

SISTEMA DE GESTIÓN EMPRESARIAL

TÉCNICO EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA

La gestión empresarial

Los programas **ERP-CRM** (ERP sería la herramienta que nos permite gestionar el “back office” de la empresa y el CRM sería la herramienta que nos permite gestionar cada interacción con el cliente “Front office”) permiten **analizar la toma de decisiones** en tiempo real con la finalidad de **cumplir los objetivos** marcados por las empresas. Todo esto es gracias al **control** que se hace de forma íntegra desde un **único software**.

Historia y evolución

Tienen orígenes militares. En los **años 50**, después de la Segunda Guerra Mundial, cuando **Estados Unidos** usaba **programas informáticos** con una función de **gestión logística**. Estos programas de gestión y organización son considerados como los **precursores de los ERPs** (Enterprise Resource Planning).

Con el tiempo, el mundo de la informática se va abriendo a las empresas, que perfeccionan estos programas hasta crear los actuales ERPs.

Los 60's softwares de gestión

Aparecen las primeras **computadoras comerciales en el mercado** y con ella los primeros SW (software) de **gestión y control de inventarios**. Estos primeros programas se basaban en mantener las existencias al mínimo, pero asegurándose de que siempre hubiese disponibilidad.

MRP Los 70's

Aparición de los **MRP (Material Requirement Planning** o Sistemas de Planificación de Requerimientos de Material) de la empresa **IBM**. Estos programas son considerados los **antecedentes más directos** de los actuales ERPs. Se usaban exclusivamente para la **gestión de materiales** o, lo que es lo mismo, para **automatizar** la gestión de procesos de **fabricación** de productos y artículos, permitiendo tener unas **previsiones reales** de stock, inventario, materias primas, etc.

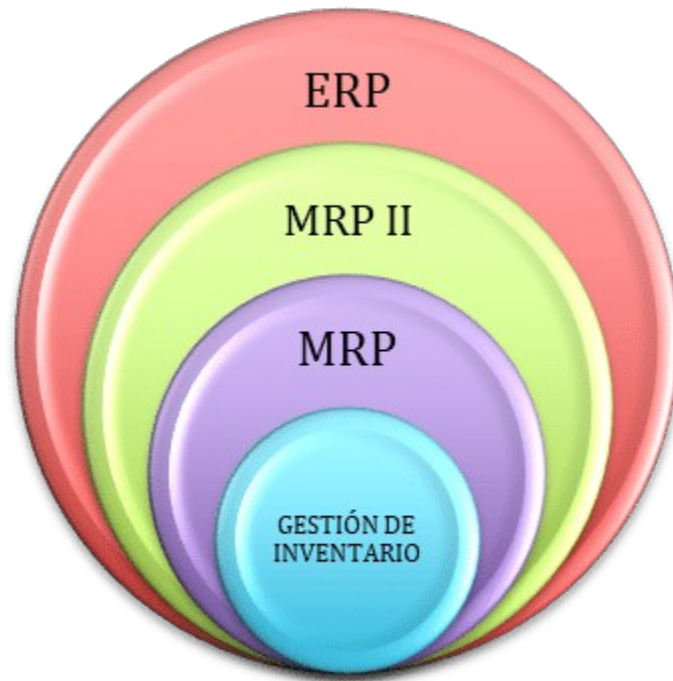
El 1 de abril de **1972**, **SAP** fue fundada en **Alemania** por cinco **empleados de IBM**. Se trata de una empresa importante, puesto que su ERP SAP (del mismo nombre que la empresa) es de los más famosos a nivel mundial.

MRP II Los 80's

MRP II (Manufacturing Resource Planning o Sistemas de Planificación de Recursos de Manufacturación), abarcan **todo el proceso logístico** de producción. El software ya no solo se centra en el **control de materiales**, sino que va un paso más allá y posibilita la **administración de recursos económicos**. Esto permite a las empresas **invertir de forma más inteligente**.

ERP Los 90's

Gartner bautizó al MRP con el término **ERP**. Nace por la constante evolución de los modelos de negocios empresariales, siendo una adaptación de los sistemas MRP. El éxito de este tipo de programas reside en la **integración** bajo un mismo software de **diferentes módulos** que trabajan independientemente entre sí, pero, a la vez, **interconectados**.



ERP en la actualidad

A partir del año **2000**, muchos desarrolladores de ERPs comenzaron a agregar **nuevas funcionalidades**, entre las que destacamos la gestión de las **relaciones con los clientes (CRM)**, de la que hablaremos en el punto 6 por ser un **binomio** muy común (**ERP-CRM**).

Es fácil pensar que estos programas tan potentes eran solo accesibles para grandes empresas y multinacionales, pero poco a poco se ha ido haciendo más **accesible a las pymes**.

ERP II o ERP extendido

Evolución del ERP. La externalización de muchos servicios por parte de las empresas hizo necesario **expandir el modelo de empresa** en forma **horizontal** frente al tradicional vertical.

La idea del ERP II es que los procesos pasen de estar disponible solo para usuarios internos a la empresa a ser procesos **transparentes para el resto de las empresas colaboradoras**, con la intención de posibilitar un **funcionamiento coordinado y ágil**. Para ello, se pasa de una arquitectura cliente-servidor al uso de **arquitectura SOA**.

El ERP es la herramienta más completa para la gestión empresarial hoy en día.

Organización estándar de una empresa

Características más comunes de una gran empresa:

La empresa está dirigida por un **director general**, encargado de **coordinar la actividad** en la empresa, y dividida en **departamentos**, los cuales tienen una **función bien delimitada** a desempeñar por los trabajadores de dicho departamento y **gestionada por los responsables** de **departamento**, siendo los siguientes los más comunes:

- Dirección
- Administración
- Recursos humanos
- Compras
- Logísticas
- Finanzas
- Ventas
- Marketing
- Almacén

ERP

ERP son las siglas de **Enterprise Resource Planning** (Sistemas de Planificación de Recursos Empresariales).

Definición según diferentes autores sobre lo que es un ERP:

- **Laudon y Laudon** (2001) piensan que los sistemas ERP son **sistemas de información** que integran los procesos claves del negocio, de forma tal que la información pueda **fluir libremente** entre las diferentes partes de la organización, mejorando con ello la **coordinación, la eficiencia y** el proceso de toma de **decisiones**.
- **Lee** (2003) define un ERP como un **paquete de software** integrado de uso **empresarial**. En el ERP, todas las **funciones necesarias** del negocio, tales como finanzas, RRHH y distribución, se integran en un único sistema con una **base de datos compartida**.
- Según **Mejía** (2004), los ERP son **sistemas** que integran todos los **aspectos funcionales** de la empresa: gestión comercial, gestión financiera, gestión de producción, etc. De esta forma, el ahorro de tiempo y la minimización de errores es máximo.
- **McGaughey y Gunasekaran** (2009) definen el ERP como un sistema de información que integra procesos de negocio con el objetivo de **crear valor y reducir los costos**, haciendo que la información correcta esté disponible para las **personas adecuadas y en el momento adecuado** para ayudarles a la toma de decisiones.

Características generales de un ERP

10 características de un ERP:

- **software:** es un programa.
- **Integrado:** todos comparten bbdd y parametros.
- **adaptable, configurable:** se adapta a empresas
- **bbdd centralizada.**
- **procesos de negocios.**
- **modular.** (departamentos).
- **estándar:** distintas areas de distintas empresas.
- **Fiable** (evita errores).
- **Seguro:** control de versiones, control de intrusiones.
- **fácil** de usar.

Un ERP es

- un **software integrado** con **procesos estándar**, pero, a la vez, **configurable** a las necesidades de las empresas
- que engloba todos los **procesos de negocios** de forma modular en una **base de datos centralizada** con el objetivo de **crear valor**
- ayudando a la empresa en la toma de **decisiones**,
- ahorrando **tiempo**
- y minimizando **errores** al máximo.

CRM

Concepto

Siglas: **Customer Relationship Management**, (Sistemas de Administración de la Relación con Clientes); es un software que ayuda en la gestión de la **relación con los clientes**.

De acuerdo con **Laudon & Laudon** (2004), CRM es una **disciplina empresarial y tecnológica** para la gestión de las relaciones con el cliente con el objetivo de **incrementar la facturación, rentabilidad, satisfacción y retención**.

Están **orientados a**:

- la **gestión comercial**,
- el **marketing**
- y la **atención al cliente**.

Hacen uso del análisis de **datos históricos de los clientes** para mejorar las relaciones comerciales con ellos o con potenciales clientes con la finalidad de retenerlos o captarlos y aumentar las ventas. Permite entender las necesidades del cliente e incluso anticiparse a ellas. Es decir, si el **ERP** centraba la **estrategia de negocio en la empresa**, el **CRM** lo hace en las **necesidades y deseos del cliente**.

Normalmente, están **asociados a un software ERP** (ERP-CRM) como complemento a este, aunque **es posible usar un CRM de forma independiente**.

Mientras el **ERP** constituía el **Back Office**, el **CRM** va a suponer el **Front Office**. El **Back Office** es la parte responsable de la **gestión de la empresa** (contabilidad, RRHH, logística, etc.), la que no está de cara al cliente. Mientras que el **Front Office** es la parte que está de **cara al cliente** y genera **ingresos directos** en la empresa.

Características

- *gestión de **contactos**: bbdd de clientes.*
- *gestión de campañas de **marketing** (personalizadas).*
- *gestión de **nuevas oportunidades** de negocios.*
- *seguimiento de **interacciones** (evitar conversaciones repetidas con clientes).*
- *centralizar la **documentación**.*
- ***automatización** de **flujos de trabajo** (ahorrar tiempo).*
- *elaborar **informes y predicciones**.*

Arquitectura ERP-CRM

Los ERPs aparecieron como arquitectura **cliente-servidor**, en la que un computador central (servidor) tiene la capacidad de atender a varios usuarios simultáneamente (clientes). Pero la evolución de internet favoreció el cambio de los modelos de negocio hacia el **e-business (comercio electrónico)** y, por tanto, nuevas y múltiples mejoras de los ERP en las relaciones entre clientes, proveedores, etc. Así, aparecen...

flujos de comunicaciones:

- **B2B:** business-to-business.
- **B2C:** business-to-consumer.
- **B2E:** business-to-employee.

Este hecho alimenta la aparición de **nuevas aplicaciones empresariales** que vienen a complementar al ERP para dar respuesta a las **nuevas relaciones multi-empresas**, es decir, aparecen procesos colaborativos con **aplicaciones de terceros**.

Y es aquí donde entra en juego la **arquitectura orientada a servicios (SOA, Service Oriented Architecture)**, que aparece como una **plataforma abierta y flexible** donde las **diferentes aplicaciones empresariales (incluido el ERP)** pueden estar *totalmente integradas* mediante tecnología de **Servicios Web**. Un Servicio Web es una representación en un lenguaje estándar (XML) de un programa, objeto, bases de datos o cualquier otro tipo de función de negocios.

Características de la arquitectura SOA:

- **facilitar los procesos entre empresas,**
- **compartir información relevante** entre los diferentes agentes,
- **facilitar el trabajo de colaboración** entre los mismos.

