

INFORMÁTICA Y COMUNICACIONES
G.S. Desarrollo de Aplicaciones
Multiplataforma



Sistemas de gestión empresarial

Juan José García Lazo



Contenidos digitales
www.sintesis.com

EDITORIAL
SINTESIS



Sistemas de gestión empresarial

Consulte nuestra página web: www.sintesis.com
En ella encontrará el catálogo completo y comentado



Queda prohibida, salvo excepción prevista en la ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra sin contar con autorización de los titulares de la propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (arts. 270 y sigs. Código Penal). El Centro Español de Derechos Reprográficos (www.cedro.org) vela por el respeto de los citados derechos.

Sistemas de gestión empresarial

Juan José García Lazo



ASESOR EDITORIAL:
Juan Carlos Moreno Pérez

© Juan José García Lazo

© EDITORIAL SÍNTESIS, S. A.
Vallehermoso, 34. 28015 Madrid
Teléfono 91 593 20 98
<http://www.sintesis.com>

ISBN: 978-84-1357-077-8
Depósito Legal: M-9.820-2021

Impreso en España - Printed in Spain

Reservados todos los derechos. Está prohibido, bajo las sanciones penales y el resarcimiento civil previstos en las leyes, reproducir, registrar o transmitir esta publicación, íntegra o parcialmente, por cualquier sistema de recuperación y por cualquier medio, sea mecánico, electrónico, magnético, electroóptico, por fotocopia o por cualquier otro, sin la autorización previa por escrito de Editorial Síntesis, S. A.

índice

PRESENTACIÓN	11
1. INTRODUCCIÓN A LOS SISTEMAS DE GESTIÓN EMPRESARIAL	13
Objetivos	13
Mapa conceptual	14
Glosario	14
1.1. Introducción	15
1.2. La gestión empresarial	15
1.2.1. Objetivo de la empresa	16
1.2.2. Procesos de negocio, datos y flujo de trabajo	16
1.3. Sistemas de información de gestión	17
1.3.1. La sopa de letras: MIS, SIG, SIE...	18
1.3.2. Una clasificación de los sistemas de gestión empresarial	19
1.3.3. Los mercados verticales	21
1.4. Un poco de historia	22
1.5. La fidelización de clientes. Concepto de CRM	24
1.5.1. Funcionalidades de un CRM	25
1.5.2. ¿Solo para grandes empresas?	26
1.5.3. Principales fabricantes	27
Supuesto práctico	29
Resumen	29
Actividades de autoevaluación	30
2. ERP (ENTERPRISE RESOURCE PLANNING)	33
Objetivos	33
Mapa conceptual	34

Glosario	34
2.1. Introducción. ¿Qué es un ERP?	35
2.1.1. Ejemplo de un proceso de negocio	36
2.1.2. Origen y evolución del concepto ERP	37
2.2. Características, componentes e infraestructura de un ERP	38
2.2.1. Características de un ERP	38
2.2.2. Componentes de un ERP	39
2.2.3. Detalle de funcionalidades de los principales módulos de un ERP	40
2.2.4. Infraestructura	42
2.3. Soluciones <i>On-premise</i> y <i>Cloud</i>. Licenciamiento	43
2.3.1. Tipos de licencia	44
2.4. Tipos, ventajas e inconvenientes de una solución ERP	45
2.4.1. <i>To do or to buy (or to rent)</i>	45
2.4.2. Importancia de los ERP. La información y el poder	46
2.4.3. Dificultades de implementar un ERP	47
2.5. El mercado actual	49
2.5.1. Soluciones propietarias	50
2.5.2. Soluciones <i>Open Source</i>	51
2.5.3. Tecnologías más utilizadas	51
2.6. La inteligencia de negocio	51
2.6.1. ETL	52
2.6.2. OLAP	52
2.6.3. DataWarehouse	53
2.6.4. DataMining	53
2.6.5. KPI	53
2.6.6. BSC	54
Supuestos prácticos	54
Resumen	55
Actividades de autoevaluación	56
3. INSTALACIÓN DE UN ERP	59
Objetivos	59
Mapa conceptual	60
Glosario	60
3.1. Introducción	61
3.2. Requisitos previos de hardware y software. Escenario práctico	61
3.2.1. Odoo ERP	62
3.3. Modalidades de instalación	64
3.3.1. Instalaciones monopuesto	65
3.3.2. Instalación de solución cloud: Odoo ERP	65
3.3.3. Instalación para Windows con instalador	67
3.3.4. Instalación en Ubuntu desktop mediante paquete .deb	69
3.3.5. Instalación manual en Ubuntu server con fuentes obtenidas de github	71
3.3.6. El problema de las dependencias en las distintas modalidades de instalación de Odoo	77
3.3.7. Estructura general de carpetas de la aplicación	78
3.4. Tipos de módulos	80
3.4.1. Módulos básicos y específicos	81
3.4.2. Instalación de aplicaciones y módulos en Odoo	82
Supuestos prácticos	84

Resumen	86
Actividades de autoevaluación	87
4. CONFIGURACIÓN INICIAL Y FUNCIONAMIENTO DE OODO ERP	89
Objetivos	89
Mapa conceptual	90
Glosario	90
4.1. Introducción	91
4.2. Configuración inicial	91
4.2.1. Datos básicos de la compañía	92
4.2.2. La comunicación interna	92
4.2.3. Contactos e importación de datos	94
4.2.4. Planificando actividades (CRM)	95
4.3. Principales aplicaciones de Odoo	96
4.3.1. Aplicación Ventas	97
4.3.2. Aplicación Facturación	98
4.3.3. Aplicación Compra	98
4.3.4. Aplicación Inventario	99
4.3.5. Aplicación Empleados	101
4.3.6. Aplicación Contabilidad	103
4.3.7. Aplicación Punto de Venta (PoS o TPV)	104
4.4. Etapas del ciclo de ventas: iniciativa, oportunidad y presupuesto	104
4.5. La gestión de las empresas de servicios profesionales	106
4.5.1. Aplicación Proyectos	107
4.5.2. Aplicación Partes de horas	108
Supuestos prácticos	109
Resumen	111
Actividades de autoevaluación	112
5. IMPLANTACIÓN DE UN ERP EN UNA EMPRESA	115
Objetivos	115
Mapa conceptual	116
Glosario	116
5.1. Introducción	117
5.2. Sites de ayuda para elección de un ERP	118
5.3. Fases de la implantación de un ERP	118
5.3.1. Primera fase. Diseño de la solución	119
5.3.2. Segunda fase. Estudio de las necesidades de infraestructura	121
5.3.3. Tercera fase. Desarrollo de la Implantación (como proyecto)	121
5.3.4. Cuarta fase. Puesta en marcha y explotación	122
5.4. Usabilidad	123
5.5. Evolución tras la implementación	124
5.6. Errores y costes ocultos	125
Supuestos prácticos 5.1 y 5.2	126
Resumen	127
Actividades de autoevaluación	128

6. VISIÓN GENERAL DE ODOO ERP	131
Objetivos	131
Mapa conceptual	132
Glosario	132
6.1. Introducción. Conceptos básicos	133
6.2. ORM (<i>Object Relational Mapping</i>)	133
6.3. Modelos	134
6.4. Interfaces de usuario: menú y opciones de menú	137
6.5. Interfaces de usuario: vistas	140
6.5.1. Herencias de vistas	141
6.5.2. Prioridad de las vistas	143
6.5.3. Vistas complejas	143
6.6. Interfaces de usuario	144
6.6.1. Acciones de ventana	144
6.6.2. Tableros	144
6.6.3. Informes. Qweb	146
6.7. Serialización e identificadores externos	148
6.8. Secuencias o contadores	148
6.9. Documentos y plantillas	150
Supuestos prácticos	152
Resumen	152
Actividades de autoevaluación	153
7. DESARROLLO DE COMPONENTES	155
Objetivos	155
Mapa conceptual	156
Glosario	156
7.1. Introducción	157
7.2. Estructura básica de un módulo de Odoo. El comando scaffold	157
7.3. Generación de un módulo	159
7.3.1. Creación de un módulo con el comando scaffold	159
7.3.2. Creación del modelo	162
7.3.3. Diseño de las vistas básicas	163
7.3.4. Acciones de ventana	164
7.3.5. Opciones de menú	165
7.4. Generación de informes	165
7.5. Permisos del módulo	167
7.6. Creación paso a paso de un módulo	168
Supuestos prácticos	176
Resumen	178
Actividades de autoevaluación	179
8. GESTIÓN Y MANTENIMIENTO IT DE UN ERP	181
Objetivos	181
Mapa conceptual	182
Glosario	182
8.1. Administración básica en Odoo	183
8.1.1. Gestión de usuarios. Usuarios, grupos y permisos	183

8.1.2. Plantillas de documentos	188
8.2. Administración avanzada en Odoo	190
8.2.1. Gestión de las bases de datos	190
8.2.2. Correo	195
8.3. Monitorización	197
8.4. Rendimiento	197
8.4.1. Configuración del arranque	199
8.4.2. Diagnóstico	204
Supuestos prácticos	205
Resumen	206
Actividades de autoevaluación	206

RECURSOS DIGITALES



Presentación

Este libro permite desarrollar los contenidos del módulo profesional siguiendo una programación respetuosa con la normativa y a la vez atractiva y actualizada, a fin de transmitir al alumnado la importancia que tiene para su futuro laboral adquirir un nivel de conocimientos adecuado, tanto sobre la operativa como sobre la programación de los módulos utilizados en los sistemas de gestión empresarial, es decir, sobre el desarrollo de aplicaciones informáticas para la gestión empresarial y de negocio.

Un porcentaje significativo de las ofertas de empleo que se publican en estos momentos de bonanza para del sector están relacionadas con el desarrollo, modificación o con la natural actualización de este tipo de software, por lo que, con bastante probabilidad se verán inmersos de una u otra manera en el uso de estas herramientas.

Por ello, por una parte, hay que ver desde el punto de vista de un programador las “tripas” de una solución –por razones obvias *Open Source*–, pero por otra, no menos importante, hay que comprender también de forma práctica la parte operativa: los procesos y procedimientos empresariales básicos relacionados con los distintos departamentos y su gestión mediante lo que se conoce como ERP (*Entreprise Resource Planning*), para entender qué espera el usuario final, que a veces difiere de lo que cree el desarrollador que necesita.

