

UNIVERSIDAD PERUANA LOS ANDES

FACULTAD DE INGENIERÍA

**ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS Y COMPUTACIÓN**



ACTIVIDAD DE CLASE

ESTUDIANTE:

PALOMINO MEZA DAVID EDUARDO

CURSO:

ARQUITECTURA DE SOFTWARE

HUANCAYO – PERU

2025

Actividad 1: Análisis de Estándares en un Proyecto Real

ERP “ODOO”

1. Descripción del sistema:

Odoo es un sistema ERP disponible tanto en la nube como local, compuesto por diferentes módulos para gestionar todas las áreas de su negocio. Divididos por áreas, los módulos de Odoo están completamente integrados entre sí para facilitar el día a día de la empresa y reducir el tiempo de gestión.

Odoo está diseñado para digitalizar completamente su empresa, permitiendo la importación de todos sus archivos tanto en Excel como en CSV. También cuenta con el sistema OCR para la digitalización de facturas, ahorrando así el trabajo manual del equipo de contabilidad.

Las automatizaciones en Odoo han sido desarrolladas para optimizar los procedimientos diarios, pero también tienes la libertad de crear tus propias reglas de automatización gracias a la herramienta de personalización (Studio) y al módulo de automatización de marketing. Además, dentro de los sistemas ERP encontramos los de código abierto, como es el caso de Odoo. Con este tipo de sistema, tendrá acceso al código del sistema para aprovechar todas las ventajas de un software profesional, desarrollado para empresas, y podrá realizar las personalizaciones e integraciones necesarias para gestionar cualquier proceso.

2. Estándares identificados (por ejemplo, ISO/IEC/IEEE 42010, ISO/IEC 25010):

Odoo, como ERP modular y de código abierto, no declara explícitamente que cumple ciertos estándares internacionales de manera certificada (como ISO o IEEE), pero sí aplica prácticas y tecnologías que se alinean con estándares internacionales reconocidos en el desarrollo de software, arquitectura, calidad, seguridad e interoperabilidad.

1. Estándares de Arquitectura de Software (ISO/IEC/IEEE 42010:2011)

Arquitectura de Sistemas de Software:

- a. Arquitectura modular basada en componentes (módulos).
- b. Múltiples vistas de arquitectura: datos, procesos, seguridad, infraestructura.
- c. Independencia y comunicación entre módulos mediante interfaces claras.

2. Estándares de Calidad del Software (ISO/IEC 25010:2011)

Modelo de calidad del software:

- a. Funcionalidad: ERP completo con múltiples módulos integrados.
- b. Fiabilidad: Pruebas unitarias, integración y continua.
- c. Usabilidad: UI amigable y personalizable.
- d. Mantenibilidad: Sistema modular y de código abierto.
- e. Seguridad: Roles de acceso, autenticación, control de permisos.

3. Estándares de Seguridad de la Información (ISO/IEC 27001)

Gestión de la seguridad de la información:

- a. Control de accesos basado en roles (RBAC).
- b. Cifrado de contraseñas y uso de HTTPS.
- c. Opcional: Autenticación de dos factores (2FA).
- d. Respaldos automáticos en Odoo.sh (plataforma oficial en la nube).

4. Estándares de Interoperabilidad y APIS (REST y JSON-RPC / XML-RPC)

Estándares de APIs para comunicación entre sistemas:

- a. Soporte para APIs RESTful.
- b. Soporte para JSON-RPC y XML-RPC, utilizados en integraciones con otros sistemas.
- c. Uso de formatos estándar como JSON y XML.

5. Estándares de Desarrollo de Software (PEP8 “Python Enhancement Proposal 8”)

Estándar de estilo para código Python:

- a. Odoo está desarrollado en Python y sigue (en gran parte) las normas de estilo de PEP8 para mantener legibilidad y consistencia en el código.

6. Estándares de Base de Datos (SQL “Estándar a través de PostgreSQL”)

Aplicación en Odoo:

- a. Odoo usa PostgreSQL, que es conforme al estándar SQL:2011 y posteriores.

- b. Soporte para integridad referencial, transacciones ACID, vistas y procedimientos.

7. Estándares de Gestión Empresarial (TOGAF “The Open Group Architecture Framework”)

Marco para la arquitectura empresarial:

- a. Aunque Odoo no implementa TOGAF como tal, su estructura modular, escalabilidad y alineación con procesos empresariales reflejan buenas prácticas de TOGAF.
- b. Permite adaptar la arquitectura del ERP al crecimiento y cambio estratégico de la empresa.