ERP-CRM actuales libres y propietarios

El copyright es la parte más restrictiva de las licencias, (todos los derechos reservados...). El lado opuesto es el "public domain". En la manera en que vayamos modificando este dominio público iremos acercándonos más al copyright. Crear software libre o de código abierto da reconocimiento ante el resto del mundo, abriendo nuevas oportunidades, aunque no se obtenga beneficios directamente, pero sí indirectamente, ya que será valorado.

¿Software libre o código abierto?

*Los términos software libre (free software) y código abierto (open source) tienen pequeñas diferencias. La principal diferencia está en sus valores, en la forma de mirar el mundo. El movimiento de **software libre** tiene una **ideología social**, es decir, considera que aquel software que no sea libre es un **problema social**. En cambio, para el movimiento de **código abierto** es más una **cuestión práctica**, no ética, o lo que es lo mismo, piensan que el **software que no sea libre es una solución ineficiente**. Ambas ideologías trabajan en la misma línea, teniendo como "**enemigo**" **común al software privado o propietario**.*

*Stallman, en su artículo "Por qué el código abierto pierde el punto de vista del software libre", llega a indicar que el **software libre** implica **sentido de libertad**, mientras que el **código abierto** no tiene por qué implicarlo, sino **solo disponibilidad de acceso** al código fuente.*

Código abierto

Se trata de un modelo de desarrollo de software basado en la **colaboración abierta**. Concretamente, cuando se habla de libre o abierto se está refiriendo al **código fuente**, cuyo contenido es consultable, modificable y distribuible por cualquier persona. Pero el hecho de que un programa sea de **código libre no implica que sea gratuito**, aunque lo normal es que sea más económico que los de código privado. El código fuente sí es gratuito, pero este puede ser la base de programas que impliquen algún coste. La “**reprogramación**” **comercial de software libre es algo habitual**.

Según la web <https://opensource.org/>, para que un programa sea considerado de código abierto se deben cumplir las diez condiciones siguientes:

- 1- Redistribucion gratuita.
- 2- Código fuente incluido, no modificar deliberadamente el código fuente.
- 3- Obras derivadas.
- 4- Integridad del código fuente del autor.
- 5- No discriminación de personas o grupos de personas.
- 6- No discriminación contra campos de trabajo.
- 7- Distribución de licencias.
- 8- La licencia no debe ser específica para un producto.
- 9- La licencia no debe restringir otro software.
- 10- La licencia debe ser neutral en tecnología.

Código propietario

El código propietario, por su parte, exige un **pago por licencia** de uso de su software. Este suele ser **más profesional** en cuanto a interfaz gráfica, rendimiento, servicios que ofrece, opciones de personalización, etc., ya que **hay una empresa por detrás** que vive de ello y se preocupa más por los pequeños detalles. Entre otras ventajas, tiene el apoyo del **servicio técnico** de dicha empresa.

Al contrario que los de código abierto, en este caso **se priva a los usuarios de consultar, modificar y distribuir** su **código** fuente, el cual solo está a disposición de su desarrollador.

Fundación para el Software Libre (FSF): “Código es propietario o privado si no es libre, o si no lo es parcialmente (semilibre)”.

Ventajas del ERP-CRM de código libre

- Al no requerir de pago de licencia, el **coste de propiedad e implantación** del ERP-CRM debería ser más bajo, además de no suponer un delito su uso o manipulación.
- La **evolución del software** no depende de un proveedor, sino de la **comunidad**. Esto suele traducirse en **estabilidad**.
- Si dicha **comunidad es activa**, puede favorecer una **rápida evolución** del software con actualizaciones frecuentes.
- Las **tecnologías suelen ser más actuales** que en los propietarios. Además, suelen disponer de una amplia gama de **herramientas** y posibilidades de **personalización**.

Desventajasdel ERP-CRM de código libre

- La **garantía suele estar limitada**. Si la hubiese, sería **por parte del distribuidor o implantador** del software, pero **no sobre el core o núcleo**, que es la **base del ERP-CRM**.
- La **evolución** del software se hace de **forma genérica** y **depende** de la **comunidad**.
- Algunos ERP de código libre tienen una **versión gratuita limitada** y es necesario pagar para usar algunas opciones.

Ventajas del ERP-CRM de código propietario

- Son desarrollados por **grandes fabricantes** de software que ofrecen **garantías y cobertura** frente a errores, con un soporte y **servicio postventa especializado**.
- Muy **fiable**, normalmente debido a sus años de experiencia en el mercado con multitud de clientes usándolo a diario. En **constante evolución**.
- Muy **especializado**. Puede evolucionar en áreas específicas.

Desventajas del ERP-CRM de código propietario

- El **coste** suele ser muy elevado. Aunque esto ha cambiado debido a la especialización y a la elevada competencia.
- **Dependencia**: la implantación y personalización del **ERP con un proveedor** dificulta un posible futuro **cambio de ERP**.

ERPs libres

ERPNext

Persiguen diferenciarse de otros ERPs defendiendo que **solo usan software libre**, a diferencia de otros sistemas que cuentan con extensiones privadas y servicios con un coste económico.

Cuenta con una versión de código abierto bajo licencia **LGPLv3** en la que se incluyen los **módulos principales** (**stock, CRM, RRHH, ventas, finanzas**, etc.), **industriales** (centrándose en industrias como salud, agricultura, etc.), **customización e integración, webs y portales**.

El módulo del **servidor** está escrito en lenguaje **Phyton** y usa **MariaDB** como sistema gestor de base de datos. Su código fuente y documentación se encuentran alojados en la web **GitHub**, siendo su **versión actual la 12**.

Odoo

Originalmente denominado **TinyERP** y posteriormente **OpenERP**, se trata de un ERP integrado que incluye, según su web, **42 módulos** organizados en aplicaciones de **ventas** (entre las que se encuentra el CRM), **financieras, operaciones, fabricación, RRHH, comunicación, marketing, sitios webs** y **customización**.

Cuenta con una versión “comunitaria” de código abierto bajo licencia **LGPLv3** y una **versión empresarial** bajo **licencia comercial**. Para ambas, además, existe una **aplicación web** y una de **escritorio**.

Su código fuente y documentación también se encuentran alojados en la web **GitHub**, y su **versión actual** es la **13**. Es el que **más actividad** tenía **en GitHub**.

El módulo del **servidor** está escrito en lenguaje **Phyton** (requiere la versión 3.6 o posterior de Python) y usa **PostgreSQL** como sistema gestor de base de datos.

ERPs propietarios

Los ERPs propietarios más extendidos son: SAP ERP, Oracle ERP Cloud y Microsoft Dynamics ERP. Todos ellos tienen la **opción de agregar también CRM**.

El **coste** de cada uno de ellos **depende de muchos factores** como pueden ser los costes por consultoría, licencias por usuario, coste del mantenimiento, etc.

SAP ERP: Se le atribuye ser la **empresa pionera** en este tipo de tecnología. (24% cuota mercado).

La empresa ha ido evolucionando, siendo su producto el más actual, y por el que quieren apostar en un futuro cercano **SAP S/4HANA**. Cuenta con la posibilidad de **usarlo en la nube**.

Durante muchos años, ha estado a la cabeza, pero hoy en día tanto **Oracle** como **Microsoft** son claros **competidores**. El segundo con mejor tiempo de implementación: 13 meses.

Oracle ERP Cloud: De las tres que vamos a exponer, es la empresa **más potente en cuanto a la nube** se refiere. (18% cuota mercado). Tienen la **mayor tasa de satisfacción** de cliente: **80%**.

Según el estudio realizado por **Parven M. y Maimani K.**, es el **mejor en la actualidad**. Es el que tiene **menor tiempo de implementación (11 meses)**.

Microsoft Dynamics ERP: Tiene mejor **flexibilidad comercial** y es el **más económico** de los tres sistemas comentados, según el estudio realizado por Parven, M. y Maimani, K. (11% cuota mercado). El tercero en tiempo de implementación: 14 meses.

CRMs libres

Tanto los ERPs Odoo como ERPNext disponen de la opción de agregar “su” CRM, pero en este apartado vamos a analizar el **software CRM “independiente” de código abierto**.

SuiteCRM

Este CRM fue desarrollado por la **misma comunidad que SugarCRM**, otro CRM que tiene una versión open source limitada que se complementa con una versión de pago. SuiteCRM es casi tan poderoso como la versión de pago de SugarCRM, siendo ambos compatibles. Puede funcionar en gestores de bases de datos como **MySQL**, **MariaDB** o **SQL Server**.

Es muy **personalizable** e incluye, entre otras, herramientas de **gestión de contactos** o **seguimiento de clientes potenciales**.

Su código fuente y documentación están disponibles en **GitHub** con una **versión (7.11.13)** de código abierto bajo licencia **AGPL-3.0**.

Fat Free CRM

Esta solución Open Source se caracteriza por estar basada en **Ruby**, una tecnología diferente a la usada en otros CRM. Cuenta con herramientas que le permiten hacer un **seguimiento a oportunidades de ventas**, **captación** de usuarios, etc.

Al igual que SuiteCRM, es muy **personalizable**. De hecho, dispone de plugins para **agregar campos**, **combinar contactos duplicados**, etc.

Su código fuente, documentación y diferentes plugins están también disponibles en **GitHub**, en donde es el **CRM más valorado**, lo que indica que tiene una **comunidad detrás que lo apoya** y lo mantiene con actualizaciones.

CRMs propietarios

Salesforce

Muchos lo consideran el **software líder en CRM** basado en la nube. Se trata de un software muy completo, tanto que puede resultar complejo para las pequeñas empresas.

Puede utilizar funciones como **datos de ventas**, gestión de **clientes potenciales**, **automatización de marketing**, etc.

Además, es fácil de integrar con otros softwares, pudiéndose utilizar **desde cualquier dispositivo**, puesto que cuenta con aplicaciones móviles.

Se adapta a los **diferentes tipos de empresa**, permitiéndoles elegir planes que pueden ir evolucionando junto a ellas.

Zoho CRM

Al igual que Salesforce, es un CRM muy completo con **más de 15 años de experiencia** y que también trabaja en la **nube**. Permite **agregar datos de clientes** de una variedad de fuentes, incluyendo **correos electrónicos**, **chats** en vivo, **llamadas** o **redes sociales**.

Está diseñado para **facilitar la gestión del ciclo de vida** de la **relación con el cliente** administrando las ventas, el marketing, el servicio de atención al cliente, etc.

Ofrece funciones para **crear y automatizar procesos de ventas**, permitiendo crear portales personalizables para sus clientes.