Este texto se reparte, una vez mostrados los conceptos básicos y ubicada su evolución en la historia reciente, entre la instalación de soluciones (quizá más propia de los profesionales de sistemas, pero que un programador también debe conocer), su parametrización y adaptación a un entorno real, la operativa del día a día con ellas y, por último, el desarrollo de sus componentes.

Va dirigido a alumnos y profesores del módulo en sí y a cualquier profesional interesado en aprender las bases de este tipo de aplicaciones.

En definitiva, el texto pretende recoger los contenidos básicos del módulo profesional de Sistemas de Gestión Empresarial correspondiente al título de Desarrollo de Aplicaciones Multiplataforma de la familia profesional de Informática y Comunicaciones desarrollado por

el RD 450/2010 de 16 de abril, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma y se fijan sus enseñanzas mínimas.

Este módulo profesional (Código: 0491) proporciona 6 de los 120 créditos ECTS del título y contiene la formación necesaria para desempeñar la función de implantación y adaptación de sistemas de planificación de recursos empresariales y de gestión de relaciones con clientes, que incluye aspectos como:

- La instalación, configuración y explotación de sistemas ERP-CRM.
- La supervisión en la implantación de sistemas ERP-CRM.
- El mantenimiento y consulta de la información de sistemas ERP-CRM.
- La adaptación de ERP-CRM a las necesidades de la empresa.
- El desarrollo de nuevos componentes para ERP-CRM.

Las competencias profesionales, personales y sociales del título específicamente proporcionadas por este módulo son:

- o)* Participar en la implantación de sistemas ERP-CRM evaluando la utilidad de cada uno de sus módulos.
- p)* Gestionar la información almacenada en sistemas ERP-CRM garantizando su integridad.
- q)* Desarrollar componentes personalizados para un sistema ERP-CRM atendiendo a los requerimientos.

Y los objetivos generales relacionados con este módulo son:

- o)* Reconocer la estructura de los sistemas ERP-CRM, identificando la utilidad de cada uno de sus módulos, para participar en su implantación.
- p)* Realizar consultas, analizando y evaluando su alcance, para gestionar la información almacenada en sistemas ERP-CRM.
- q)* Seleccionar y emplear lenguajes y herramientas, atendiendo a los requerimientos, para desarrollar componentes personalizados en sistemas ERP-CRM.

Los contenidos especificados en la Orden EDU/2000/2010, de 13 de julio, por la que se establece el currículo del ciclo formativo de grado superior correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma son:

- a)* Identificación de sistemas ERP-CRM.
- b)* Instalación y configuración de sistemas ERP-CRM.
- c)* Organización y consulta de la información.
- d)* Implantación de sistemas ERP-CRM en una empresa.
- e)* Desarrollo de componentes.

Además, el título desarrollado por el RD 450/2010 de 16 de abril incluye la cualificación profesional incompleta de Administración y programación en sistemas de planificación de recursos empresariales y de gestión de relaciones con clientes IFC 363_3 (RD 1701/2007, de 14 de diciembre):

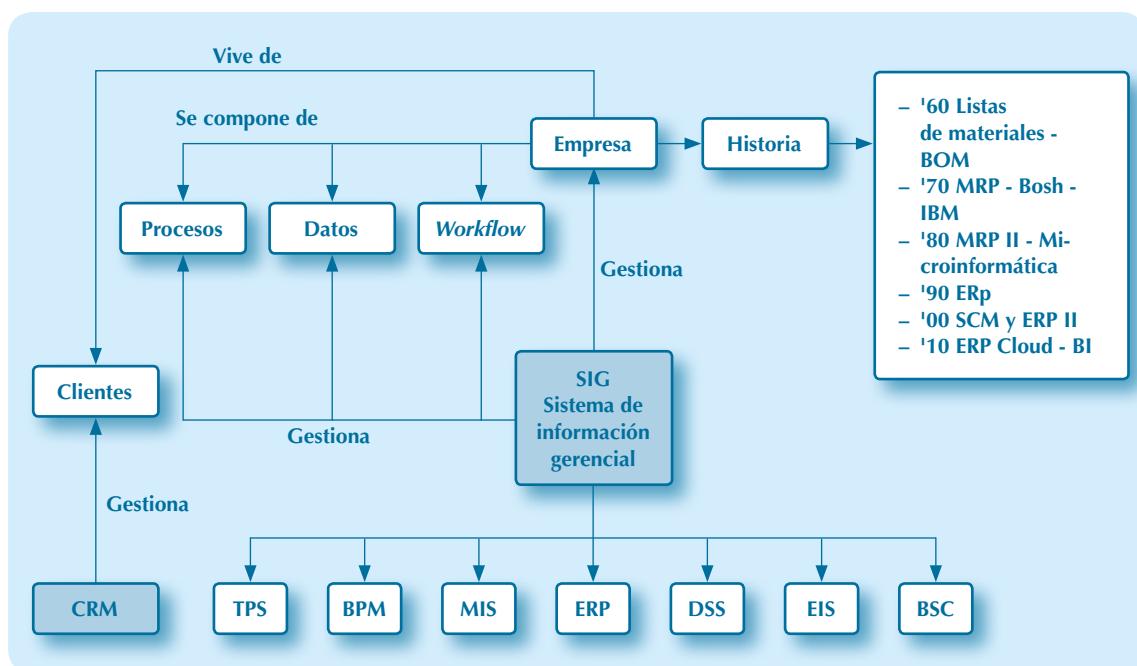
- UC1213_3: Instalar y configurar sistemas de planificación de recursos empresariales y de gestión de relaciones con clientes.

Introducción a los sistemas de gestión empresarial

Objetivos

- ✓ Entender la función de la empresa en la sociedad actual y la problemática asociada a su gestión.
- ✓ Conocer el concepto de sistema de información y las distintas herramientas informáticas que ayudan a la gestión empresarial.
- ✓ Establecer el concepto de MIS (*Management Information System*) [SIG o sistemas de información de gestión o gerencial, en español].
- ✓ Estudiar la evolución que los sistemas de información de gestión han experimentado desde los años sesenta del siglo pasado.
- ✓ Entender la importancia de la gestión de clientes.
- ✓ Reconocer las principales soluciones CRM, tanto comerciales como *Open Source*.

Mapa conceptual



Glosario

Acrónimo. Conjunto de siglas o partes ordenadas de varias palabras que se lee y pronuncia como una palabra.

Cumplimiento (compliance). Procedimientos que garantizan la observancia de la normativa interna, así como de la legislación actual y los códigos éticos, por parte de directivos, empleados y demás actores relacionados con una empresa.

DataMining. Minería de datos. Conjunto de técnicas y tecnologías orientadas a buscar patrones no evidentes, tendencias y reglas que expliquen el comportamiento de los datos en un contexto determinado en grandes bases de datos.

DataWarehouse. Almacén de datos que se caracteriza por contener también los metadatos (datos sobre la procedencia, periodicidad de refresco, fiabilidad, cálculos realizados para su obtención) sobre los propios datos.

Flujo de trabajo (workflow). Es la automatización regulada de los procesos de la empresa para que la información y las tareas circulen entre los distintos departamentos siguiendo un cierto orden.

KPI (Key Performance Indicator). Indicadores clave de desempeño. Indicadores que permiten medir magnitudes de interés.

OLAP (On-Line Analytics Processing). Bases de datos multidimensionales orientadas al procesamiento analítico.

ROI (Return On Investment). Retorno de la inversión. Cálculo del tiempo que se necesita para recuperar lo invertido en un sistema con los beneficios generados por él.

Sistemas de información. Conjunto de datos convertidos en información mediante procesos y mecanismos automatizados e interrelacionados.

Sostenibilidad. Búsqueda del equilibrio entre el crecimiento económico y el cuidado del medio ambiente. Calidad de satisfacer necesidades actuales sin comprometer a generaciones futuras.

Tecnologías de información y comunicación (TIC). Dícese del conjunto de herramientas hardware y software utilizadas para el almacenamiento, tratamiento y transmisión de la información.

Transacción. Todo aquello que modifica o genera datos de un sistema de información.

1.1. Introducción

A lo largo de la vida profesional de un programador, es altamente probable que este participe en proyectos relacionados con los sistemas de información-gestión empresarial, ya sea como desarrollador, como soporte técnico o como usuario.

Por eso, es necesario entender la función de la empresa en la sociedad actual, la problemática asociada a su gestión, así como familiarizarse con los sistemas de información y las distintas herramientas informáticas que ayudan a la gestión empresarial aplicando las tecnologías de la información y las comunicaciones (TIC) para el tratamiento y manipulación de la ingente información que se maneja, así como la evolución que han experimentado desde los años sesenta del siglo pasado.

La vital importancia de conseguir clientes y, sobre todo, de mantenerlos, explica la aparición de los sistemas de gestión de las relaciones con clientes, más conocidos por sus siglas en inglés, CRM (*Customer Relationship Management*) que ayudan en el día a día de la fuerza comercial de cualquier empresa o negocio.

TOMA NOTA



La finalidad de la gestión empresarial es conseguir que la empresa sea viable mediante una correcta planificación y un control de los aspectos productivos, comerciales, financieros, logísticos, etc., del negocio.

1.2. La gestión empresarial

Se entiende por *gestión empresarial* el conjunto de acciones y estrategias que persigue el objetivo de *mejorar el funcionamiento general de una empresa*. Mediante ellas, el personal responsable procura

conseguir el necesario aumento de la productividad, la mejora de la competitividad y el crecimiento de la rentabilidad de la empresa.

1.2.1. Objetivo de la empresa

Con sus recursos humanos, materiales y financieros, la empresa genera los productos y servicios que, debidamente comercializados, producen los *beneficios* que son la razón de su existencia.

Estos beneficios se obtienen mejorando distintos aspectos de la actividad empresarial. Generalmente, se consiguen:

- Maximizando ventas.
- Minimizando costes.
- Eliminando tareas innecesarias.
- Agilizando los procesos cotidianos.
- Automatizando tareas.
- Optimizando recursos.

Y, en definitiva:

- Controlando de manera minuciosa todos los detalles de la empresa.

De hecho, la mejor forma de incrementar los beneficios de una empresa es mediante una acción combinada sobre todos estos aspectos anteriormente relacionados.

En la actualidad, las empresas no solo deben ser *competitivas*, sino que además deben preocuparse por ser cada vez más *sostenibles* y conseguir un mayor nivel de cumplimiento normativo (*compliance*).