3. Justificación de su aplicación:

❖ ISO/IEC/IEEE 42010:2011 – Arquitectura de Sistemas de Software

○ ¿Por qué se aplica?

- Odoo es un ERP altamente **modular**: cada área funcional (ventas, compras, contabilidad, inventario, etc.) se implementa como un módulo independiente.
- Esta arquitectura facilita la separación de responsabilidades, la reutilización de código y el mantenimiento del sistema.

○ ¿Para qué se aplica?

- Para permitir el desarrollo independiente y paralelo de módulos.
- Para facilitar la personalización sin afectar el núcleo del sistema.
- Para documentar y comunicar claramente la arquitectura entre desarrolladores y partes interesadas.

❖ ISO/IEC 25010:2011 – Modelo de Calidad del Software

○ ¿Por qué se aplica?

- Odoo busca garantizar la **calidad integral** del software que ofrece a miles de empresas en todo el mundo.
- Las empresas demandan software **fiable, seguro, usable, mantenible y funcional**.

○ ¿Para qué se aplica?

- Para asegurar que el ERP funcione correctamente bajo diferentes condiciones (calidad funcional).
- Para mantener un sistema estable que pueda evolucionar sin introducir errores (mantenibilidad).

- Para asegurar que los datos empresariales estén protegidos (seguridad).
- Para proporcionar una experiencia de usuario sencilla e intuitiva (usabilidad).

❖ **ISO/IEC 27001 – Gestión de Seguridad de la Información (*Buenas prácticas aplicadas*)**

- **¿Por qué se aplica?**
 - Odoo maneja datos sensibles: financieros, personales, nóminas, registros de clientes, etc.
 - Las amenazas de ciberseguridad están en aumento.
- **¿Para qué se aplica?**
 - Para proteger la **confidencialidad, integridad y disponibilidad** de la información.
 - Para aplicar medidas de control de acceso (RBAC), cifrado de datos y respaldos seguros.
 - Para cumplir con regulaciones como **GDPR** en Europa o similares en otros países.

❖ **REST, XML-RPC y JSON-RPC – Estándares de APIs**

- **¿Por qué se aplica?**
 - Odoo debe **interactuar con otros sistemas**, como tiendas online, bancos, plataformas contables, etc.
 - Las integraciones son críticas para la transformación digital de las empresas.
- **¿Para qué se aplica?**
 - Para permitir que otros sistemas consulten, envíen o modifiquen datos en Odoo (por ejemplo, registrar ventas desde una tienda en línea).
 - Para desarrollar aplicaciones móviles, portales de cliente o integraciones con plataformas externas (como Shopify, PayPal, Amazon, etc.).

❖ **PEP8 – Estándar de Estilo de Código en Python**

- **¿Por qué se aplica?**
 - Odoo está escrito en **Python**, y mantener un código limpio y legible es esencial para la colaboración entre múltiples desarrolladores.
- **¿Para qué se aplica?**
 - Para garantizar la **consistencia del código fuente**.
 - Para facilitar la **lectura, revisión y mantenimiento** del código por la comunidad de desarrolladores y socios de Odoo.
 - Para reducir errores y mejorar la calidad del software a largo plazo.

❖ **SQL Estándar – A través de PostgreSQL**

- **¿Por qué se aplica?**

- Odoo usa **PostgreSQL**, un sistema de gestión de bases de datos relacional que sigue el estándar **SQL**.
- **¿Para qué se aplica?**
 - Para asegurar la **integridad y consistencia de los datos** almacenados.
 - Para soportar transacciones complejas, informes y consultas personalizadas sin depender de tecnologías propietarias.
 - Para facilitar la migración, respaldo y mantenimiento de la base de datos.
- ❖ **TOGAF – Marco de Arquitectura Empresarial (*Aplicación conceptual*)**
 - **¿Por qué se aplica?**
 - Aunque Odoo no implementa TOGAF formalmente, muchas de sus prácticas de diseño reflejan los principios de este marco.
 - **¿Para qué se aplica?**
 - Para asegurar que la arquitectura del ERP esté alineada con los objetivos estratégicos de la empresa usuaria.
 - Para permitir que Odoo crezca con la organización, adaptándose a nuevos procesos y tecnologías.
 - Para facilitar la planificación y el control del ciclo de vida del ERP dentro de una arquitectura empresarial más amplia.