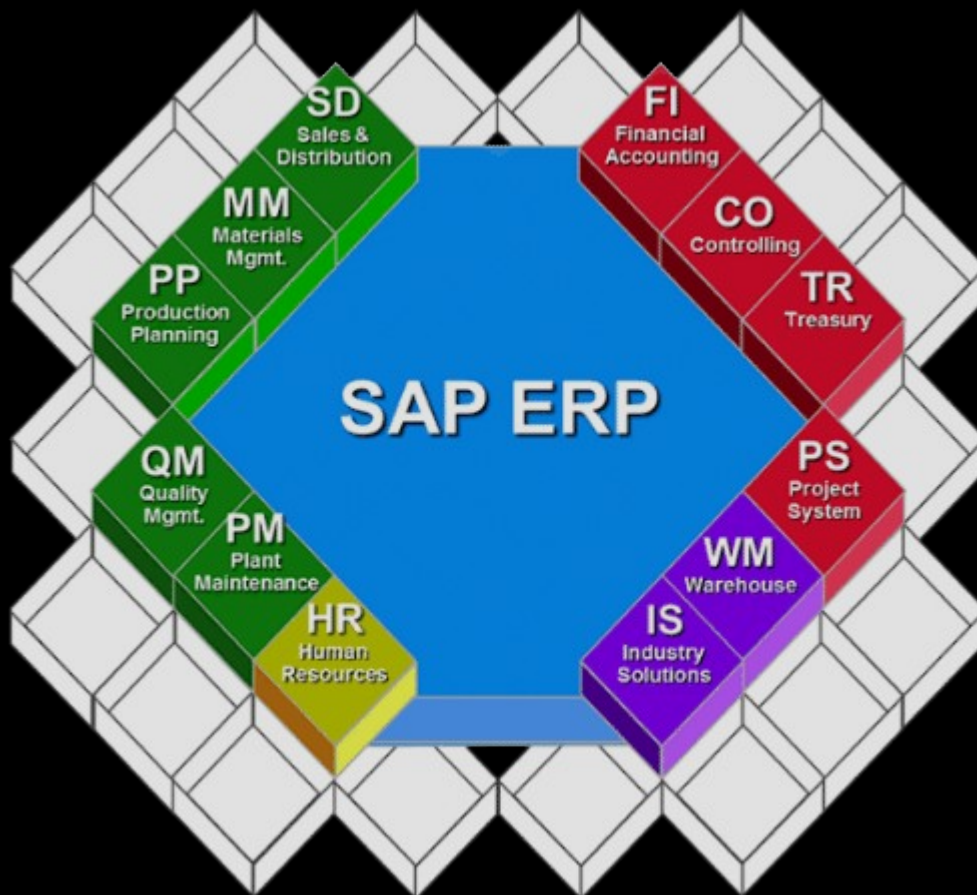
Al igual que Salesforce, cuenta con **diferentes planes** que se adaptan a todo tipo de empresas.

SISTEMA DE GESTIÓN EMPRESARIAL

TÉCNICO EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Módulos ERPs

Módulos SAP
(ERP propietario)



Estos son algunos de los módulos SAP más importantes:

Módulos **financieros**

Casi cualquier operación de los procesos de la empresa acaba teniendo alguna actividad en este módulo.

Se relaciona con los módulos de (entre otros):

SD (ventas y distribución),

MM (Manejo de **Materiales**),

PP (Planificación de **Producción**)

para obtener los resultados...

Algunos de sus **módulos** son:

- **Finanzas** - **FI**: Ayuda a controlar los datos que intervienen en las **operaciones financieras** como pueden ser el balance, la cuenta de resultados, contabilidad presupuestaria, apuntes contables, cuentas a pagar y a cobrar...

- **Contabilidad** - **CO (Controlling)**:

Análisis de la gestión interna de la empresa usando los datos de **costes y beneficio** de esta.

Se **planifican, reportan y monitorizan** las **operaciones del negocio**.

Se pueden configurar los **datos maestros** de tipos de coste, la contabilidad por centros de coste, el análisis de la rentabilidad...

Módulo de **recursos humanos**

Todo lo relacionado con el **capital humano** de una empresa: **Control** exhaustivo de sus **empleados**. Acceder a los **datos maestros de los empleados**, sus nóminas, vacaciones, gestión de la compensación, promociones, evaluación del desempeño, gestión de tiempos...

Módulos logísticos

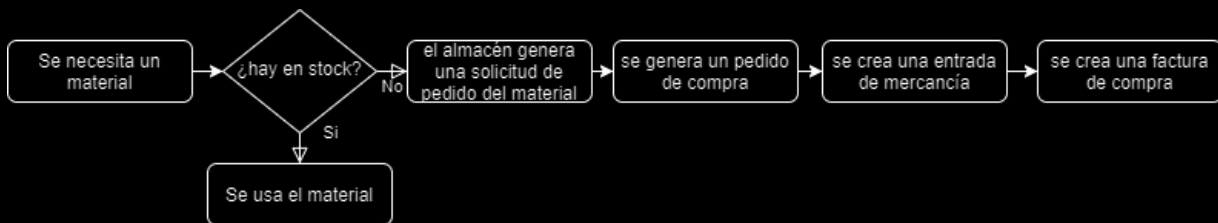
• **Gestión de materiales - MM (Materials Management)**: Gestiona el **stock** y la **entrada de materiales** a la empresa mediante **aprovisionamiento**.

Submódulos:

- Planificación de **compras**,
- Verificación de **facturas**,
- Gestión de **stock**,
- Maestro de **proveedores**,
- etc.

En **MM** podrían crean **flujos de compra** como el siguiente:

solicitud de pedido - pedido - entrada de mercancía - **factura** de compra.



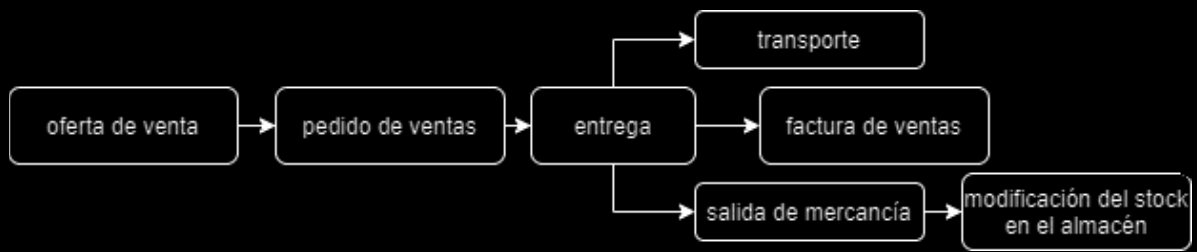
• **Ventas y distribución - SD (Sales and Distribution)**: Controla las fases del ciclo de venta y distribución de los productos y servicios de una empresa **hacia sus clientes**.

Integra acciones como:

- **Entrega** de mercancías
- **Facturar** pedidos de ventas,
- etc.

En **SD** podrían crean **flujos de compra** como el siguiente:

pedido de ventas - **entrega** de mercancías - **transporte** y **factura** de venta.



• **Planificación de la producción** – PP (Production Planning): Muy útil en **plantas de fabricación**, puesto que se usa para la **planificación y gestión** de la producción. Se incluyen **procesos completos de fabricación**:

- **Fabricación repetitiva,**
- Por **lotes,**
- De gestión de **cadenas de suministro**, etc.

Ejemplo de relación entre módulos:

1. **Ciente nos compra** el producto (**SD**),
2. Se crea automáticamente un **flujo de compra (MM)** para adquirir dicho producto de nuestro proveedor, y poder vendérselo al cliente.
3. La compra del producto al proveedor externo, y nuestra venta hacia el cliente **han generado movimientos** en el área de **finanzas (FI (facturas de compra y de venta))**.

Módulos Odoo (ERP libre)

Los módulos en los distintos ERPs

El número de módulos que puede tener un ERP varía de uno a otro. Normalmente disponen de algunos en común. Puede que difieran en el nombre, tengan distintos flujos, distinta interfaz, distintos campos... pero el concepto de ese módulo será el mismo.

Características de algunos módulos del ERP Odoo

Aplicaciones financieras

3 módulos:

- **Contabilidad:** Registrar **facturas** y controlar **pagos**.
- **Facturación:** Gestionar **contratos**, crear **facturas** y demás opciones que nos facilita facturar rápido.
- **Gastos:** Monitorizar **gastos** y volver a **facturarlos por lote** (si es necesario).

Aplicaciones de operaciones

7 módulos:

- **Compra:** **Automatizar** flujos de compras, licitar compras, listar precios y disponibilidad del producto en el proveedor, estadísticas de compra, etc.
- **Inventario:** Trazabilidad total del **stock**.
- **Proyecto:** Gestionar **proyectos** (metodología **Agile** si fuese necesario).
- Además, se cuenta con **Hojas de horas**, **Servicio de asistencia**, **Documentos** y **Servicio de campo**.

Aplicaciones de ventas

5 módulos:

- **CRM:** Contacto y captación de **clientes**.
- **Punto de venta:** **Gestionar pagos** en metálico o tarjeta, facturas, recibos, compra en línea, etc. **Compatible** con casi cualquier **hardware** (TPVs, PCs, etc.).
- **Ventas:** **Presupuestos** claros, **estrategias de precios** para maximizar ingresos, automatizando reglas para **calcular el precio por cliente** en función de sus condiciones, etc.
- Además, se cuenta con los módulos de: **Suscripciones** y **Alquiler**.

Aplicaciones de fabricación

4 módulos:

- **MRP (Material Requirement Planning):** **Fabricación**. Planificar **pedidos de fabricación** automáticamente.
- **PLM (Product Lifecycle Management):** Gestionar el **ciclo de vida** de los productos, desde la concepción y el **diseño** hasta la **producción**.
- **MRP Mantenimiento:** Automatiza un mantenimiento **preventivo**. Realizar pedidos de mantenimiento y planes de **control y estadística**.
- **Calidad:** Definir los planes de **control** de calidad y administrar **alertas de calidad...**

Aplicaciones de recursos humanos

7 módulos:

- **Contratación:** Organizar puestos **vacantes**, llevar un seguimiento de las **ofertas** de empleo y procesos de contratación.
- **Empleados:** Todo lo relacionado con empleados (**gastos, vacaciones, asistencias, bajas, evaluación**, etc.).
- **Recomendación de empleados:** Útil para llevar a cabo un programa de **referencias con recompensas**.
- **Aprobaciones:** Administrar fácilmente las **solicitudes de los empleados** (viajes de negocios, pago de dietas, etc.).
- Además, se cuenta con los módulos de: **Flota, Ausencias y Valoración**.

Aplicaciones de sitios webs

8 módulos:

- **Creador de sitios webs:** Crear **páginas webs** de un modo muy amigable y sin necesidad de tener conocimientos de programación.
- **Comercio electrónico:** **e-commerce**, crear una web desde donde **vender** sus productos integrando **pagos y envíos**.
- **Chat en directo:** **Chatear** con los **usuarios** de la web.
- **Citas:** Gestionar citas online.
- Además, se cuenta con los módulos de: **Blogs, Foro, Aprendizaje en línea y Eventos**.

Aplicaciones de comunicación

3 módulos:

- **Hablar**: Chatear en privado o en grupo con los **empleados** desde cualquier parte (**integrado con todos los módulos**). Además, se cuenta con: **Firma electrónica** y **Encuesta**.

Aplicaciones de marketing

4 módulos:

- **Automatización**: Crear **campañas** de **mercadotecnia** específicas y automáticas.
- **E-mail**: Marketing por correo electrónico (crear **campañas** o **segmentar** a los **clientes**).
- **SMS**: igual que E-mail, pero con SMS.
- **Social**: **Manejar** desde un mismo sitio las **redes sociales (Facebook, Twitter y LinkedIn)**, **chat** y **notificaciones push** (globos de información). Crear **campañas integradas con correos** electrónicos, **SMS**, etc.