Por otra parte, es sobradamente sabido que la *eficiencia* y la *efectividad* influyen en el diferencial de beneficio.

Por todo ello, en la práctica, hacer crecer los beneficios se convierte en una tarea ardua que entraña dificultades que dependen, como se ha podido comprobar, de distintos factores.

Actividad propuesta 1.1



- a) Reflexiona sobre las diferencias entre empresa pública y privada.
- b) ¿Deben las empresas públicas obtener beneficios? Razona la respuesta.
- c) Estudia la diferencia entre objetivos y propósito.

1.2.2. Procesos de negocio, datos y flujo de trabajo

La actividad de una empresa está basada en los llamados *procesos de negocio*, conjunto de tareas relacionadas y ordenadas que proporcionan un producto o servicio, ya sea interno (para otro departamento de la propia empresa) o externo (para el cliente final).

A menudo, los procesos son *secuenciales*, de tal manera que la salida obtenida de un proceso es el inicio para otro proceso.

Los procesos empresariales pueden descomponerse en otros procesos de menor entidad hasta llegar al nivel que se considere elemental, momento en el que reciben el nombre de *transacciones*.

Por otra parte, los *datos* que se manejan en el desempeño diario de la actividad principal de una empresa son de un volumen considerable.

Desde los que son proporcionados por las distintas máquinas empleadas en los procesos de fabricación, si los hubiere, los detalles de las transacciones que se realizan en el día a día, los datos de contacto con el exterior, el histórico de las relaciones con proveedores y clientes, hasta la relación de recursos materiales y humanos internos, el control del almacén, los datos económicos, la publicidad y el marketing, la presencia en la web y en las redes sociales, etc., las empresas van acumulando a lo largo de su existencia datos obtenidos por los distintos departamentos que la componen, que se pueden y deben convertir en información vital.

El manejo de estos datos, y, sobre todo, su tratamiento para la extracción de información relevante, así como la relación entre los distintos departamentos y el intercambio de informaciones entre ellos de forma ordenada y eficiente (flujo del trabajo) son aspectos primordiales para mejorar el funcionamiento de la empresa e incrementar así el beneficio.

1.3. Sistemas de información de gestión

Gran variedad de sistemas de información, como la mensajería, los sistemas bancarios, la venta en línea, las bibliotecas de recursos académicos, los sistemas de concertación de citas, etc., son utilizados a diario de forma casi inconsciente por un considerable y creciente número de usuarios cada vez más inmersos en el mundo digital.

Como en tantos otros ámbitos, la utilización de las TIC y, en particular, el desarrollo de soluciones tecnológicas para la revisión de los procesos, para el control del flujo de trabajo (*workflow*) y, sobre todo, para el tratamiento centralizado de todos esos datos generados por los distintos departamentos, ha supuesto un paso fundamental en la gestión de negocios: ha significado la aparición de los *sistemas de información de gestión empresarial*, conjunto de herramientas muy útiles sin las que hoy día sería imposible sobrevivir como negocio.

Se trata de sistemas de información orientados a resolver problemas empresariales.

Suponen la aplicación de soluciones basadas en las TIC a los requerimientos específicos de los negocios, es decir, están enfocados al negocio y por tanto se dicen empresariales o gerenciales y sustentan el gobierno de las organizaciones y empresas.

TEN EN CUENTA

- ✓ Un sistema de información no es solo el conjunto de recursos tecnológicos que lo soporta, sino también la organización de esos recursos y los métodos de obtención de la información necesaria para el correcto funcionamiento del sistema.

En particular, se puede definir *sistema de información empresarial (SIE)* o *sistema de información gerencial (SIG)* como el conjunto de aplicaciones que acometen las necesidades de tratamiento

simultáneo de la información necesaria para el funcionamiento de la empresa por parte de un grupo de usuarios, y proporcionan así una gestión automatizada del negocio en sentido amplio.

No solo se trata de aprovechar las aplicaciones de escritorio y de productividad personal como procesadores de texto, hojas de cálculo y bases de datos más o menos compartidas, sino desarrollar plataformas de uso simultáneo y común que incluyan los módulos específicos que sostienen todos los aspectos de las necesidades de administración de una empresa.

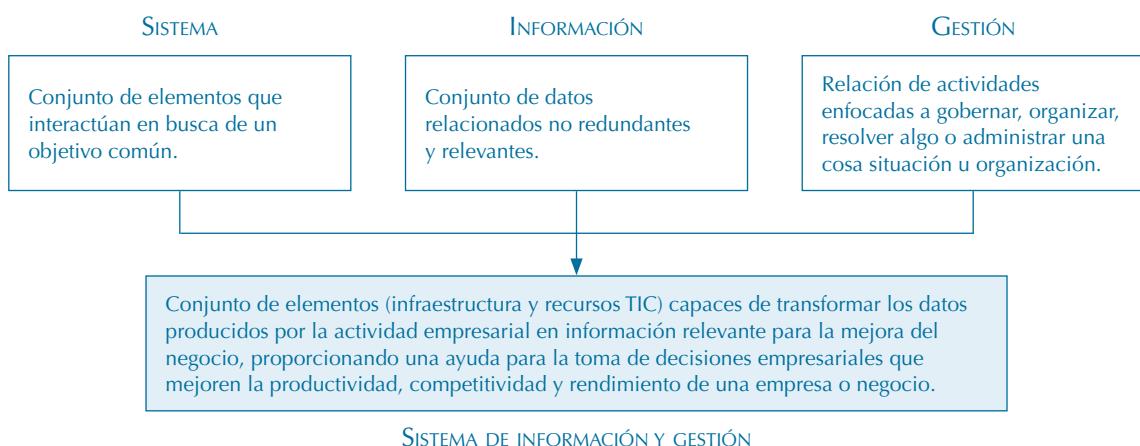


Figura 1.1

Definición de SIG como composición de conceptos.

Actividad propuesta 1.2



Existen también sistemas de información no gerenciales. Busca al menos tres y explica sus procesos, los datos que almacenan y estudia el flujo de información que se produce.

1.3.1. La sopa de letras: MIS, SIG, SIE...

El sector tecnológico utiliza su propia jerga e, intentando simplificar, abusa con frecuencia de la utilización de abreviaturas y acrónimos.

Llegados a este punto, y ante el planteamiento de intentar entender el entorno, es fácil encontrarse con una primera dificultad provocada por la abrumadora existencia de acrónimos (normalmente en inglés con su consiguiente adaptación o traducción directa al español) de conceptos que tienen que ver con alguna fase de los procesos empresariales.

Por eso, es recomendable entender bien lo que el autor consultado expresa con esas siglas que utilizará a lo largo de su escrito. A veces se utilizan las siglas *MIS* (*Management Information System*) para referirse, en general, a estos sistemas de información relacionados con la administración de empresas.

Académicamente, se puede convenir que engloban a las distintas herramientas informáticas enfocadas a proporcionar, mediante el tratamiento de datos generados en una actividad empresarial, todo tipo de procesos automatizados, informes, controles, sistemas de aviso, análisis de resultados, etc., de los distintos estamentos de una empresa.

En la práctica, las herramientas comercializadas no se corresponden estrictamente con las definiciones y ocurre que, a veces, los departamentos de marketing de los desarrolladores de software empresarial las utilizan en sentido laxo.

MIS (Management Information System) son sistemas de información que se alimentan de los datos proporcionados por los distintos procesos propios de la actividad de la empresa, los procesan mediante diversos tratamientos y elaboran información “cocinada” como estadísticas, informes, gráficos, simulaciones, tendencias, etc., que reflejan el funcionamiento corporativo de la organización.

TOMA NOTA



ACRÓNIMOS. En la actualidad es posible encontrar distintos tipos de sistemas de información aplicados a la gestión empresarial en alguno de sus procedimientos, tales como:

- TPS: *Transaction Processing System*. Sistema de procesamiento de transacciones.
- OAS: *Office Automation System*. Sistema de automatización de oficinas.
- MRP: *Material Requirements Planning*. Planificación de los requisitos de material.
- MRPII: *Manufacture Resource Planning*. Planificación de los recursos de fabricación.
- PLM: *Product Lifecycle Management*. Gestión del ciclo de vida de productos.
- SCM: *Supply Chain Management*. Gestión de la cadena de suministro.
- SRM: *Supplier Relationship Management*. Gestión de la relación con proveedores.
- MIS: *Management Information System*. Sistema de información de gestión (SIG).
- BPM: *Business Process Management*. Administración de procesos de negocio.
- ERP: *Enterprise Resource Planning*. Planificación de recursos empresariales.
- CRM: *Customer Relationship Management*. Gestión de la relación con los clientes.
- POS: *Point Of Sale*. Terminal punto de venta (TPV).
- CMS: *Content Management System*. Sistema de gestión de contenidos.
- DMS: *Document Management System*. Sistema de gestión documental.
- KMS: *Knowledge Management System*. Sistema de gestión del conocimiento.
- BI: *Business Intelligence*. Inteligencia de negocio.
- DSS: *Decision Support System*. Sistema de apoyo a la toma de decisiones.
- EIS: *Executive Information System*. Sistema de información ejecutiva.
- BSC: *Balanced Score Card*. Cuadro de mando integral.
- GIS: *Geographical Information System*. Sistema de información geográfica.
- Blog: *Weblog*. Bitácora.

En castellano, hablamos de sistemas de información gerencial (SIG) o incluso sistemas de inteligencia empresarial (SIE). Quizá, los matices que las diferencian serían objeto de un estudio más avanzado.

1.3.2. Una clasificación de los sistemas de gestión empresarial

Es fácil encontrar diferentes clasificaciones de los sistemas de información empresarial. Para verlo en perspectiva respecto a las áreas que cubren, se podría considerar la siguiente clasificación general de los sistemas de información empresarial más destacados, empezando desde su nivel inferior:

A) Sistemas de procesamiento de transacciones (TPS)

Son el nivel operacional más bajo. Soportan la rutina diaria del negocio (transacciones comerciales, económicas, existencias, etc.), pero son muy importantes desde el punto de vista de la consistencia y la integridad de estas. Constituyen la base que utilizarán los sistemas de capa superior.

B) Sistemas de gestión por procesos de negocio (BPM)

Gestionan los *procesos* de la organización, es decir, las acciones que deben realizar personas y máquinas de forma ordenada para lograr un determinado objetivo y, en particular, los procesos físicos de producción, generalmente mediante la monitorización de sensores electrónicos.

