el estado de pago.
- No se registraron fallos técnicos ni bloqueos.
- La experiencia general fue positiva, con una curva de aprendizaje baja.

Patrones y Arquitectura de Software

Mvc (model-view-controller):

- Descripción:
 - Separa la aplicación en tres componentes: Modelo (datos y lógica), Vista
 - (interfaz de usuario) y Controlador (lógica de control).
- Justificación:
 - Separación clara de responsabilidades
 - Facilita el mantenimiento y testing
 - Es el patrón base del framework Odoo
- Implementación en el módulo:
 - Model - archivos en /models/
 - Horeca_user_level.py → define niveles de usuario
 - Res_partner.py → extiende modelo de contactos
 - Account_payment_term.py → lógica de cálculo de pagos
 - Payment_status_tracking.py → agregación de datos de tracking
 - View - archivos en /views/
 - Res_partner_view.xml → formulario de contacto con nivel
 - Account_payment_term.xml → vista de términos de pago
 - Graph.xml → gráfico de estados de pago
 - Controller - métodos en modelos
 - _compute_terms() en account_payment_term.py → calcula cuotas de pago según nivel del partner

Arquitectura en capas

- Descripción:
 - Organización del código en capas horizontales con responsabilidades específicas.
- Justificación:
 - Reduce acoplamiento entre componentes
 - Facilita testing independiente de cada capa
 - Mejora la mantenibilidad del código
- Implementación:
 - Capa de presentación views/*.xml (ui)
 - Capa de lógica de negocio models/*.py (business logic)
 - Validaciones

- Cálculos de pagos
- Reglas de negocio
- Capa de acceso a datos odoo orm
 - Crud operations
 - Relaciones
- Capa de persistencia | postgresql database

Arquitectura modular

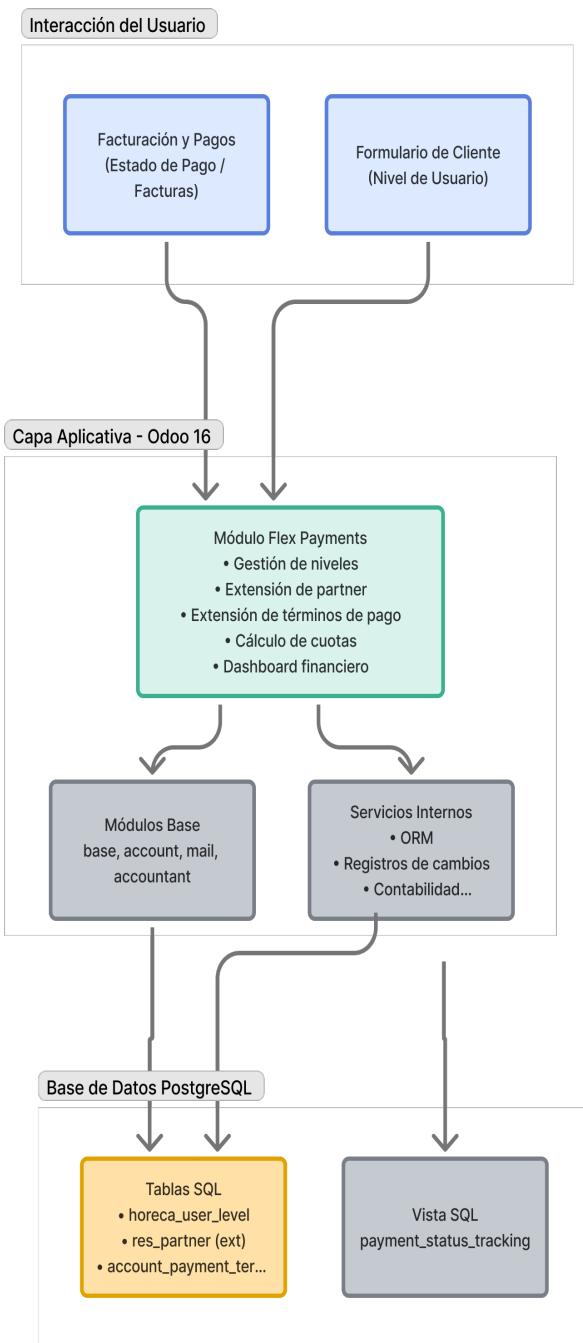
- **Descripción:**
 - El código está organizado como un módulo independiente de odoo que puede instalarse, actualizarse o removese sin afectar otros módulos.
- **Justificación:**
 - Reutilización en diferentes instalaciones de odoo
 - Fácil mantenimiento y actualización
 - Despliegue flexible según necesidades
- **Estructura del módulo**

```

odoo-flex-payments/
├── __init__.py                  # Inicialización
├── __manifest__.py               # Metadatos y dependencias
└── models/
    ├── horeca_user_level.py     # Lógica de negocio
    ├── res_partner.py           # Modelo principal
    ├── account_payment_term.py  # Extensión Partner
    └── payment_status_tracking.py # Lógica de pagos
        # Reporting
    └── views/                   # Interfaces de usuario
    └── data/
        └── horeca_user_levels.xml # Datos iniciales
            # 3 niveles predefinidos
    └── security/                # Permisos de acceso
        └── ir.model.access.csv

```

Diagrama de arquitectura del desarrollo



Conclusión

La implementación del módulo en Odoo 16 permitirá a Tribu SAS reducir errores manuales, agilizar los procesos contables y de facturación, y obtener información confiable para la toma de decisiones estratégicas. Asimismo, la integración de esquemas de pago flexibles impactará positivamente en la satisfacción de los clientes y en la eficiencia de la empresa.

Desde el punto de vista académico y personal, este proyecto ha representado un proceso de aprendizaje significativo, ya que me permite aplicar conocimientos en análisis de requerimientos, diseño e implementación de módulos en Odoo, uso de metodologías ágiles y gestión de datos financieros. Estos aprendizajes enriquecen mi perfil profesional y me preparan mejor para afrontar retos en el desarrollo de soluciones empresariales.

En conclusión, este trabajo no solo plantea una solución tecnológica para la organización, sino que también aporta a mi formación como ingeniero de software, al brindarme experiencia práctica en proyectos reales con impacto en la transformación digital de las empresas.

Bibliografía

- <https://elibro.net/es/ereader/biblioibero/197008>
- <https://visuresolutions.com/es/gu%C3%ADa-de-limosna/requisitos-funcionales-vs-no-funcionales/#:~:text=Los%20requisitos%20funcionales%20claros%20y,en%20condiciones%20del%20mundo%20real.>
- <https://businessmap.io/es/recursos-de-kanban/primeros-pasos/que-es-kanban#:~:text=Kanban%20es%20un%20m%C3%A9todo%20Lean,la%20eficiencia%20y%20mejorar%20continuamente.>