Aplicación de customización

Módulo único:

- **Estudio de Odoo**: Crear **aplicaciones personalizadas**.

Dependencias entre módulos Odoo

INDEPENDIENTES	DEPENDIENTES	DEPENDE DE
CRM	CONTABILIDAD	FACTURACIÓN
WEB	FABRICACIÓN	INVENTARIO, FACTURACIÓN
PLANIFICACIÓN	SUSCRIPCIONES	FACTURACIÓN, VENTAS
FACTURACIÓN	CALIDAD	INVENTARIO
MARKETING ELECTRÓNICO	GESTIÓN DEL CICLO DE VIDA	FABRICACIÓN, INVENTARIO
AUSENCIAS	CONSOLIDACIÓN	CONTABILIDAD, FACTURACIÓN
DOCUMENTOS	APRENDIZAJE EN LÍNEA	WEB
PROYECTO	COMERCIO ELECTRÓNICO	WEB, FACTURACIÓN
CONTRATACIÓN	COMPRA	FACTURACIÓN
FIRMAR	GASTOS	FACTURACIÓN
STUDIO	RECOMENDACIONES DE EMPLEADOS	WEB, CONTRATACIÓN
CITAS	ALQUILER	FACTURACIÓN
CAJAS IoT	VENTAS	FACTURACIÓN
VALORACIÓN	PUNTOS DE VENTA	FACTURACIÓN, INVENTARIO
MANTENIMIENTO	HOJAS DE HORAS	PROYECTO
SERVICIO DE ASISTENCIA	EVENTOS	WEB
APROBACIONES	AUTOMATIZACIÓN DE MARKETING	MARKETING ELECTRÓNICO
MARKETING SOCIAL	SERVICIO DE CAMPO	PROYECTO, HOJAS DE HORAS

Ejemplo práctico de elección de módulos de un ERC Odoo para un pequeño negocio

Partimos del **módulo de punto de venta** que **requiere** disponer también de los **módulos de factura e inventario**.

Para la **gestión de las cuentas**, se podría contar con el **módulo de gastos** y con el de **contabilidad**.

Podríamos contar con el **módulo de compras** para automatizar el proceso de **compra** cuando algunos artículos bajen de un número de **stock** concreto o mantener contactos con **proveedores**...

Si el **número de empleados** creciese, podríamos usar el **módulo Empleados**.

Podríamos gestionar una **página web** con el **Creador de sitios webs**, con **citas online** (**módulo Citas**).

No quiere desembolsar mucho dinero. A medida que lo vaya necesitando, puede ir **haciendo uso de otros módulos** que se interconectan con los que ya tiene en uso (**CRM, etc.**).

<https://www.sap.com/index.html>

https://www.odoo.com/es_ES/

SISTEMA DE GESTIÓN EMPRESARIAL

TÉCNICO EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA

El proyecto de implantación de un ERP

Se verá:

Fases de la implantación de un ERP y la **documentación** en cada una de ellas.

Roles que intervienen en cada fase.

Implantación

Tendremos **tres opciones**:

Adaptar la empresa al ERP,

Adaptar el ERP a la empresa,

Una **mezcla**; es la **opción recomendada** (rara vez el estándar de un ERP se adapta a la perfección a la empresa, y es muy difícil cambiar por completo una empresa para adaptarla al ERP).

Implementación correcta: **ahorro en gastos** y/o aumento en **beneficios** para la empresa.

Implementación incorrecta: causará **muchos problemas**.

Es necesario disponer de una **metodología** para el **antes, el durante y el después** de la implantación del proyecto. El proyecto de implantación del ERP consta de varias **etapas o fases**, cada una de ellas con unas **tareas** y una **documentación** asociada:

1. **Planificación.**
2. **Análisis o consultoría.**
3. **Parametrización y diseño.**
4. **Pruebas.**

DESPLIEGUE:

5. **Formación.**
6. **Migración y arranque.**

POST IMPLANTACIÓN:

7. **Soporte y seguimiento.**
8. **Mantenimiento**

Planificación

Es el punto de partida del proyecto de implantación.

Buena planificación == **implantación de éxito**.

Estimaciones documentadas de:

El **objetivo**.

Recursos (tanto materiales como de trabajadores).

Tiempos.

Se indican los **criterios de éxito y cierre** del proyecto. Esta documentación es una especie de contrato entre la empresa cliente y la empresa implantadora, que está **dirigida y representada** para este trabajo por el **jefe de proyecto**. Es decir, al finalizar esta fase de planificación, se firma un **contrato** estipulando que, con unos recursos (tanto materiales como de trabajadores) y un **presupuesto concreto**, la **empresa implantadora se compromete** a realizar el **proyecto de implantación** antes de una **fecha concreta** y cumpliendo con unos **criterios** de éxito.

Es muy importante que queden bien **documentados y estipulados** los **requerimientos del proyecto**.

Con respecto al equipo de trabajo, se definen todos los **integrantes** con sus correspondientes **roles** bien detallados para que cada trabajador sepa cuáles son sus **responsabilidades**. De esta forma, podríamos encontrarnos con estos roles:

Empresa contratante:

1. **Gerente** (Expone requerimientos y acepta planificación).
2. **Usuarios líderes** (usuarios con experiencia de la empresa y además planifican).
3. **Usuarios finales** (los que harán uso del ERP).
4. **Administrador del sistema** (prepara el sistema para implantar e implementar el ERP).

Empresa contratada:

1. **Jefe de equipo** (planifica y coordina el proyecto y hace que se cumpla).
2. **Consultores** del ERP (conocen los módulos, y adaptan los parámetros del ERP a la empresa. Niveles: junior, senior, analistas...).
3. **Programadores** (programan necesidades de la empresa que no dispone el ERP. Niveles: junior, senior, analistas...)

Análisis y Consultoría (consulta con cliente)

Se detallan los **alcances específicos** del proyecto de implantación, llegando incluso a elaborar **diagramas** de flujo con los **procesos y procedimientos**. El objetivo final de esta etapa es **documentar el diseño funcional** del proyecto con los requerimientos y funcionalidades acordadas con el **cliente**. Este documento describe **qué se debe hacer en el proyecto** y es **realizado por los consultores o analistas** del ERP, que son personas que tienen experiencia en dicho software.

Para ello, una vez realizada la toma de requerimientos en la etapa anterior, los consultores **traducen** dichos **requerimientos** que el cliente ha expuesto con un lenguaje coloquial de su empresa **a requerimientos del ERP** con un **vocabulario de alto nivel**, **no** un vocabulario **técnico**.

Diseñado el documento funcional, será **validado por el cliente**, **ya que es lo que se va a implementar** en la siguiente fase (parametrización y diseño).

Si hubiese algún **error**, **mejor detectarlo en esta fase** que en la siguiente, donde ya se habría implementado y habría que echar atrás el proyecto y volver a implementarlo correctamente.

Se debe alcanzar un **entendimiento común** entre el **cliente** y el **proveedor** con respecto a cuáles son los procesos actuales de la empresa cliente y cómo van a impactar sobre ellos los requisitos del proyecto que se va a llevar a cabo.

Se debe crear también el **plan de pruebas**, las cuales, en la fase de pruebas deberán tener un **resultado satisfactorio** para **dar por bueno el proyecto**.

Parametrización y diseño

Se debe documentar el **diseño técnico**.

Si el diseño funcional describe qué se debe hacer en el proyecto, el diseño técnico describe **cómo hay que hacerlo**. Este documento tiene como base el diseño funcional y es creado por el **personal técnico** (los programadores). Consiste en **traducir a un lenguaje de bajo nivel** los procesos, requerimientos y procedimientos que se han indicado en el **diseño funcional**.

Tres niveles o entornos en los que trabajar:

1. De **Desarrollo**:
 1. Más bajo (**técnicos**).
 2. Permite implementar **cambios**.
 3. Pruebas **unitarias**.
2. De **Integración o pruebas**:
 1. Nivel intermedio (**analistas**).
 2. **No** permite implementar **cambios**.
 3. Pruebas **integrales**.
3. De **Producción**.
 1. Más alto (**cliente** final).
 2. Entorno real.
 3. **Datos reales**.
 4. Asegurarnos que todo va bien **antes de** la subida a **producción**.

Pruebas

Se **simulan los procesos de la empresa** a los que afecte la implementación del ERP. Las pruebas deben de ser **lo más reales y detalladas posibles**.

Debemos **validar todos los procesos** para evitar sorpresas en el entorno real que impliquen fallos graves en la empresa del cliente.

En proyectos grandes, se usa la figura del **tester** cuya finalidad es realizar **pruebas y buscar errores**.

Tipos de pruebas:

1. Pruebas **unitarias**.
2. Pruebas **integrales**.
3. Pruebas **finales o reales**.

Todas ellas:

documentadas (ya sea con **grabaciones** de pantalla o con **textos y capturas** de pantalla),
y con **resultado satisfactorio**,

para que los cambios pasen a la siguiente etapa: entorno de desarrollo al de producción.

Tipos de pruebas

Pruebas unitarias: las realizan los **técnicos** en el entorno de desarrollo para comprobar que los **cambios implementados** funcionan correctamente. Mientras las pruebas sean insatisfactorias, el técnico seguirá implementando sus cambios en el entorno de desarrollo. Una vez que estas pruebas son completadas, ya se podrán **transportar los cambios al entorno de integración**.

Pruebas integrales: la realizan los **analistas** en el **entorno de integración**. Son las del **plan de pruebas** que se documentó previamente en la **fase de análisis o consultoría**. Para poder poner en producción los cambios, estas pruebas han de ser satisfactorias en su totalidad. Si, por alguna razón, alguna prueba de la batería diera un **resultado insatisfactorio**, el analista se lo debería **comunicar al técnico** para que éste solucionara el problema. Una vez **solucionado**, lo volvería a **probar primero en desarrollo** (pruebas unitarias) y posteriormente **transportarlo a integración** para que el **analista** lo vuelva a probar.

Pruebas finales o reales: se realizan **directamente en el entorno de producción**. Éstas son realizadas por el **cliente final** (los **únicos que tendrán autorización** para manipular datos en dicho entorno).

DESPLIEGUE...

Formación

Se procede a la **capacitación de los usuarios finales** del cliente (incluyendo al **administrador del sistema**) en el ERP. Aún se está en el entorno de **integración o pruebas**. Es decir, esta fase se produce **antes de la subida a producción** para que, de esta forma, el **usuario final aprenda cómo operar** con el entorno ERP antes de que este esté funcionando en el entorno de producción (son **ellos los que tendrán que interactuar** con el ERP). Al ser un entorno de pruebas, los usuarios finales pueden plantear todas las **dudas** todas las que necesiten sin miedo a cometer errores, ya que **no se trata de un entorno real**.

Migración y arranque

Última fase de la implantación del ERP: la **puesta en marcha**. El ERP empieza a funcionar en el **entorno de producción** y el cliente ya hace un **uso completo** de la solución de **forma real**.

Es una fase muy intensa y estresante, puesto que los **errores** que se produzcan en esta fase son **críticos** y necesitan **solucionarse rápidamente**. Se puede sumar la inexperiencia del cliente final con el ERP y los posibles **errores humanos** que se cometan.