C) Sistemas de información de gestión (MIS)

Como ya se ha comentado, recogen información de diferentes fuentes internas y la procesan para proporcionar informes, estadísticas y proyecciones a futuro que ayuden a la gerencia a tener una visión fidedigna de la situación de una parte de la empresa.

D) Sistemas de colaboración empresarial (ERP)

Es un MIS integrado. Son sistemas *integrales* de información que, en una única base de datos, recogen, procesan y analizan datos de todos los estamentos de la organización empresarial proporcionando así información relevante sobre procesos de producción, ventas, logística, recursos humanos, gestión de proyectos, contabilidad y finanzas de la empresa, con el fin de mejorar la gestión empresarial.

Son utilizados por una amplia gama de usuarios que tienen acceso autorizado a la información en función del trabajo que desempeñan.

E) Sistemas de apoyo a la toma de decisiones (DSS)

Son sistemas interactivos basados en la combinación de los datos para su análisis que proporcionan un paso más: información organizacional y funciones de *simulación y modelado* que podrá utilizar el responsable para seleccionar la mejor opción entre varios escenarios.

Es un tipo especial de inteligencia de negocio (BI) que almacena información interna y mediante OLAP, DataWarehouse y DataMining facilita la toma de decisiones (por ejemplo: gestión de tráfico, aprobación de créditos, diagnósticos médicos, etc.) mediante simulaciones, análisis estadísticos y estudios de tendencia.

F) Sistemas de información ejecutiva (EIS)

Proporcionan, generalmente en formato gráfico e intuitivo, acceso con distintos niveles de detalle a la información interna y también a la proporcionada por *fuentes externas*. Ayudan en

la *toma de decisiones estratégicas* que afectan a toda la organización mediante la elaboración de informes, consultas y listados de las distintas áreas de la empresa de forma consolidada.

G) Cuadro de mando integral (BSC)

El BSC, *Balanced Scorecard* o *Dashboard* (cuadro de mando) es la herramienta de control que monitoriza el grado de consecución de los objetivos de los distintos departamentos o áreas de negocio de una empresa.

Lo hace mediante el establecimiento de *indicadores*, conocidos como KPI (*Key Performance Indicators*), organizados normalmente en cuatro tipos: financiero, conocimiento del cliente, procesos internos y de aprendizaje y crecimiento. Están más orientados a la monitorización de indicadores que al análisis meticoloso de la información.



Se podrían considerar otros tales como los sistemas expertos basados en la inteligencia artificial (IA) o los sistemas de ayuda a decisiones de grupo (GDSS).

1.3.3. Los mercados verticales

Como no podía ser menos, en un campo tan heterogéneo como la actividad empresarial, existen también las llamadas *soluciones verticales*, es decir, soluciones específicamente desarrolladas para un tipo de negocio o mercado en particular. A diferencia del software horizontal, generalista, y sin llegar a ser un software a medida, suelen constituir la solución para sectores específicos como pueden ser el agrícola, el inmobiliario, el industrial, los despachos profesionales, etc.

Son productos normalmente desarrollados a partir de la experiencia en el sector y en ocasiones participan asociaciones sectoriales en su elaboración por lo que incluyen las mejores prácticas y los indicadores propios del negocio.



Figura 1.2
Página web
de un proveedor
de soluciones verticales
Ceesa.

Actividad propuesta 1.3



Estudia en qué casos un software vertical puede ser la mejor solución de gestión para una empresa.

1.4. Un poco de historia

A menudo, al comparar estudios históricos, se pueden advertir diferencias de décadas según el autor y las fuentes utilizadas. Eso es debido, normalmente, al desfase que solía y suele producirse entre la expansión de los desarrollos tecnológicos en EEUU y su aparición y despliegue en Europa y, en particular, en España.

A mediados del siglo pasado, las empresas utilizaban procesos de gestión manuales y poco automatizados.

Heredados de los sistemas militares, en la década de los sesenta se establecieron los conceptos de gestión automatizada y se empezaron a utilizar las herramientas de planificación en el ámbito comercial. Aplicaciones básicas como los Gestores de Listas de Materiales, BOM (*Bill of Materials*) o IMC (*Inventory Management Control*) fueron adaptadas desde el mundo militar al empresarial.

En esta época se fundan numerosas empresas dedicadas al desarrollo de software que en una versión básica se incluía con la venta del hardware. Los distintos departamentos empezaron a utilizar un software específico para desarrollar sus funciones dentro de la empresa.

Además de la ya mencionada, las primeras áreas funcionales del negocio informatizadas fueron la de contabilidad, la financiera y la de almacén.

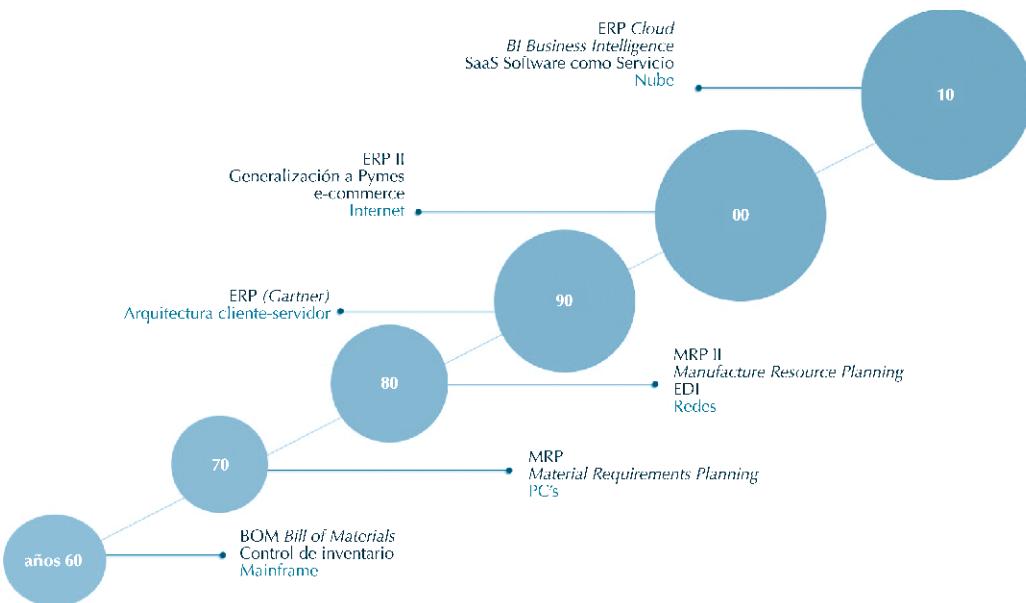


Figura 1.3

Evolución temporal de los sistemas de información empresarial.

A finales de esta década y principios de los setenta, aparecieron los primeros sistemas de planificación de la producción o planificación de las necesidades materiales (MRP) (*Material*

Requirement Planning), software que se ejecutaba en mainframes, sistemas propietarios tipo IBM S36, u ordenadores “mini” tipo DEC (*Digital Equipment Corporation*) con sistema operativo VAX/VMS..., naciendo un mercado dominado por IBM. El primero de los MRP se atribuye a la empresa alemana Bosch unos años antes.

Los MRP, basados en el BOM, utilizaban los ordenadores para resolver el mayor problema con el que se encontraban las empresas con procesos de fabricación en etapas: calcular el material que se necesita, compararlo con lo que se tenía en el almacén y obtener así cuándo se debía abastecer.

Posteriormente, en los años ochenta, la revolución de la microinformática con la aparición de las primeras computadoras personales (IBM PC), provocó que más empresas de menor tamaño empezaran a utilizarlos para administrar su negocio.

Los MRP evolucionaron a los MRP II, siglas que ahora se correspondían con *Manufacturing Resource Planning*, y que incluían también la gestión de los costes de la materia prima, de la mano de obra, los logísticos, etc., es decir, integraron los MRP con componentes financieros.

A partir entonces se empezaron a utilizar sistemas informáticos para mecanizar las tareas de algunos departamentos de forma independiente. Así, se instalaba un sistema de contabilidad, a veces un programa de facturación que complementaba al contable; después se añadía un programa de nóminas, casi siempre alguno específico aplicado al proceso de fabricación o de la prestación de servicios realizada, hasta que se empezó a buscar la integración de los distintos sistemas y a constatar la necesidad de conectar los PC, apareciendo la estructura cliente-servidor.

Los fabricantes de software empezaron entonces a proporcionar aplicaciones que permitían a varios usuarios acceder a los datos de forma simultánea. Así aparecieron los ERP en la década de los noventa, añadiendo la gestión de la fabricación, la gestión de las relaciones con proveedores y clientes, la gestión de los recursos y, de forma incipiente, la inteligencia de negocio. La idea base consistió en utilizar una *fuente de información centralizada* común a todos. De esa manera, los datos se introducían una sola vez en el sistema y estaban disponibles para el resto de los integrantes de la organización: una única base de datos y la posibilidad de extraer, de forma regulada, la información contenida en función de las necesidades y privilegios de cada usuario.

Fue durante esta era que los primeros sistemas de planificación de recursos empresariales, *Enterprise Resource Planning* (ERP) se desarrollaron y ejecutaron en arquitecturas cliente-servidor. Un sistema ERP es, en definitiva, una aplicación de software estructurada en módulos que se complementan y que utilizan una base de datos centralizada, y que se puede utilizar para gestionar todo el negocio de una empresa.

Se trataba de organizar el trabajo mediante una planificación previa de las necesidades de recursos, el control del consumo de los mismos y la gestión por procesos. Durante mucho tiempo y enfocada a un mercado de grandes empresas fue un mercado liderado por la alemana SAP.



SABÍAS QUE...

Los 2000 son la década de la expansión del comercio electrónico (*e-commerce*), es decir, la comercialización de productos y servicios utilizando únicamente las TIC en un espacio digital sin un contacto físico, ya sea entre empresas (B2B-*Business to Business*) o al cliente final (B2C-*Business to Customer*).

La evolución empresarial en la década de los 2000 hacia la externalización de las operaciones en las que la empresa no está especializada provoca la necesidad de la coordinación con el

exterior apareciendo el concepto de Sistemas de Gestión de la Cadena de Suministro, o SCM (*Supply Chain Management*), es decir la gestión del intercambio de información electrónica (*EDI-Electronic Data Interchange*) entre los sistemas de gestión, generalmente distintos, de las empresas y sus proveedores, y en consecuencia, la de ERP II, que añade estas funcionalidades al ERP clásico.