4. Beneficios obtenidos por el uso de dichos estándares:

- ❖ **Modularidad:** Permite personalizar y escalar el sistema fácilmente.
- ❖ **Seguridad:** Protege los datos con accesos, cifrado y respaldos.
- ❖ **Interoperabilidad:** Se integra con otros sistemas mediante APIs estándar.
- ❖ **Calidad:** Asegura un software confiable, funcional y usable.
- ❖ **Mantenibilidad:** Código limpio y organizado facilita actualizaciones.
- ❖ **Mejora continua:** Recibe actualizaciones frecuentes con nuevas funciones.
- ❖ **Portabilidad:** Usa tecnologías estándar que evitan dependencia del proveedor.
- ❖ **Auditoría:** Permite rastrear y controlar acciones dentro del sistema.

5. ANEXOS:

❖ ESTANDARES USADOS EN ODOO

Categoría	Estándar / Tecnología	Aplicación en Odoo
Arquitectura	ISO/IEC/IEEE 42010	Arquitectura modular basada en vistas y componentes
Calidad de Software	ISO/IEC 25010	Funcionalidad, mantenibilidad, seguridad, usabilidad
Seguridad de la Información	ISO/IEC 27001 (buenas prácticas)	Control de acceso, cifrado, backups
Interoperabilidad	REST, JSON-RPC, XML-RPC	Integración con otros sistemas
Desarrollo	PEP8	Buenas prácticas de desarrollo en Python
Base de Datos	SQL (PostgreSQL)	Consulta y manipulación de datos conforme a estándares
Arquitectura Empresarial	TOGAF (no oficial)	Alineación con procesos y escalabilidad

❖ TABLA DE BENEFICIOS

Categoría	Beneficio	Resultado
Modularidad	Adaptación por módulos	Mayor personalización y escalabilidad
Seguridad	Cifrado, control de accesos, backups	Protección de datos empresariales y cumplimiento normativo
Interoperabilidad	APIs estándar, base de datos SQL	Integración con sistemas externos
Mantenibilidad	Código limpio y modular	Menores costos de soporte y evolución más ágil
Mejora continua	Actualizaciones frecuentes	Innovación constante sin grandes migraciones
Calidad y fiabilidad	Software robusto y funcional	Menor tasa de errores y fallos
Colaboración	Buenas prácticas de codificación	Desarrollo más eficiente en equipos distribuidos
Portabilidad	Uso de tecnologías estándar	Flexibilidad y libertad tecnológica
Auditoría y trazabilidad	Registro de actividades	Control y cumplimiento normativo

Actividad 2: Diseño de una Arquitectura Basada en Estándares

Sistema: Biblioteca Virtual Miguel de Cervantes (BVMC)

Normas aplicadas: ISO/IEC/IEEE 42010, ISO/IEC 25010

1. Objetivos del Sistema:

- a. Proporcionar acceso digital a obras literarias, científicas y culturales en español.
- b. Facilitar la búsqueda, recuperación y lectura de contenido desde cualquier dispositivo.
- c. Integrar estándares internacionales para interoperabilidad y preservación digital.
- d. Ofrecer servicios semánticos para enlazar contenidos con otros repositorios globales.

2. Stakeholders y sus Preocupaciones:

Stakeholder	Rol	Preocupaciones
Usuarios finales	Lectores, investigadores	Acceso rápido, contenido confiable, experiencia de usuario
Bibliotecas	Curadores de contenidos	Organización, catalogación, preservación
Desarrolladores	Equipo Técnico	Modularidad, escalabilidad, mantenimiento
Instituciones aliadas	Repositorios externos	Interoperabilidad, estándares comunes
Administradores de TI	Infraestructura y seguridad	Alta disponibilidad, seguridad, cumplimiento normativo

3. Vistas Arquitectónicas (según ISO/IEC/IEEE 42010):

- a. Vista Lógica:
 - i. Modelo multicapa:
 - 1. Capa de presentación (interfaz web).
 - 2. Capa de servicios web (REST/JSON).
 - 3. Capa de negocio (EJBs con lógica de búsqueda y catalogación).
 - 4. Capa de acceso a datos (repositorio RDF, bases relacionales).

ii. **Servicios:**

1. Búsqueda semántica.
2. Recuperación de obras.
3. Enlace con repositorios externos (DBpedia, Europeana, Wikidata).

b. Vista de Desarrollo:

i. **Tecnologías:**

1. J2EE sobre servidor JBoss.
2. RDF para representación semántica.
3. JSON para intercambio de datos.
4. RDA (Resource Description and Access) como estándar de catalogación.

ii. **Estructura modular:**

1. Componentes distribuidos (EJB).
2. Servicios desacoplados para interoperabilidad.

c. Vista de Procesos:

i. **Flujos clave:**

1. Consulta → Búsqueda semántica → Recuperación → Visualización.
2. Catalogación → Enlace semántico → Publicación.

ii. **Procesamiento distribuido:**

1. Clúster de servidores para alta disponibilidad.
2. Balanceo de carga para múltiples usuarios concurrentes.

d. Vista Física:

i. **Infraestructura:**

1. Servidores JBoss en clúster.
2. Repositorio RDF distribuido.
3. Integración con repositorios externos vía web semántica.