POST IMPLANTACIÓN...

Soporte y seguimiento

Durante un **tiempo** previamente **estipulado** en el contrato (etapa planificación), la empresa implantadora debe **dar soporte** mientras el cliente está haciendo uso del ERP en su entorno real.

Los **equipos**, tanto del cliente como de la empresa implantadora, deben estar **alineados**: **pulir posibles errores** reales que puedan surgir.

Si alguna encontrasen un **error**, deberán **reportarlo lo antes posible** para solucionarlo, porque en esta fase los errores ya sí **repercuten sobre un proceso real** del cliente que puede ocasionar problemas reales. Los **errores quedan documentados** también.

Seguimiento: comprobar que el cliente está haciendo un **uso correcto del ERP** implantado.

Pasado el tiempo estipulado en el contrato, se da por **concluido el proyecto de implantación**.

Mantenimiento

Tras terminar el proyecto, el cliente tiene la opción de contratar un **nuevo proyecto**, pero en este caso de **mantenimiento**.

La función de un proyecto de mantenimiento podría ser la de llevar a cabo **nuevas herramientas personalizadas** en el ERP, **modificaciones** del ERP, **corrección** de posibles errores que surjan, **etc**. Es decir, la finalidad del proyecto de mantenimiento es la de **adaptar aún más el ERP** a la empresa y solucionar posibles **problemas que surjan**.

<https://blog.corponet.com.mx/etapas-de-implementacion-de-un-erp-o-sistema-administrativo-integral>

<https://www.actiobp.com/etapas-de-un-proceso-de-implantacion-erp/>

<https://www.evaluandoerp.com/software-erp/implementar-erp/>

SISTEMA DE GESTIÓN EMPRESARIAL

TÉCNICO EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Requisitos de instalación de ERP-CRM SAP y Odoo

Analizaremos los requisitos necesarios para poder **instalar un ERP** (**SAP**, de código propietario y **Odoo**, de código libre), en un ordenador **local** o un **servidor**.

Será la primera toma de contacto con la instalación en sí (si nuestro **equipo cumple o no los requerimientos**).

SAP Business Suite 4 SAP HANA



La principal característica de SAP S/4 HANA es el uso de **BBDD en memoria** (RAM).

Los módulos ahora se llaman **LoB (Line of Business - Líneas de trabajo)**, y son 10:

- Finanzas
- Gestión de activos
- Ventas
- I+D
- Compras y aprovisionamiento
- Fabricación
- Marketing y Comercio
- Recursos Humanos
- Cadena de Suministro
- Servicios

SAP S/4 HANA

Requisitos para instalación en local

SAP dispone de una serie de socios que comercializan **hardware certificado** para su uso con SAP HANA, como pueden ser **Cisco, HP, Fujitsu o IBM**, entre otros.

Instalación de la versión **SAP HANA 2.0 Express Edition** en local:

Se puede instalar de **tres formas diferentes** (requisitos similares):

- 1- Mediante **instalable** (código binario) en **Linux**.
- 2- Mediante **máquina virtual** preconfigurada.
- 3- A través de **Docker**.

Requisitos SAP para cada una de las opciones de instalación:

¿Dispone su ordenador de **suficiente memoria RAM** para realizar su instalación en local?

Si: On-Premise (Local)

- Docker (Linux).
- **Virtual Machine** (Linux, OS X, Windows).
- Binary Install (Linux).

No: Cloud Providers (Nube)

- Google Cloud Platform.
- Microsoft Azure Cloud.
- Amazon Web Services (**AWS**).

Notamos la importancia de tener conocimientos de Linux para lograr la instalación en local de un ERP SAP.

Instalación mediante instalable

Es la **más compleja** (requiere **conocimientos de Linux**).

Requisitos **hardware** del sistema Linux:

- **16GB** RAM (recomendables **24GB**). Si se está usando **16GB**, **aumentar la cantidad de espacio de intercambio** entre el HDD y la RAM a 32GB.
- **120GB disco duro SSD**.
- Procesador con **dos núcleos** (recomendables **cuatro**).

Requisitos **software**:

- **JRE 8** o superior, versión de **64 bits**.
- El sistema operativo debería ser uno de los siguientes:
 - **SUSE Linux Enterprise Server** for SAP Applications, 12.1, 12.2, 12.3 (SPS 02 Rev 23 o superior).
 - **SUSE Linux Enterprise Server** for SAP Applications, **IBM Power Systems** (It - Little Endian), **12.1, 12.2**.
 - **Red Hat Enterprise Linux** for SAP Applications **7.2, 7.3** (SPS 02 Rev 21 o superior), **7.4** (SPS 02 Rev 23 o superior).
 - **Red Hat Enterprise Linux** for SAP Applications **for Power 7.3** (SPS 02 Rev 21 o superior), **7.4** (SPS 02 Rev 23 o superior).

Dependiendo de nuestros requerimientos, se puede optar por:

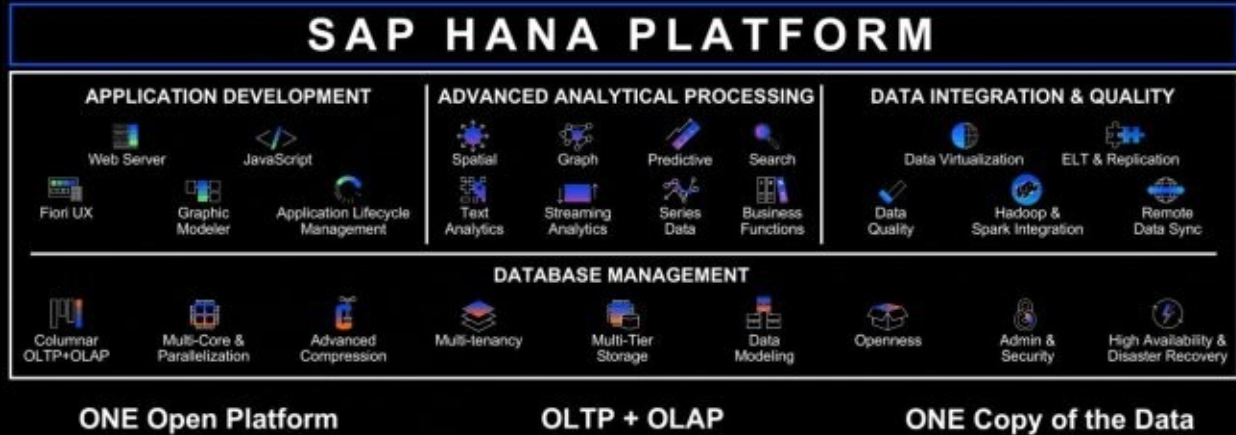
- Una versión que incluya **solo el servidor + XSC** (servicio de **aplicación extendida clásico**) + **AFL** (biblioteca de **funciones de aplicación**).
- Una versión que incluya el **servidor + aplicaciones + XSA** (servicio de **aplicación extendida avanzado**) + **Web IDE** (entorno de **desarrollo web** integrado) + SAP HANA **Cockpit**.

SAP HANA Platform

The in-memory data platform for digital business

— All Devices —

— SAP, ISV and Custom Applications —



La versión **Express Edition** no tiene:

- Data Quality
- Remote Data Sync
- Multi-Tier Storage
- High Availability & Disaster Recovery

Instalación mediante máquina virtual

Se instala una máquina virtual con el S.O. **SUSE Linux Enterprise Server (SLES) for SAP Applications 12 SP2** y con **SAP HANA 2.0 Express Edition preconfigurada** y lista para ser ejecutada. Al igual que con el instalador, con la máquina virtual también existen **dos posibilidades de descarga**, con **XSC** y con **XSA** (este paquete requiere **más** cantidad de memoria **RAM**).

Requisitos **software**:

- Tener instalado Java Runtime Environment (**JRE**) **8** o superior, la versión de **64 bits**.
- Tener instalado un monitor de **máquina virtual** (se recomienda **VMware**).
- El S.O. en este caso podría ser **Windows**, **OS X** o **Linux**.

Requisitos **hardware**:

- **8GB** de RAM solo para la **máquina virtual**.
- **16GB** de RAM para el **servidor**. Se **recomienda 24GB**.
- **120GB** de espacio libre en el **disco duro SSD**.
- Procesador con **dos núcleos**. Se **recomienda cuatro**. Si el procesador es **Intel** hay que chequear si soporta VT-x (**Virtualization Technology**). Los procesadores **AMD** soportan VT-x.

Instalación mediante Docker

Para usar esta opción, es necesario tener **conocimientos básicos de Docker**. En este caso las dos **opciones de instalación** son:

- Solo el **servidor**, no contiene XSC.
- **Servidor + XSA**.

Los requisitos **hardware** son los mismos, pero los **softwares** cambian (**solo** disponible para **Linux**):

Versiones Linux compatibles con SAP en **Docker**:

- openSUSE (openSUSE Leap): Enterprise**
- Ubuntu 17.04 (Zesty Zapus): Community, Enterprise**
- CentOS 7 (Core): Community, Enterprise**
- Debian 9 (Stretch): Community**
- Fedora 28 (Server Edition): Community**

Odoo

Hay **múltiples formas** de instalar Odoo:

- **Online**: la **más fácil**. Realmente no tenemos que instalar nada, solo hacer uso de él a través del **SaaS** (**Software as a Service** o Software como un servicio) desde un navegador web. En contra, dependeríamos de un **proveedor del servicio** con su correspondiente **coste económico**.
- **Instalador**: con este se **instalan los módulos necesarios** pudiendo usar Odoo para **largo plazo** realizando tareas de mantenimiento (como actualizaciones).
- Instalar el **código fuente**: lo cual proporciona una **mayor flexibilidad**. Es una buena opción para poder **desarrollar en los módulos** y puede usarse como base para un **despliegue en producción**.
- **Docker**: sería a través de un **contenedor Docker**. Existe una **imagen oficial** de Odoo disponible en la web de Docker.

Hay **dos versiones** diferentes de Odoo:

- La versión **comunitaria**: que es mantenida por la comunidad de **software libre** vía **GitHub**. El acceso al código de esta versión está disponible a través de GitHub (<https://github.com/odoo/>). Se puede actualizar a la versión Enterprise.
- La versión **empresarial**: que es mantenida por el proveedor oficial belga **Odoo S.A.** El acceso a esta versión está restringido a **clientes y socios**.

Requisitos hardware de Odoo

Especifica unas fórmulas para calcular el **número de usuarios** y el tamaño de la memoria **RAM**.

Para calcular el **número de trabajadores** que pueden usar Odoo a la vez:

Regla de oro: (#CPU * 2) + 1.

Los trabajadores de Cron necesitan **CPU**.

1 trabajador ~ = 6 usuarios concurrentes.

Para calcular el **tamaño de la memoria**:

El **20%** de las solicitudes que se hacen en Odoo son **pesadas**. Un trabajador “**pesado**” puede llegar a consumir 1GB (**1024MB**) de RAM.

El **80%** s de las solicitudes son más **simples**. Estiman que, Un trabajador “**ligero**” puede llegar a consumir **150MB** de RAM.