Aparecen conceptos como CRM (*Customer Relationship Management*) para administrar las relaciones con los clientes y PLM (*Product Lifecycle Management*) que proporciona la gestión de la información técnica del producto fabricado a lo largo de todo su ciclo de vida.

Por otra parte, la explosión de las TIC y su adopción cada vez más generalizada por parte de la pequeñas empresas, universaliza la utilización de los ERP y aparecen soluciones adaptadas a cualquier tamaño de organización.

A partir de la década de 2010, la aparición de conceptos como el *Cloud Computing* y el *SaaS* permite una explosión de crecimiento, pues basta un dispositivo y un navegador para disponer de todas las funcionalidades relativas a la gestión empresarial que se desee, independientemente del tamaño de la empresa.

Actualmente, y englobados bajo el nombre de ERP de nuevo, todos estos sistemas han pasado de tener una función meramente operativa a jugar un rol estratégico como sistemas de ayuda a la gerencia de las empresas y soporte fundamental para la toma de decisiones tanto operativas como estratégicas, dando paso a la aparición de los conceptos de BI, DSS y EIS.

Actividad propuesta 1.4



Averigua qué significado tienen las siglas JIT. Estudia dónde se creó y las consecuencias de su utilización en las cadenas de producción.

1.5. La fidelización de clientes. Concepto de CRM

Conseguir clientes, y lo que a la postre se convierte en más importante, mantenerlos, es un objetivo fundamental de la gestión empresarial.

Fidelizar es conseguir una venta recurrente en el tiempo, es decir, aplicar estrategias para que el cliente se convierta en habitual. La importancia de la fidelización estriba en que mantener un cliente a lo largo de su vida útil es siempre mucho menos costoso que conseguir nuevos.

Pero los clientes se han vuelto más exigentes y la oferta se ha globalizado, por lo que es importante conocer sus necesidades y aspirar a satisfacerlos incluso antes de que el cliente sepa de su necesidad.

Por eso, las empresas han evolucionado hacia un planteamiento basado en el cliente y no en el producto, enfoque habitual del siglo pasado.

La visión del comercial con su cartera de clientes y libretas de pedido y, todo lo más, una hoja de cálculo en el caso más informatizado, ha dado paso a otra visión más moderna, englobada en la denominada automatización de la fuerza de venta, basada en la utilización de sistemas apoyados en la movilidad y el acceso desde cualquier dispositivo que gestionan todo tipo de interacciones con el cliente con el objetivo de mejorar las relaciones comerciales con él.

Un sistema de gestión de las relaciones con el cliente, CRM por sus siglas en inglés (*Customer Relationship Management*), es un software colaborativo basado en la orientación al cliente que registra toda su información de contacto, pero también, y lo que es más importante, almacena

las transacciones de todo tipo mantenidas con él de tal manera que proporcionan una visión global de un ecosistema al que pertenecen los productos, los servicios, los clientes actuales y potenciales y los recursos de la empresa.

Engloban y centralizan las bases de datos de las interacciones que se tienen con los clientes. Esta centralización de información, accesible en mayor o menor medida por el personal de la empresa en función de sus privilegios, proporciona un conocimiento profundo del cliente y permite establecer fácilmente y de forma personalizada las estrategias comerciales, de marketing y de servicio al cliente que proporcionarán en definitiva su fidelización.

Puede decirse que son la evolución inteligente de las hojas de cálculo que gerentes, comerciales y muchos profesionales en general suelen utilizar al comenzar su actividad.

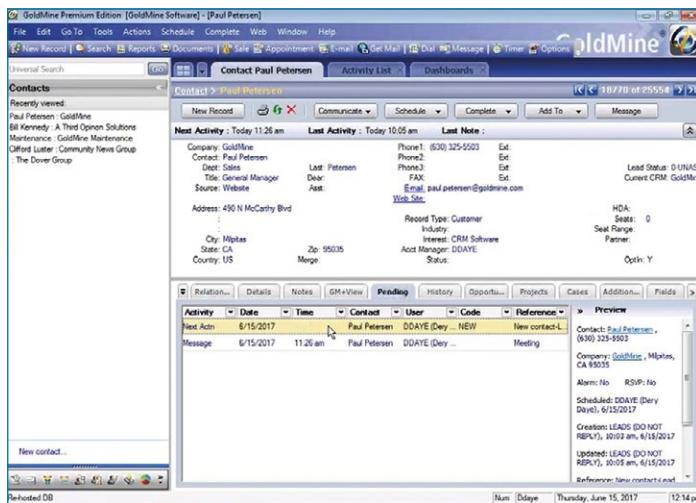


Figura 1.4

Pantalla de GoldMine, uno de los primeros CRM.

1.5.1. Funcionalidades de un CRM

Un CRM es una herramienta de implementación, normalmente sencilla, dirigida en especial a los departamentos comerciales y de marketing de las empresas, aunque no únicamente a ellos, que maneja la gestión de datos de clientes, las oportunidades de venta, los presupuestos, los ingresos por ventas y las campañas publicitarias y de marketing.

Las principales funcionalidades, sin pretender ser exhaustivos, de un CRM son las siguientes:

- Recopila y organiza toda la información de contacto de los clientes.
- Permite múltiples clasificaciones de estos (actuales y potenciales y dentro de cada categoría clasificaciones por cualquier aspecto: localización, tamaño, sector, etc.).
- Gestiona oportunidades de venta desde el inicio.
- Dispone de plantillas personalizables de mail y casi cualquier tipo de documento relacionado con el proceso de venta, como presupuestos, órdenes de compra).
- Realiza el seguimiento de las operaciones y del personal de ventas involucrado.
- Dispone de calendario y sistema de avisos para los usuarios.
- Calcula previsiones, obtiene estadísticas y elabora informes.
- Permite personalizar el trato en el servicio de atención al cliente.
- Automatiza el marketing.

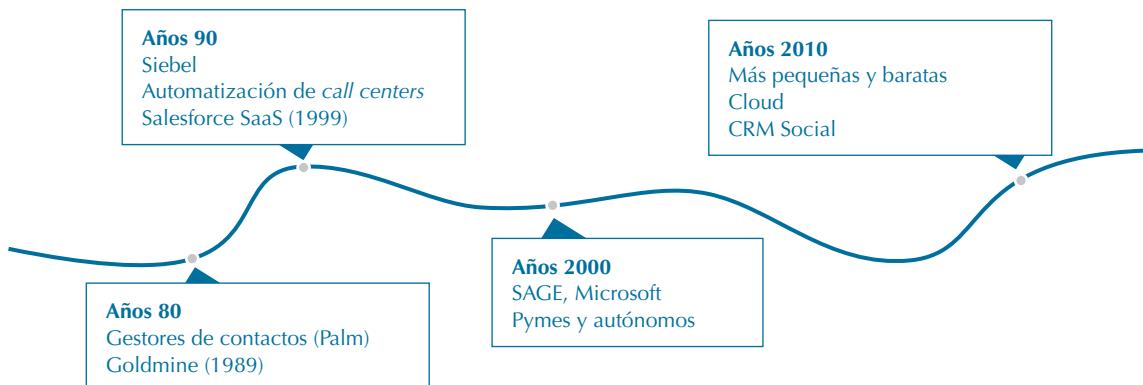


Figura 1.5
Evolución del software de CRM.

Un CRM es una herramienta de seguimiento, de comunicación y de análisis.

Un CRM permite hacer un seguimiento automatizado de las oportunidades de venta, desde la fase de obtención de datos, proporcionando normalmente:

- Sistemas de captura automatizada desde ficheros de estructura simple como “.csv”.
- Campos especiales como fechas de contacto o de marcado de hitos.
- Campos para notas amplias.
- Histórico de operaciones y comunicaciones.
- Recordatorios y avisos.
- Archivos adjuntados en las comunicaciones.

Usándolo como herramienta básica de comunicación, permite controlar las “conversaciones” en el tiempo, disponer de plantillas de respuesta o primer contacto tipo, utilizarla en muchos casos como plataforma de e-mailing.

Y, por supuesto, proporciona información analítica mediante estadísticas e informes por cliente, por tipo de propuesta, por estimación de ventas, etc.

RECUERDA

- ✓ Estudiar la tasa de retención de clientes, la cantidad de nuevos clientes referenciados por clientes actuales, la evolución del gasto de cada cliente, y el factor de reiteración de las compras, entre otras cosas, permitirá ponderar los beneficios que se están obteniendo después de implementar un CRM, lo que proporcionará el ROI de su implementación.

1.5.2. ¿Solo para grandes empresas?

Independientemente de su tamaño y del número de clientes que tenga una empresa, esta debe preocuparse de gestionar sus relaciones con ellos de la mejor forma.

Los gestores actuales ya no tienen esa idea preconcebida hasta hace relativamente poco tiempo de que los CRM son herramientas complicadas y costosas y que, por tanto, solo las grandes empresas les podrían sacar partido.

Existe en el mercado actual una gran variedad de sistemas CRM para todo tipo de empresas. Un CRM no solo proporciona valor añadido a organizaciones grandes, sino que las pymes y los autónomos también pueden beneficiarse de la automatización de la gestión con sus clientes.

Hay una solución para cada negocio y lo importante es saber calcular el retorno de la inversión (ROI) para encontrar la apropiada al tamaño y a la actividad de la empresa.

Existen incluso soluciones monousuario.

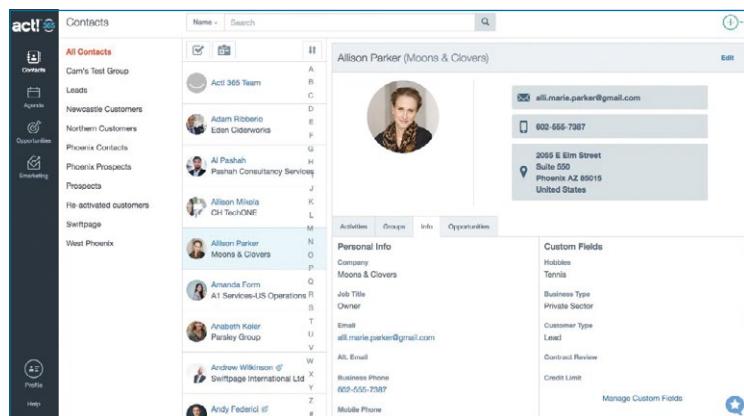


Figura 1.6
Captura de pantalla de ACT!,
otro CRM con historia.