4. Relación con los Estándares Aplicados:

a. ISO/IEC/IEEE 42010

i. Aplicado en la definición de:

1. Stakeholders y sus preocupaciones.
2. Vistas arquitectónicas.
3. Decisiones arquitectónicas documentadas.
4. Uso de puntos de vista para abordar preocupaciones específicas.

b. ISO/IEC 25010

i. Atributos de calidad considerados:

1. **Compatibilidad:** interoperabilidad semántica
2. **Fiabilidad:** arquitectura en clúster
3. **Usabilidad:** interfaz web intuitiva
4. **Mantenibilidad:** modularidad con EJBs
5. **Seguridad:** control de acceso, cifrado de datos

5. Decisiones Arquitectónicas Clave:

Decisión	Justificación
Uso de SOA (Arquitectura orientada a servicios)	Facilita la interoperabilidad y reutilización de componentes
RDF + RDA para catalogación	Permite enlazar obras con repositorios semánticos externos
JBoss en clúster	Alta disponibilidad y escalabilidad
JSON como formato de intercambio	Ligero, compatible con múltiples plataformas

Actividad 3: Evaluación de Calidad Arquitectónica

Informe de Evaluación de Calidad Arquitectónica:

Sistema Evaluado: Biblioteca Virtual Miguel de Cervantes
Norma Aplicada: ISO/IEC 25010
Atributos Evaluados: Usabilidad, Rendimiento, Seguridad, Mantenibilidad

1. Matriz de Evaluación de Calidad

Atributo	Subcaracterísticas ISO/IEC 25010	Estado Actual (1–5)	Estado Ideal (1–5)	Brecha	Observaciones
Usabilidad	Interfaz intuitiva, accesibilidad, ayuda contextual	3	5	2	Interfaz funcional pero poco adaptada a dispositivos móviles
Rendimiento	Tiempo de respuesta, uso de recursos, capacidad	4	5	1	Buen rendimiento general, pero no optimizado para alta concurrencia
Seguridad	Confidencialidad, integridad, autenticación	3	5	2	Autenticación básica, sin MFA ni cifrado avanzado
Mantenibilidad	Modularidad, reusabilidad, facilidad de análisis	3	5	2	Arquitectura distribuida, pero con acoplamiento fuerte entre módulos

2. Análisis Crítico:

- a. Usabilidad:
 - i. **Problema:** La interfaz no está completamente adaptada a estándares de accesibilidad (WCAG), ni optimizada para dispositivos móviles.
 - ii. **Mejora propuesta:** Rediseñar la UI con enfoque responsive y accesible; incorporar guías de usuario y navegación contextual.
- b. Rendimiento:
 - i. **Problema:** Aunque el sistema responde bien, no está preparado para picos de tráfico ni para usuarios concurrentes masivos.

- ii. **Mejora propuesta:** Implementar caché distribuido (ej. Redis), optimizar consultas semánticas y escalar horizontalmente.

c. Seguridad:

- i. **Problema:** Falta autenticación multifactor, cifrado de datos sensibles y monitoreo de accesos.
- ii. **Mejora propuesta:** Integrar MFA, cifrado TLS/SSL completo, y sistema de auditoría de accesos.

d. Mantenibilidad:

- i. **Problema:** Aunque se usa J2EE y EJBs, hay acoplamiento entre servicios que dificulta el mantenimiento.
- ii. **Mejora propuesta:** Migrar hacia microservicios con APIs RESTful bien definidas, aplicar principios SOLID y documentación automatizada.

3. Reflexión Final:

El estándar **ISO/IEC 25010** proporciona una guía clara y estructurada para evaluar la calidad de software, permitiendo identificar brechas concretas entre el estado actual y el ideal. En este caso, la BVMC muestra fortalezas en rendimiento, pero necesita mejoras en usabilidad, seguridad y mantenibilidad para alinearse completamente con los estándares internacionales.

Aplicar estos estándares no solo mejora la calidad técnica, sino también la experiencia del usuario, la sostenibilidad del sistema y su capacidad de integración con otros entornos digitales.