Por lo tanto, la fórmula para calcular la RAM quedaría como sigue:

$$\text{RAM} = \text{trabajadores} * (\text{80\% con 150MB} + \text{20\% con 1GB})$$

Ejemplo **cálculos hardware en Odoo**:

Tenemos 50 usuarios concurrentes: ¿cuánta **RAM** necesitamos?

1 trabajador = 6 usuarios concurrentes → $50 / 6 = 8.33 \rightarrow 9$ trabajadores

$\text{RAM} = 9 * (0.8 * 150\text{MB} + 0.2 * 1024\text{MB}) \rightarrow 2923.2\text{MB} \rightarrow 3\text{GB}$ aproximadamente como máximo.

Estimación de CPU y RAM según el nº de trabajadores

Trabajadores	CPUs	Ram
5	2	2GB
20	4	8GB
100+	2-8	32GB

Requisitos software Odoo en la versión online (SaaS)

Sería necesario disponer de un **PC** con un **navegador** web instalado. En la web oficial solo especifican los **navegadores soportados** en **versiones anteriores** a la actual (13):

- **Odoo 8: IE9**, Mozilla Firefox, Google Chrome, Safari, Microsoft Edge.
- **Odoo 9: IE11**, Mozilla Firefox, Google Chrome, Safari, Microsoft Edge.
- **Odoo 10:** Mozilla Firefox, Google Chrome, Safari, Microsoft Edge.

*Desde la web de Odoo se ofrece una **versión de demostración** donde se le permite al usuario interactuar y probar sin compromiso todos los módulos del ERP **durante unas horas** con el **entorno** que ofrecen de **SaaS** (Software as a Service o Software como un Servicio).*

*Si tras probarla, nos decidiésemos finalmente por hacer **uso de la versión SaaS**, indicar que esta versión está totalmente administrada y migrada por Odoo S.A. Proporciona instancias privadas y comienza siendo **gratuita si se usan pocos módulos** y para **usuarios ilimitados**. Por ello, podríamos usarla gratuitamente indefinidamente **como versión final** siempre y cuando **no se necesiten más usuarios ni módulos**. No obstante, esto puede suponer un handicap a la hora de realizar un despliegue 'profesional' en cliente, ya que esta gratuidad estaría **limitando módulos, usuarios y servicios**. En el caso de querer **migrar a otra versión sin limitaciones**, se tendría que hacer a través de un **plan de pago** cuyo importe dependerá del **número de módulos**.*

Requisitos software Odoo en la versión instalador

Odoo proporciona instaladores para las versiones comunitaria y empresarial para los siguientes S.O.:

- **Windows.**
- Distribuciones basadas en **Debian** (Debian, Ubuntu...).
- Distribuciones basadas en **RPM** (RedHat Package Manager) (Fedora, CentOS, RHEL...).

Otro requisito software en cualquiera de los S.O. es tener instalado un **servidor PostgreSQL** en el mismo **host** donde se encuentre la instancia de Odoo.

PostgreSQL: Es un potente **Sistema de Base de Datos** objeto relacional de código abierto que usa y amplía el lenguaje **SQL** combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas.

Además de ser **gratuito** y de **código abierto**, es altamente **extensible**;

*Por ejemplo, se pueden definir **tipos de datos propios**, crear **funciones personalizadas**, incluso escribir códigos desde diferentes **lenguajes de programación**, **recompilar** su base de datos. Además se ejecuta en todos los **principales sistemas operativos**, por lo que no sorprende que se haya convertido en la base de datos relacional de código abierto **elegida por muchas personas y organizaciones**.*

Se ha ganado una sólida reputación por su **arquitectura comprobada**, **confiabilidad**, **integridad** de datos, un conjunto de **características robustas**, **extensibilidad** y la **dedicación** de la comunidad de código abierto para ofrecer **soluciones innovadoras** y de **alto rendimiento**.

Dispone de muchas **características** destinadas a ayudar a los **desarrolladores a crear aplicaciones**, a los **administradores** a proteger la **integridad** de los datos y a crear entornos **tolerantes a fallos** y **ayudarlo a administrar sus datos** sin importar cuan grande o pequeño sea el conjunto de datos.

Requisitos software Odoo en la versión de código fuente

Los requisitos de S.O. son los **mismos** que en la versión de **instalador**, **también** es necesario tener instalado el sistema gestor de base de datos **PostgreSQL**.

Hay **dos posibilidades** de obtener el código fuente:

- Archivo **zip**.
- A través de un **repositorio Git**: es necesario tener instalado Git, evidentemente.

Es necesario disponer de una versión de **Phyton 3.6** (puesto que sus **módulos están implementados en dicho lenguaje**) o superior, y el **compilador de Visual Studio** (requiere herramientas de **compilación de C++**).

g

Requisitos software Odoo en la versión Docker

Para usar esta opción, simplemente es necesario tener **instalado Docker** y poseer **conocimientos** de dicho software.

SISTEMAS DE GESTIÓN EMPRESARIAL

TÉCNICO EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Tipos de instalaciones de sistemas ERP-CRM

Instalación de Odoo

Tipos de instalación

Instalación mediante SaaS

La instalación mediante **SaaS** (Software as a Service o Software como un Servicio) a través del proveedor oficial u otro es la que presenta **mayor dependencia**, puesto que se hace uso de un **servidor propiedad del proveedor** donde está instalado el ERP. Este ERP se presenta como un servicio al que se accede **mediante pago** y es el proveedor quien gestiona las versiones. Esta opción presenta la ventaja de que nos podremos **despreocupar** tanto de la **instalación** del SW como del **mantenimiento del HW**. Basta con conectarse y empezar a usarlo.

Instalación en un hosting

Este caso es parecido a la opción anterior, pero **más difícil** de implementar. Es decir, vamos a hacer uso del ERP que también va a estar alojado en un **servidor**, que puede ser del proveedor oficial del ERP, nuestro, o de un tercero, pero con la desventaja de que **si el proveedor del hosting no ofrece el ERP preinstalado**, tendremos que **instalarlo**. Además, dependiendo de si el servidor es nuestro o de algún proveedor, y de si el SW lo tenemos que instalar nosotros o no, tendremos que **preocuparnos tanto del HW como del SW**. Eso sí, pudiendo **instalar cualquier módulo y versión** que necesitemos.

Instalación en un servidor local

Este tipo de instalación es la que **mayor independencia** del proveedor ofrece, pero, en este caso, tendremos que hacernos cargo tanto de la **configuración hardware como software**. Esta opción está enfocada a empresas que no quieran o no puedan tener el ERP conectado a internet. La ventaja que tiene esta opción es que **es la más económica y flexible**.

Tipos de licencias software

“Se trata de un **contrato** entre el desarrollador de un software sometido a **propiedad intelectual** y a **derechos de autor** y el usuario, en el cual se definen con precisión los **derechos y deberes** de ambas partes. Es el **desarrollador**, o aquel a quien este haya cedido los derechos de explotación, quien **elige la licencia** según la cual se distribuye el software.”

Labrador, R. M. G. (2012).

Además de los tipos de licencia de **software libre**, software de **código abierto** y **software propietario** nos encontramos con:

- Software de **dominio público**: no tiene derechos de autor.
- Software bajo **licencias laxas o permisivas**: es aquel que permite **usar el código** de cualquier forma.
- Software con **copyleft**: se trata de un software libre cuyos términos de distribución **no permiten** a los redistribuidores agregar ninguna **restricción adicional**. Por lo tanto, en el software libre **sin copyleft sí pueden agregar restricciones**.
- Software con **licencia GPL** (General Public License o licencia pública general) de **GNU**: conjunto de cláusulas de distribución para publicar **programas con copyleft**. El núcleo de Odoo, por ejemplo, tiene licencia LGPLv3.
- Software **comercial**: el desarrollado por una empresa que pretende **ganar dinero** por su uso.

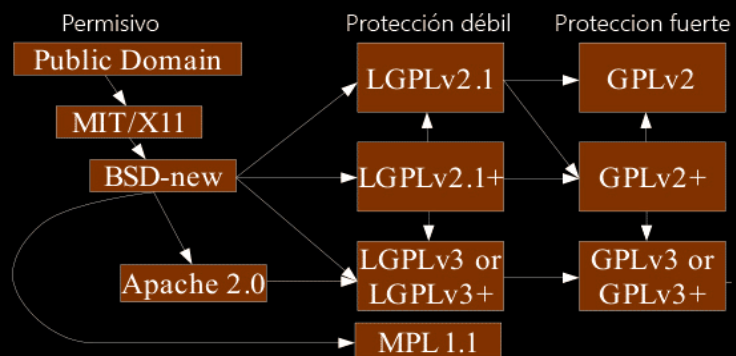
Software libre:

Software de fuente abierta:

- Software de dominio público (con fuente).
- Software bajo licencias laxas o permisivas.
- Software bajo copyleft.
 - Software bajo GLP.

Software privativo:

- Software de dominio público (sin código fuente).
- Shareware (software de descarga gratuita).



Para instalar una versión de **SAP** en **local** y realizar en ellas **tareas de Big Data**, necesitamos que la versión que instalemos permita integración con **Hadoop y Spark**, y, además, se desea que esta **integración** sea con **todas las aplicaciones**, no solo con las de SAP.

Por tanto, se prefiere que venga de base en la versión que usemos, **que no sea opcional la integración con Hadoop y Spark**.

La única versión que trae integración con Hadoop y Spark de base y está optimizada para el uso de Big Data es la versión **Enterprise Edition**.

	SAP HANA, enterprise edition	SAP HANA, standard edition	SAP HANA, express edition	SAP HANA, runtime edition
Integración Hadoop y Spark	Sí	Opcional	Sí	Solo para aplicaciones SAP
Optimizado para Big Data.	Sí	Sí	No	Sí

Opciones de implantación de SAP (cuatro, con posibilidad de ser on-premise (en local) o cloud (en la nube)):

- **Runtime Edition**: ofrece una plataforma **restringida**, exclusivamente diseñada para ejecutar aplicaciones SAP con un uso **limitado** de **características avanzadas**.

Las **otras tres opciones** pueden usar **aplicaciones SAP y no SAP**:

- **Express Edition**: ofrece un paquete **simplificado gratuito** con **hasta 32GB** de uso en memoria, que puede ser **incrementada** previo **pago**.
- **Standar Edition**: la cual ofrece una **base de datos** para casos de **usos innovadores**, con opciones flexibles para poder expresar al máximo las **funciones avanzadas**.
- **Enterprise Edition**: se trata de una plataforma **sin restricciones** con la finalidad de apoyar los esfuerzos en la **innovación** y el uso de aplicaciones SAP en **entornos híbridos modernos**.

Tutoriales SAP para **instalar** fácilmente la versión que se requiera:

- Para **versión instalable**, se podría seguir los siguientes:

<https://developers.sap.com/group.hxe-install-binary.html>,

<https://developers.sap.com/group.hxe-install-binary-xsa.html>

- Para **máquina virtual**, los siguientes:

<https://developers.sap.com/group.hxe-install-vm.html>,

<https://developers.sap.com/group.hxe-install-vm-xsa.html>

- para **Docker**, el siguiente:

<https://developers.sap.com/tutorials/hxe-ua-install-using-docker.html>

La opción cloud

Si nos decantásemos por usar la opción **Cloud** o nube, existen **2 posibles opciones**:

SAP HANA Cloud: que proporciona acceso a SAP HANA con las últimas **opciones de configuración** de una instancia SAP HANA en la nube, y con la posibilidad de **expandir** el tamaño de la **memoria** sin interrupciones.