1.5.3. Principales fabricantes

La diferenciación entre soluciones Open Source, semigratuitas y comerciales, se ha ido reduciendo en los últimos años. Varias de las aplicaciones que en sus inicios tuvieron licencia Open Source han evolucionado hacia una oferta mixta con versiones sin coste hasta un cierto número de usuarios o que incluyen un número limitado de funcionalidades o, algunas como OpenBravo, directamente, se han convertido en versiones totalmente comerciales.

Las grandes compañías fabricantes de software empresarial como Microsoft, Oracle, Salesforce o SAP proporcionan soluciones CRM de forma independiente o como parte integrante de sistemas ERP, sobre los que se trabajará en el próximo capítulo.



Figura 1.7
Cuadrante Mágico CRM.
Fuente: SoftwareReviews.

De entre las que siguen siendo Open Source, aunque la comunidad que las desarrolló puede haber abandonado el mantenimiento y encontrarse en la actualidad bajo el control de empresas que además proporcionan servicios con coste, se pueden mencionar las siguientes: Suite CRM, vTiger CRM, OroCRM y Zurmo.

En la figura 1.7 se muestra un cuadrante de aplicaciones CRM de implantación en el mercado europeo y español. No obstante, un estudio semejante se realizará en el próximo capítulo cuando se estudien los ERP pues sus resultados son totalmente aplicables a los CRM.

WWW

Recurso web

Información sobre software empresarial:

- SoftwareReviews: <https://www.softwarereviews.com>
- Capterra: <http://www.capterra.com>
- Technology Evaluation Centers (TEC): <https://www3.technologyevaluation.com>

La variedad de soluciones es considerable. Sería imposible relacionar los innumerables fabricantes que ofrecen soluciones CRM.

Capterra es una compañía parte de Gartner que, situada en una posición intermedia entre compradores y proveedores, proporciona una de las mayores fuentes de información y de opiniones verificadas sobre software.



Figura 1.8
Resumen informe
Capterra.

Según un informe elaborado por esta empresa en 2018, que recoge, entre otros, datos relativos a clientes y usuarios, las diez marcas más populares son las de la figura 1.8.



Actividad propuesta 1.5

De los CRM citados, escoge tres y realiza una comparativa entre ellos.

Supuesto práctico

1. Para practicar el uso de una solución cloud de CRM, abre una cuenta en la página web de Zoho CRM (www.zoho.com) y configura una instancia de la aplicación, para lo que necesitarás proporcionar una cuenta de correo. Haz un recorrido por la aplicación y consulta la ayuda que proporciona el fabricante. A continuación:
 - a) Elabora un informe de usuario básico de no más de ocho páginas sobre este CRM de uso gratuito y explica las condiciones.
 - b) Haz la importación de los datos contenidos en la hoja de cálculo *listado clientes* disponible en los recursos digitales. Quizá haya que manipularla para adaptarla al formato que necesita Zoho.
 - c) Configuración: realiza la configuración básica, introduciendo los datos completos de una empresa ficticia.
 - d) Configura el correo electrónico de la aplicación y finaliza el proceso de confirmación.
 - e) Crea tres tareas diferentes en tres clientes distintos.
 - f) Crea una plantilla con el *layout* de dos columnas con una imagen. Personaliza la plantilla con, por ejemplo, una oferta comercial de un producto.
 - g) Envíaselo a un cliente (haz el envío a una cuenta de un compañero para comprobar que está bien configurado y ver lo que recibe el supuesto cliente).

Resumen

- Un programador debe conocer conceptos básicos de la actividad empresarial que le permita participar en el desarrollo de sistemas de gestión empresarial, una de las actividades más demandadas en el mercado laboral actual.
- Las empresas deben ser competitivas, sostenibles y con un alto nivel de cumplimiento.
- Los beneficios se consiguen maximizando ventas, minimizando costes, eliminando tareas innecesarias, agilizando los procesos cotidianos, automatizando tareas, optimizando recursos y, en definitiva, controlando con detalle todos los aspectos de la empresa.

- Los tres pilares sobre los que se construye una gestión eficiente de la empresa son los procesos de negocio, los datos y el flujo de la información.
- Las TIC han proporcionado herramientas cada vez más sofisticadas para la gestión empresarial.
- Un sistema de información empresarial (SIE) o sistema de información gerencial (SIG), es un conjunto de aplicaciones que proporcionan una gestión automatizada del negocio en sentido amplio.
- Los sistemas de gestión empresarial, en sus numerosas variedades, han contribuido al éxito de los negocios.
- Existe infinidad de acrónimos relacionados con herramientas de gestión empresarial.
- Aunque a veces se solapan algunos, se puede categorizar los tipos de sistemas de información según que actúen sobre determinados procesos de la actividad empresarial, en: sistemas de procesamiento de transacciones (TPS), sistemas de gestión por procesos de negocio (BPM), sistemas de información de gestión (MIS), sistemas de colaboración empresarial (ERP), sistemas de apoyo a la toma de decisiones (DSS), sistemas de información ejecutiva (EIS) y cuadro de mando integral (BSC).
- La evolución experimentada desde los años sesenta hasta la actualidad ha llevado a estos sistemas a jugar un rol estratégico en la gestión de empresas.
- Se llaman "soluciones verticales" a las específicamente desarrolladas para un tipo de negocio o mercado en particular.
- La vital importancia de conseguir clientes y, sobre todo, de mantenerlos, explica la aparición de los sistemas de gestión de las relaciones con clientes (CRM).
- Un CRM es una herramienta de seguimiento, de comunicación y de análisis.
- Hay una solución para cada negocio, independientemente del tamaño y su actividad. Solo hay que encontrar la que asegura el retorno de la inversión (ROI) apropiado.
- Con un CRM se vende más y mejor.

ACTIVIDADES DE AUTOEVALUACIÓN

1. Los beneficios en una empresa se mejoran:

- a) Aumentando ventas.
- b) Ajustando los costes.
- c) Automatizando tareas.
- d) Todas las opciones son correctas.

2. Los procesos de negocio son:

- a) Actividades de mejora del negocio.
- b) Conjunto de tareas que proporcionan un producto o servicio.
- c) Conjunto de tareas que proporcionan un producto o servicio externo.
- d) Conjunto de transacciones comerciales.

3. MRP es el acrónimo inglés de:

- a) *Material Requirements Planning*.
- b) *Manufacture Resource Planning*.
- c) *Management Resource Planning*.
- d) *Multi Relationship Process*.

4. Los indicadores KPI los utiliza:

- a) El sistema de procesamiento de transacciones (TPS).
- b) Los CRM para clasificar clientes.
- c) El cuadro de mando integral.
- d) Ninguna de las respuestas es correcta.

5. Un CRM es una herramienta:

- a) De seguimiento y análisis.
- b) De comunicación.
- c) De seguimiento, comunicación y análisis.
- d) Ninguna de las respuestas es correcta.

6. Un SIG (sistema de información gerencial) gestiona:

- a) Datos y procesos.
- b) Datos, procesos y el flujo de trabajo o *workflow*.
- c) Clientes y propietarios.
- d) Procesos.

7. Señala las marcas y/o fabricantes de software ERP no comercial:

- a) Microsoft.
- b) SAGE.
- c) Dolibarr.
- d) OpenBravo.

8. La gestión del ciclo de vida del producto la hace un módulo denominado:

- a) MIS.
- b) MRP.
- c) PLM.
- d) MRP II.

9. Los sistemas de gestión de procesos de negocio (BPM) son:

- a) Sistemas de información gerencial.
- b) Sistemas ERP.
- c) Es un concepto de *business intelligence*.
- d) También se llaman sistemas de procesamiento de transacciones.

10. Una clasificación de los MIS es la que los encuadra en:

- a) DSS (*Decision Support System*) y ERP (*Enterprise Resource Planning*).
- b) CRM y ERP.
- c) DSS (*Decision Support System*) y *Experts Systems*.
- d) CMS (*Content Management System*) y blog.

SOLUCIONES:

1. **a** **b** **c** **d**

2. **a** **b** **c** **d**

3. **a** **b** **c** **d**

4. **a** **b** **c** **d**

5. **a** **b** **c** **d**

6. **a** **b** **c** **d**

7. **a** **b** **c** **d**

8. **a** **b** **c** **d**

9. **a** **b** **c** **d**

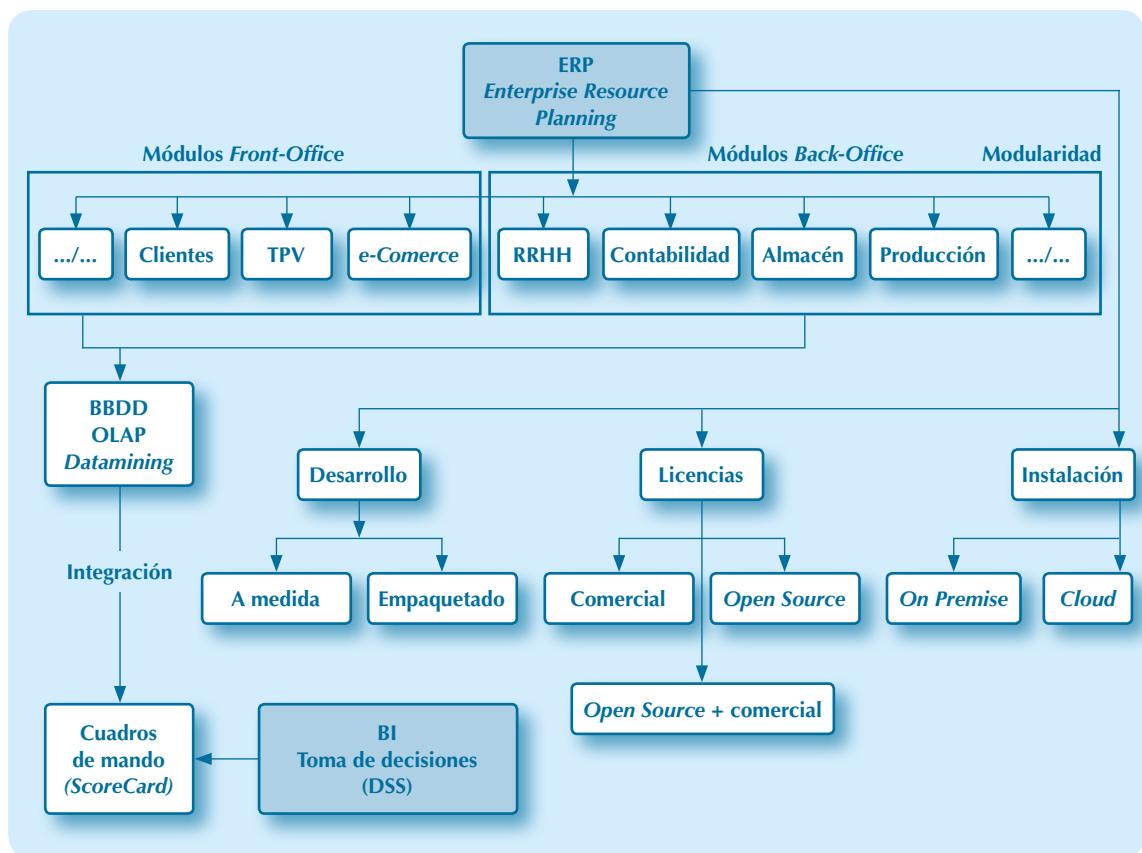
10. **a** **b** **c** **d**

ERP (*Enterprise Resource Planning*)

Objetivos

- ✓ Comprender el concepto de ERP, sus componentes modulares, así como la adaptación de cada uno de ellos a los distintos departamentos de una empresa.
- ✓ Estudiar las distintas opciones de implementación de un ERP en la empresa: las soluciones empaquetadas o los desarrollos a medida.
- ✓ Advertir las ventajas e inconvenientes de la implantación de un ERP.
- ✓ Conocer las tendencias del mercado actual y los tipos de licencias disponibles.
- ✓ Diferenciar las soluciones *On-Premise* y *Cloud*.
- ✓ Relacionar los conceptos ERP, ETL, OLAP, *DataWarehouse*, *DataMining*, *ScoreCard*, BI...
- ✓ Reconocer las principales soluciones propietarias y *Open Source* actuales.

Mapa conceptual



Glosario

B2B (Business to Business). Transacciones comerciales entre empresas.

B2C (Business to Customer). Transacciones entre empresas y consumidores finales.

Back-Office. Relación de actividades de carácter administrativo que tienen que ver con los procesos internos de una empresa. Mantienen a la empresa en funcionamiento.

BigData. Procesamiento de volúmenes muy grandes de datos de todo tipo.

Cloud Computing. Computación en la nube. Servicios de computación (hardware y software) proporcionados a través de la red internet.

e-commerce. Comercio electrónico. Compra y venta a través de internet.

Front-Office. Actividades de una empresa realizadas de cara al cliente. Son las que generan los beneficios.

IoT (*Internet of Things*). Internet de las cosas. Conexión de todo tipo de dispositivos electrónicos a través de internet.

Know-how. En castellano, “saber hacer”. Conocimientos no protegidos de una organización que le proporcionan una ventaja competitiva frente al resto de competidores.

On-Premise. Se dice de la instalación de un sistema en local, es decir, los servicios de computación están localizados en la propia empresa.

SaaS (*Software as a Service*). Software como servicio. Modalidad de computación en la nube por la que el proveedor ofrece licencias de uso de un software determinado a través de internet.

Técnicas ETL (*Extract, Transform and Load*). Herramientas de extracción, transformación y carga posterior (ETC).

2.1. Introducción. ¿Qué es un ERP?

En el capítulo 1 se ha tratado la gran variedad de sistemas de información gerencial que mecanizan *procesos estratégicos, operativos y de gestión*: sistemas de facturación comercial, de gestión de nóminas, de contabilidad, de control de asistencia y permisos, de producción, de gestión de inventario, de logística, de gestión de clientes, etc.

Pero los procesos de negocio generalmente son multifuncionales, es decir, implican varias funciones departamentales. Si cada uno de ellos está gestionado por un software independiente con su base de datos particular, el flujo de información en la empresa no es eficiente pues tiene que trasmítirse de forma no automática entre los departamentos involucrados.

Esto ha motivado que estos sistemas independientes hayan sido sustituidos paulatinamente en la empresa por los ERP (*Enterprise Resource Planning* por sus siglas en inglés).

Un ERP, en castellano (sistema de) planificación de recursos empresariales, es un sistema de información *integrado* compuesto por *módulos* que se corresponden normalmente con funciones departamentales de la empresa y que utiliza una *base de datos común*, que permite automatizar los procesos, compartir la información de los distintos departamentos y gestionar el flujo de trabajo, y proporciona a los directivos información consolidada que les ayudará en el proceso de toma de decisiones debidamente informadas.

En realidad, el software desarrollado para cada función departamental será distinto pero su estructura lógica (el esquema de la base de datos) será compatible con la base de datos común.

Un ERP es algo más complejo que el conjunto de sus módulos por la interoperabilidad mencionada y por eso es objetivamente mejor solución que la de elegir módulos de distintas soluciones del mercado para cada departamento, porque estas no utilizarán la misma base de

datos ni tendrán el mismo esquema, lo que dificultará el flujo automatizado de la información interdepartamental en el día a día de la empresa.

RECUERDA

- ✓ Los objetivos básicos que persigue un ERP son:
 1. El acceso a la información de la empresa de forma inmediata, confiable y precisa.
 2. La optimización de los procesos de negocio y la eliminación de operaciones innecesarias.
 3. La compartición de información entre distintos departamentos.

2.1.1. Ejemplo de un proceso de negocio

Estudiar un supuesto práctico puede aclarar el concepto. La situación es la siguiente: una empresa ha vendido a un cliente un equipamiento informático y lo tiene que preparar y configurar antes de entregárselo.

Esta venta ha implicado la cooperación coordinada de varios departamentos: un departamento *comercial*, quizás como consecuencia de una campaña emprendida por el departamento de *marketing*, ha presupuestado un equipo y unos servicios de configuración y puesta en marcha a un nuevo cliente al que ha dado de alta en su *CRM* y le ha hecho llegar el presupuesto de alguna manera.

En caso de aceptación, el departamento *financiero* ha estudiado la solvencia del cliente para establecer la forma de pago, el departamento de *almacén* ha comprobado a través de su control de *stock* la existencia del material, el departamento de *recursos humanos* o el departamento de *proyectos* ha asignado, en función de su disponibilidad el personal técnico encargado de la configuración del equipo.

Una vez preparado, *logística* se encargará de enviarlo, mientras *contabilidad* lo facturará y añadirá los apuntes contables correspondientes.

Todos estos procesos habrán modificado la base de datos añadiendo transacciones al sistema que afectarán a todos los departamentos involucrados y al sistema gerencial en sí: el ERP habrá recogido y consolidado todos ellos.

Sin embargo, este planteamiento de máximos no debe ocultar que, en la práctica, en las pymes y micropymes esos “departamentos” mencionados se difuminan. Pero eso, siguiendo el razonamiento empleado en el caso de los CRM del capítulo anterior, no significa que un ERP no pueda ser de aplicación en estos entornos pequeños.

En definitiva, los procesos, aun realizados por la misma persona o por un grupo muy reducido de ellas, se tienen que ejecutar, y el ERP permitirá gestionarlos eficientemente desde un único punto.

Simplemente, habrá que optar por el ERP adecuado para el modelo y el tamaño del negocio y utilizar los módulos apropiados para las funcionalidades que no se subcontraten externamente.



Figura 2.1
Proceso de ventas.

En el caso de un autónomo, por ejemplo, preparará el presupuesto y lo enviará. Si no tiene stock (cuanto más pequeño el negocio, menos stock suelen tener), cumplimentará una orden de compra para adquirir el equipamiento al mayorista, por el que recibirá una factura y con el que tendrá negociado un sistema de pago.

Cuando reciba el equipo, lo configurará seguramente él y lo enviará a través de una empresa de logística, o incluso lo entregará él mismo o algún colaborador. Y con seguridad se encargará de hacer las gestiones de cobro pertinentes.

Parece claro que en este escenario no necesitará utilizar los módulos que gestionan los recursos humanos y el módulo de contabilidad porque, con toda probabilidad, los tendrá externalizados, con lo que simplemente se preocupará, en ese caso, de poder exportar los datos de sus actividades a un formato legible por su proveedor de servicios contables y de personal externo.

2.1.2. Origen y evolución del concepto ERP

Fue la consultora Gartner quien utilizó por primera vez el término ERP en 1990 para referirse a los sistemas que habían superado las funcionalidades relacionadas con fabricación y finanzas, ámbitos que hasta entonces gestionaban los MRP II, incluyendo otras para otras funcionalidades necesarias a nivel interno.

Posteriormente, la implantación del comercio electrónico (*e-commerce*) hizo evolucionar a los sistemas ERP a lo que, de nuevo Gartner denominaría *ERP II*, concepto que apareció por primera vez en el artículo “*ERP is dead. Long live ERP II*” publicado en el año 2000.

Mientras que ERP comprende todos los procesos internos, ERP II incluye la gestión de la interacción de la empresa con su entorno, es decir, su relación con proveedores y administraciones mediante la inclusión de funcionalidades de SRM y PLM (véase el cuadro de acrónimos del capítulo 1).

Se puede considerar una tercera generación, los *ERP III*, que integra a los clientes, potenciando las herramientas de colaboración tanto a nivel interno como externo, la nube y añaden el concepto de SaaS, la tecnología de internet para las transacciones sin contacto, las redes sociales, el IoT, y, en definitiva, el *C-Commerce*.

En la actualidad, ya se habla de la cuarta generación de ERP (*ERP IV*), soluciones que añaden al ecosistema de clientes y proveedores a socios, competidores y otros actores y, sobre todo, que utilizan la inteligencia de negocio (BI) y la inteligencia artificial (IA).

Actividad propuesta 2.1



Multitud de autores han definido el concepto de ERP. Algunos de los más utilizados como fuente bibliográfica son: Davenport (1998), Holland y Light (1999), Esteves y Pastor (1999) Kumar y Hillegersberg (2000), Shanks y Seddon (2000) Laudon y Laudon (2000), Orton y Marlene (2004), o McGaughey y Gunasekaran (2009).

- a) Busca las definiciones y compáralas.
- b) Busca definiciones más actuales y compara.