SAP HANA Enterprise Cloud: que implementa y administra una instancia para SAP HANA en una **nube empresarial** administrada por SAP. Esto nos ofrece la opción de usar nuestra **propia licencia local** o comprar **licencias basadas en suscripciones**.

Además, nos dan la flexibilidad de **implementar SAP HANA en la nube o multinube** que el usuario elija, recomendando algunos **proveedores** como pueden ser:

SAP HANA en **Amazon Web Services**.

SAP HANA en **Google Cloud Platform** y SAP HANA, express edition en **Google Cloud Launcher**.

SAP HANA en **IBM Cloud**.

SAP HANA en **Microsoft Azure**.

SAP HANA One en **Alibaba Cloud** (AliCloud)

Opciones de implantación en Odoo

Tal y como vimos en el tema anterior, hay múltiples formas de instalar Odoo:

- online (SaaS),
- instalador,
- instalar el código fuente,
- Docker.

Además de dos versiones diferentes: la **versión comunitaria** y la **versión empresarial**.

Versión online (SaaS)

Desde la web de Odoo, se ofrece una **versión de demostración** donde se le permite al usuario interactuar y probar sin compromiso los módulos durante unas horas, con el entorno **SaaS** que ofrecen.

Si nos decidiésemos finalmente por hacer uso de la versión SaaS, tenemos que comentar que esta versión está totalmente administrada y migrada por Odoo S.A. Proporciona instancias privadas y comienza siendo **gratuita si se usan pocos módulos** y para **usuarios ilimitados**, por lo que es ideal para descubrir la versión SaaS de Odoo, aunque podríamos usarla como versión final ,siempre y cuando, no se necesiten más usuarios ni módulos. Si se necesitan **más módulos** o más **usuarios** se tiene que abonar una **mensualidad** que depende del número de módulos.

Tanto en el uso de la versión demo como de la versión SaaS, simplemente, es necesario disponer de un navegador web, no es necesario instalar nada en nuestro PC o servidor.

Versión con instalador

Tal y como comentamos en el tema anterior, Odoo proporciona **instaladores** para las **versiones comunitaria y empresarial**, tanto para **Windows** como para distribuciones basadas en **Debian** y **RPM**. Para poder usar la **versión empresarial**, es necesario **ser cliente o socio**.

Versión instalador en Windows

Basta con descargarse el instalador y seguir los pasos. Instala todo lo necesario para ejecutar Odoo en Windows.

Versión instalador en Debian y RPM

Es necesario instalar un **servidor PostgreSQL** en el mismo host donde se encuentre la instancia de Odoo.

Para ello, en distribuciones **Debian**, se puede hacer uso del siguiente comando:

```
$ sudo apt install postgresql -y
```

Mientras que en **RPM**, tras asegurarnos que el comando “sudo” está disponible y configurado correctamente, se usa:

```
$ sudo dnf install -y postgresql-server
```

```
$ sudo postgresql-setup --initdb --unit postgresql
```

```
$ sudo systemctl enable postgresql
```

```
$ sudo systemctl start postgresql
```

Una vez instalado el servidor, podemos instalar el ERP de dos formas:

1. **Mediante un repositorio** que ofrece Odoo S.A. para instalar la versión comunitaria.

En las distribuciones **Debian** y Ubuntu ejecutando los siguientes comandos como root:

```
# wget -O - https://nightly.odoo.com/odoo.key | apt-key add -  
  
# echo "deb http://nightly.odoo.com/13.0/nightly/deb/ ." >>  
/etc/apt/sources.list.d/odoo.list  
  
# apt-get update && apt-get install odoo
```

Y en **RPM**:

```
$ sudo dnf config-manager  
--add-repo=https://nightly.odoo.com/13.0/nightly/rpm/odoo.repo  
  
$ sudo dnf install -y odoo
```

```
$ sudo systemctl enable odoo
```

```
$ sudo systemctl start odoo
```

2. Mediante **paquetes “deb”** con los que se puede instalar tanto la versión comunitaria como la empresarial.

Para ello, hay que **descargarse los paquetes** desde su página oficial y en Debian, a continuación, ejecutar los siguientes comandos como root:

```
# dpkg -i <path_to_installation_package>
```

```
# apt-get install -f
```

```
# dpkg -i <path_to_installation_package>
```

Y para **paquetes ‘rpm’**:

```
$ sudo dnf localinstall odoo_13.0.latest.noarch.rpm
```

```
$ sudo systemctl enable odoo
```

```
$ sudo systemctl start odoo
```

Ambas opciones instalan Odoo como un servicio, crean el usuario en PostgreSQL y arrancan automáticamente el servidor.

Versión instalador de código fuente

La “instalación” del código fuente no se trata realmente de instalar Odoo, sino de ejecutarlo desde el propio sistema. Esta modalidad suele ser **más conveniente para los desarrolladores**, puesto que hace el código más accesible.

Esta versión la podremos hacer desde Windows, Linux y Mac OS:

Windows: Hay dos posibilidades de obtener el código fuente:

1. Archivo zip: desde la web de Odoo, GitHub o servidor Nightly.
2. A través de un repositorio Git ejecutando:

```
C:\> git clone https://github.com/odoo/odoo.git
```

Pero, para poder ejecutarlo, debemos tener una versión de **Phyton 3.6 o superior**, el sistema gestor de base de datos **PostgreSQL** y el **compilador de Visual Studio**, concretamente, las herramientas de **compilación de C++**. Esto último, es necesario para **instalar las dependencias** que se encuentran listadas en el archivo denominado **requirements.txt** que se aloja en la **raíz del directorio Odoo**:

```
C:\> cd \CommunityPath
```

```
C:\> pip install setuptools wheel
```

```
C:\> pip install -r requirements.txt
```

Con todo esto ya podríamos pasar a ejecutar Odoo usando la instrucción odoo-bin (que se encuentra en la raíz del directorio Odoo) en la interfaz de línea de comandos del servidor:

```
C:\> cd CommunityPath/
```

```
C:\> python odoo-bin -r dbuser -w dbpassword --addons-path=addons -d mydb
```

Donde dbuser es el usuario de PostgreSQL, dbpassword su contraseña y mydb es el servidor por defecto (localhost:8069).

Linux y Mac OS: Sería **igual**, solo que para descargar e **instalar** PostgreSQL se hace uso del **gestor de paquetes**.

Además, antes de instalar las dependencias de Python para Odoo, es necesario instalar las dependencias nativas, herramientas de desarrollo, y otras librerías usando el siguiente código:

```
$ sudo apt install python3-dev libxml2-dev libxslt1-dev libldap2-dev libsasl2-dev
```


SISTEMAS DE GESTIÓN EMPRESARIAL

TÉCNICO EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA

El flujo de compra-venta

Odoo

Usaremos los módulos de Venta, Contabilidad, Inventariado y Compra de Odoo.

Simularemos que tenemos una empresa de venta de muebles, la cual tiene Odoo instalada, procediendo a realizar un flujo de compra-venta básico en la misma.

Somos los jefes de una compañía de muebles que, para hacer crecer nuestra empresa, hemos instalado recientemente Odoo.

El principal negocio de nuestra empresa es el de **comprar al por mayor mobiliario** a nuestros proveedores a nivel mundial y luego **venderlos al por menor a clientes**, ya sea vía online o en nuestras tiendas. Para ello disponemos de almacenes donde guardar el stock y desde donde distribuir a las tiendas.

Por ello, el flujo de **compra-venta** es una de las principales utilidades que usaremos en Odoo.

Necesitaremos, al menos, tener instalados los siguientes módulos:

El módulo de Ventas.

El módulo de Contabilidad.

El módulo de Inventario.

El módulo de Compras.

El flujo de compra-venta que llevaremos a cabo en este tema se puede considerar un flujo de compra-venta **básico**, puesto que **no recoge problemas** de que un pedido no llegue, se devuelvan pedidos, facturas rectificativas, etc. Es decir, existen muchísimas posibles variables. En este tema analizaremos los siguientes pasos:

- **Crear** un producto.
- **Comprar** un producto.
- **Recibir** un producto.
- Controlar la **facturación**.
- **Vender** un producto.
- **Entregar** el producto al cliente.
- **Facturar** al cliente.
- El registro de **pagos**.
- Analizar los **beneficios**.

Crear un producto

El jefe de compras nos indica que está interesado en comprarle a uno de nuestros proveedores un escritorio que cuesta 140€, pero que al comprar 10 o más unidades, su precio será de 100€. Además, nos indica que luego podremos venderlo a 220€ cada uno. Necesitamos crear el producto en nuestro catálogo. Para ello, abriremos el **módulo de Compras** y entraremos en **productos**:

A continuación, pulsaremos sobre el botón “**Crear**” y crearemos el **producto**. Nos interesa informar de las siguientes características:

- **Nombre** del producto: en nuestro caso “Escritorio” .
- **Imagen**: podremos cargar una imagen.
- **Puede ser vendido**: marcando este checkbox el producto puede ser vendido. En nuestro caso, lo marcaremos.
- **Puede ser comprado**: marcando este checkbox el producto puede ser comprado. En nuestro caso, lo marcaremos.
- Pestaña de **información general**: aquí debemos definir las características generales del producto.
 - ◇ **Tipo de producto**: en nuestro caso es almacenable.
 - **Consumible**: un producto consumible es un producto para el que no se gestionan las existencias.
 - **Servicio**: un servicio es un “producto” no material, por ejemplo, las horas de mano de obra.
 - **Almacenable**: un producto almacenable es un producto para el que se gestiona stock. La aplicación de inventario debe estar instalada.
 - ◇ **Precio de venta**: precio al que le venderemos el producto a nuestros clientes. En nuestro caso, 220€.
 - ◇ **Impuestos cliente**: en nuestro caso, el 21% de IVA.
 - ◇ **Coste**: a priori, en nuestro caso son 140€.
- Pestaña de **compras**: aquí debemos definir una lista con el precio de compra del producto en uno o varios proveedores. En nuestro caso, como dependiendo de la cantidad tiene un precio u otro, debemos agregar una línea para la cantidad 1 (con un precio) y otra para la cantidad 10 (con otro precio). Nuestro proveedor es “Azure Interior” .

Al guardar, ya podremos ver el producto en nuestra lista de productos, pero, como podemos apreciar en la imagen, con stock 0.

Comprar un producto

Como vimos en el punto anterior, una vez que creamos el producto, este aparece en nuestra lista de productos, pero con stock 0. En este punto aumentaremos su stock haciendo una compra de 10 escritorios. Para ello, en el módulo de Compras, pulsaremos sobre **Pedidos, Solicitudes de presupuesto** y sobre **“Crear”**.

Informaremos como proveedor a “Azure Interior” y agregaremos el producto “Escritorio” que creamos en el listado de productos de la pestaña “productos”. Aquí podremos observar que si pedimos de 1 a 9 escritorios, su precio es 140€, pero si pedimos 10 o más, su precio es de 100€.

Una vez indicado el proveedor, el producto y la cantidad, ya solo nos queda pulsar sobre el botón de **“Confirmar pedido”**. Esto genera la orden de compra, en nuestro caso, la P00008.

Pero si volvemos a la lista de productos, observamos que este **aún tiene stock 0**.

Esto es debido a que nosotros hemos hecho la orden de compra, pero **aún no hemos recibido el producto**. En el siguiente punto veremos cómo **confirmar la recepción del producto**.

El abastecimiento automático

Odoo permite crear reglas de stock mínimo con la intención de que nunca nos quedemos sin existencia.

Es útil sobre todo para casos en los que haya una alta demanda del producto o que el producto tenga un tamaño muy grande y requieran gran cantidad de espacio de almacenamiento. Esta automatización nos ayuda a desentendernos del trabajo constante de abastecimiento de este tipo de productos para poder centrarnos en otros.

Para realizar una regla de stock mínimo, haremos uso del **módulo de Compras**. En el módulo de Compras, entraremos en la lista de productos y, concretamente, al producto al que le queramos crear la regla.

A continuación, haremos click sobre la pestaña de **“Reglas de abastecimiento”**.

Seguidamente pulsaremos sobre “Crear” y cumplimentaremos los campos de la pantalla según como necesitamos el pedido, teniendo en cuenta lo que significa cada uno:

- **Producto:** en este caso el “Escritorio”.
- **Cantidad mínima:** es con la que el sistema activa una nueva orden para la reposición, en este caso, es 5.
- **Cantidad máxima:** la cantidad máxima que debe haber en stock, en nuestro caso, indicaremos 10.
- **Múltiplo de la cantidad:** el **menor número** de producto que **se pueden pedir**. En nuestro caso se pueden de 1 en 1.
- **Compañía:** a qué almacén se hace el pedido.
- **Plazo de entrega:** cuanto tardan en entregárnoslo.

A partir de ahora, cada vez que el stock alcance la cantidad mínima, se crea una orden de compra de forma automática para alcanzar la cantidad máxima, pero teniendo en cuenta que las reposiciones se llevan a cabo cuando el planificador en el módulo de Inventario se ejecute. Por defecto en Odoo, los **cambios** se ejecutan todas las **noches a las 00:00**, pero se **puede ejecutar manualmente** desde el módulo de **Inventario** pasamos a **Operaciones** y **Ejecutar planificador**.

Recibir un producto

En este punto, suponemos que el fabricante ya nos ha hecho llegar el material que hemos comprado, por lo que veremos cómo **confirmar la recepción** de este pedido.

Esta acción debemos hacerla desde el **módulo de Inventario**. Al abrirlo, en la pantalla principal, nos aparecen, entre otras cosas, las **recepciones pendientes de validar** (en nuestro caso, son cuatro, de las que una de ellas son los 10 escritorios que hemos pedido):

Al pulsar sobre **“a procesar”**, veremos el listado con los cuatro pedidos de compra pendiente de recepción y sus estados. Pero a nosotros el que nos interesa es el del proveedor “Azure Interior”, concretamente, el que tiene como documento de origen la orden de compra P00008 que generamos anteriormente. Al pulsar sobre dicha orden, veremos, en la parte superior derecha (como en muchas otras pantallas nos encontraremos), un flujo con los estados del objeto que estemos tratando, resaltando el estado actual; en este caso, la recepción del pedido está **preparada**:

Para confirmar la recepción, debemos editarla para informar en la columna “Hecho” el número de artículos que hemos recibido, que debe coincidir con el que hemos pedido, en este caso, son 10, y **pulsar sobre validar**.

En ese momento, veremos cómo el status pasa a **“Hecho”**.

Si ahora volvemos a la lista de productos, observamos que este **ya tiene un** stock de 10 unidades.

Control de facturación

Para llevar un correcto control de las cuentas, es necesario **registrar la factura** que el proveedor “Azure Interior” nos ha enviado por la compra de los 10 escritorios que le hemos hecho. Para ello, hacemos uso del **módulo de “Contabilidad”**. Al abrirlo, pulsamos sobre **“Proveedores, Facturas”** y, a continuación, sobre **“Crear/Nuevo”**.

Al seleccionar en el campo **“Proveedor”** a “Azure Interior”, en el desplegable “Completar automáticamente”, nos aparecen las facturas de dicho proveedor, en nuestro caso, la referente a la orden de compra P00008 que estamos tratando en este flujo de compra...

Si seleccionamos el pedido de compra P00008, entonces, **se autocompletan los campos** con la información expuesta en dicho pedido...

En este momento, debemos comprobar que la **factura que nos genera** tenga el **mismo** importe que la que nos envía el **proveedor**, y si es igual, entonces, solo tenemos que guardarla. Si hubiese algún **descuadre** debemos ponernos en **contacto con el proveedor** para que nos **envíe la factura correctamente**.

Ya con el producto en nuestro stock y con la factura almacenada, **estamos a la espera** de que nuestros clientes nos hagan un **pedido**.

Venta y entrega de un producto

Vender un producto

A los pocos días de actualizar el stock de nuestro artículo “Escritorio”, recibimos nuestro primer pedido de dicho artículo. El cliente “Miguel” nos ha pedido que le enviemos por correo un presupuesto de 2 unidades:

En el módulo “**Ventas**” vamos a **pedidos** → **presupuesto** → **Crear y editar “Miguel”**, si aún no está **registrado**.

Y rellenamos algunos datos:

Nombre del cliente...

Y le ponemos un **correo electrónico** (esto es necesario).

Agregamos el producto.

En **cantidad**, tenemos una ayuda donde nos aparece el stock actual.

Hasta aquí estamos en el estado de **presupuesto**...

Para pasar a **presupuesto enviado**:

Pulsamos en **enviar por correo electrónico**.

Y nos aparecerá una **plantilla de correo** para enviar al comprador, con datos como presupuesto, y número de presupuesto, que adjunta el pdf con datos del presupuesto... pulsamos **enviar** y le enviamos el correo.

Hasta ahora el estado es **presupuesto enviado**.

Quando el cliente acepte el presupuesto, volvemos al mismo sitio y pulsamos el botón de **confirmar**.

Y entonces se pasa a estado de “**pedido de venta**” que aún no está en manos del cliente...

A continuación veremos qué hacer para que llegue al cliente el pedido.

Entregar el producto al cliente

Una vez aceptado el presupuesto por parte del cliente y tramitado el pedido de ventas, ahora toca enviarlo. Para ello, en el **módulo de Inventario**, si cuando hacemos un pedido de compra nos fijamos en “Recepciones”, ahora, al ser un pedido de ventas, deberemos hacerlo sobre “**Órdenes de entrega**” (actualmente es Expediciones).

Al pulsar sobre “**Procesar**”, nos aparece el pedido de ventas S00023 que creamos en el paso anterior con estado “**Preparado**”.

Al entrar en dicho pedido, tenemos que **editar la cantidad “Hecho”** con la **cantidad que le enviamos** (que debe ser igual a la cantidad demandada) y pulsar sobre “**Validar**”. De esta forma, el pedido de venta es enviado al cliente y su **estado** pasa a “**Hecho**”.

Cancelando un presupuesto

Un cliente nos ha pedido que le enviemos un presupuesto de 1 unidad de “Gabinete grande”, un producto que tiene un coste de 320€. Al recibirlo, lo ve excesivo y decide que no nos lo va a pedir.

El presupuesto, además de ser “Confirmado” puede ser “**Cancelado**”. Para ello, es necesario abrir el módulo de **Ventas**, dirigirnos al **listado de presupuestos** y seleccionar el **presupuesto** concreto. En nuestro caso, es el S00025. Al abrirlo, en la parte superior derecha, podemos ver que el **estado actual** es “**Presupuesto enviado**”, por lo que **aún no se ha creado el pedido de ventas**.

En la parte de abajo, podemos ver los pasos que se han dado en este presupuesto hasta ahora (se ha creado el presupuesto y se ha enviado el e-mail al cliente).

Para **cancelarlo**, basta con pulsar el botón de “**Cancelar**”. En ese momento, el estado pasa al estado “Cancelado”.

Y podremos verlo en el listado de pasos inferior.

Facturación y registro de pagos

Facturación al cliente

Para enviarle la factura al cliente, lo haremos desde el **módulo de Ventas**. Al entrar en el pedido de ventas, veremos que ahora indica como cantidad 2, como entregada 2, pero como facturado 0.

Para crear la factura, simplemente, tenemos que pulsar sobre el botón **“Crear factura”**. En ese momento, nos pregunta si necesitamos una **“Factura regular”**, un **“Depósito (porcentaje)”** o un **“Depósito (cantidad fija)”**. Seleccionaremos la **“Factura regular”**, puesto que las otras dos opciones se utilizan para **facturar solo una cantidad del pedido**. Se nos crea un borrador de factura y tendremos que publicarla para crear la factura real.

Una vez pulsado el botón **“Publicado”**, nos permite, entre otras cosas, **enviar e imprimir** el pago o **registrar el pago** (que es el siguiente paso). Si pulsamos en **“Enviar e Imprimir”**, se abre una nueva ventana con un e-mail (plantilla) y la factura adjunta en el mismo.

El registro de pagos

Una vez que le enviamos la factura al cliente y este hace el abono, nos llega la notificación de pago y deberemos registrarlo en nuestro flujo de ventas. Tal y como vimos anteriormente, al publicar la factura se nos habilitó el botón de **“Registrar pago”**. Desde ahí podemos registrar el pago, pero en Odoo, normalmente, hay varios puntos desde donde hacer la misma función. En este caso, **también podemos registrar el pago** desde el **módulo de Contabilidad**. Si entramos en **“Clientes”** y en **“Facturas”** nos aparece la factura de Miguel. Al entrar en dicha factura, podemos pulsar el botón de **“Registrar pago”** en cuya ventana ya encontramos todos los campos autoinformados para que, simplemente, le pulsemos sobre **“Validar”** (tras chequear que se corresponden los importes). En ese momento, nos aparece la factura como **pagada**.

Analizar los beneficios

Cada cierto tiempo, es recomendable visualizar los informes sobre las cuentas de la empresa para analizar si las decisiones que estamos tomando son las correctas o no. Puede hacerse en cualquier momento.

En el modulo de contabilidad podremos ver todas las opciones... puede cambiar la forma y estructura de este módulo, por eso mejor que le echemos un vistazo y veamos sus posibilidades.

El **proceso de compra-venta** lo podemos resumir así:

En el módulo de Compras creamos la orden de compra, indicando el proveedor, el producto y la cantidad.

Posteriormente, en el módulo de Inventario, en las recepciones pendientes de validar, confirmamos la recepción del stock. Es en ese momento cuando vemos en la lista de productos que este ya tiene el stock que hemos comprado.

Finalmente, para llevar un correcto control de las cuentas, debemos registrar la factura que el proveedor nos ha enviado, haciendo uso del módulo de Contabilidad.

Hemos hecho uso de los **módulos de Compra, Inventario y Contabilidad**.