2.2. Características, componentes e infraestructura de un ERP

Para que una solución de gestión empresarial pueda ser considerada un ERP, debe poseer una serie de características y, sobre todo, abarcar el mayor número posible de funcionalidades que, por norma general, se necesitan administrar en una empresa o negocio.

2.2.1. Características de un ERP

Las principales características de un ERP son la *modularidad* y la *integración*:

- a) La *modularidad* se traduce en que cada módulo se corresponde con un área funcional y, aunque utiliza normalmente un paquete de software distinto, comparte la información con el resto de los módulos o paquetes funcionales.
- b) La *integración*: en un ERP los datos se introducen una única vez en el sistema, en una base de datos central accesible desde los distintos módulos, lo que permite el flujo de la información entre ellos de tal manera que la información generada por las transacciones de uno de los módulos pueda ser reutilizada en otro.

Otras características que presentan los ERP de forma generalizada son:

- Están *orientados a los procesos* empresariales.
- Normalmente son desarrollos multiplataforma.
- Permiten el acceso desde múltiples localizaciones con distintos dispositivos.
- Como se han diseñado considerando el funcionamiento básico estándar de las empresas, se pueden implementar en cualquier sector de negocio. Es lo que se conoce por *estandarización*.
- Permiten una cierta adaptación a las particularidades de la empresa, es decir, poseen una cierta *adaptabilidad*.
- Son *integrables* en sistemas que proporcionan la base de información para los cuadros de mando (*Dashboard* y *ScoreCard*) y sistemas BI.
- Gestionan información de la compañía *en tiempo real* poniéndola a disposición de la organización de forma selectiva y jerárquica.
- Son *escalables*, es decir, que pueden crecer de forma natural en función de las necesidades sin comprometer su funcionamiento (en módulos, número de usuarios, tamaño de la BBDD, capacidad de respuesta a número de peticiones, etc.).

2.2.2. Componentes de un ERP

Un ERP se compone de módulos. Enumerar todos los módulos funcionales que puede incluir un ERP es una tarea casi imposible.

En general, los principales desarrollos de software ERP proporcionan aplicaciones para añadir a las funcionalidades mínimas de la instalación base otras funcionalidades adaptadas, en mayor o menor medida, para los siguientes departamentos o funciones empresariales:

- Contabilidad y finanzas.
- Comercial y marketing.
- Recursos humanos.
- Relaciones con el cliente (CRM).
- Compras y proveedores.
- Gestión de almacén.
- Gestión de la producción.
- Gestión de proyectos.
- Comercio electrónico (B2B y B2C).
- Optimización de la cadena de suministros (SCM).
- Gestión de relaciones con proveedores (SRM).
- Gestión de vida del producto (PLM).
- Gestión del conocimiento, *Knowledge Management System* (KMS).
- *Point of Sale* o terminal punto de venta (POS o TPV).
- Etcétera.

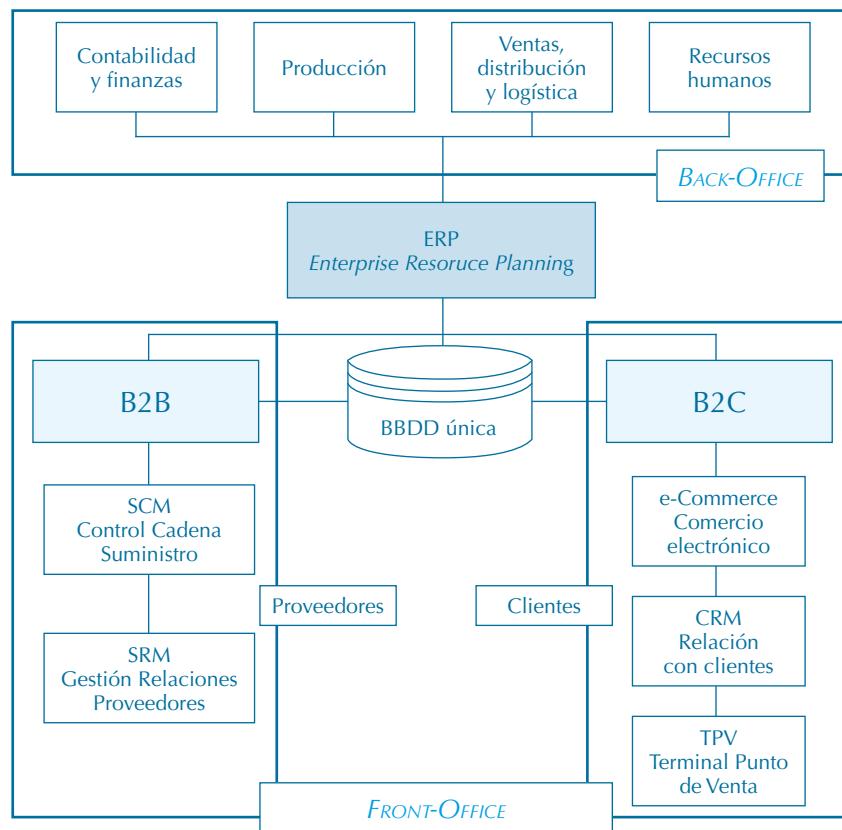


Figura 2.2
Componentes de un ERP.

Estos módulos están englobados en dos grandes grupos: aquellos que gestionan los procesos internos de la empresa como son los que aplican a recursos humanos, almacén, contabilidad y finanzas, producción, etc., conocidos normalmente como módulos de *Back-Office*; y los que tienen relación con el exterior (clientes y proveedores), como pueden ser el módulo de CRM, el de SCM y los relativos al *e-business*, entre otros, que se conocen como módulos de *Front-Office*.

La totalidad de sistemas ERP ofrecen un módulo CRM. Se podría decir que mientras que el CRM está enfocado al cliente y a aumentar las ventas, el ERP está orientado al negocio, a optimizar recursos y procesos y, por tanto, a reducir los costes.



SABÍAS QUE...

Además de estos, dependiendo de los fabricantes desarrolladores de la solución y su especialización, se puede encontrar todo tipo de módulos o *plugins* adaptados a casi cualquier proceso de negocio que uno pueda imaginar.

2.2.3. Detalle de funcionalidades de los principales módulos de un ERP

Se especifican aquí algunos de los módulos más habituales de un ERP con una relación no exhaustiva de sus funcionalidades más representativas. Se reitera que la amplia variedad de oferta de soluciones ERP en el mercado hace que estas clasificaciones teóricas se vean después difuminadas por el desarrollo específico realizado por el fabricante del software ERP.

Pero en líneas generales, sirve para entender las funcionalidades de cada módulo.

A) Contabilidad y finanzas

Este módulo automatiza las operaciones contables. Se alimenta fundamentalmente de la información proporcionada por los módulos de ventas y de compras a proveedores para facturar a clientes y gestionar los pagos de las compras realizadas a los proveedores. Gestinan los impuestos, las cuentas bancarias y, además, suelen proporcionar una contabilidad analítica y una gran variedad de informes.

Tiene que seguir, lógicamente, el plan contable local.

B) Comercial (y marketing)

Otras veces llamado “módulo de ventas y facturación”, en esta amplia categoría se gestiona desde la aparición de una oportunidad, la cotización de presupuestos y las actividades comerciales (visitas, llamadas al cliente, etc.) conducentes al cierre de la venta y la facturación.

Se elaboran las listas de precios, los descuentos, las campañas que se realizarán (en colaboración directa con el departamento de marketing si existiera), etc.

La importancia del marketing y sus nuevas formas y canales hace que, con frecuencia, hayan aparecido módulos específicos.

C) Recursos humanos (RRHH)

Realiza la gestión de personal en sentido amplio. Desde la selección y la contratación de personal a la gestión de la política de remuneraciones y beneficios sociales.

También se encarga de la elaboración de nóminas y su correspondiente cumplimiento normativo, el control de asistencia, de permisos y de vacaciones. En definitiva, gestiona todo lo relativo al personal de la compañía, proporcionando una visión global del rendimiento y los costes de cada departamento.

D) CRM

Como ya se ha comentado, este módulo controla todo tipo de contactos realizados con los clientes. Este módulo, tan específico, se ha estudiado detalladamente en el apartado 1.5, y aunque sigue existiendo como solución independiente, la mayoría de los ERP lo incluyen. Es además uno de los casos que se comentaba anteriormente, de los que se pueden solapar en cuanto a funcionalidades con módulos de facturación, comerciales, de venta y marketing.

E) Compras y proveedores

Con este módulo, que a veces también incluye el almacén, se controla todo lo relativo a proveedores y listas de precios, órdenes de compra, reglas de contratación, métodos de reposición, etc., de manera análoga a lo que hace un módulo de ventas.

Las listas de precios de compra y su control, los contratos con proveedores, la negociación de las condiciones de compra, el seguimiento de pedidos a proveedores, el control de entrega son funcionalidades de este tipo de módulo.

F) Almacén y logística

Gestiona el control de las entregas realizadas por los proveedores, las existencias, el reabastecimiento, la rotación del inventario, el *stock*, los traspasos entre almacenes. Además, gestiona la logística, ya sea propia o subcontratada, mediante módulos de conexión con proveedores de transporte.

G) Otras aplicaciones de interés

Dejando aparte las aplicaciones y módulos más específicos relacionados con mercados verticales y otros de terceras partes, otras aplicaciones que se pueden considerar interesantes desde el punto de vista de una pyme y que constituyen un abanico de posibilidades desde el punto de vista docente, son los siguientes:

- Sitio web.
- Parte de horas.
- Fabricación.
- MRP II.
- PLM.
- Calidad.
- Comercio electrónico.
- *eLearning*.
- Etcétera.

Actividad propuesta 2.2



Estudia las funcionalidades de los módulos relacionados con la fabricación de productos, como *Fabricación, MRP, PLM, Calidad, Trazabilidad*.

2.2.4. Infraestructura

Desde el punto de vista funcional, ha debido quedar claro que los ERP obedecen a una estructura modular en la que cada componente tiene una función específica relacionada con un departamento o función empresarial y que se nutre y a la vez alimenta una base de datos central, común a toda la estructura.

Desde la perspectiva tecnológica, en la actualidad los sistemas ERP están basados en una estructura cliente-servidor soportada sobre bases de datos relacionales.

Desde luego que siguen existiendo soluciones monousuario que requieren la instalación de la aplicación en un único equipo, pues ya se ha dicho que las soluciones ERP son adaptables a cualquier tamaño de negocio, pero el objetivo perseguido es el estudio de soluciones multiusuario.

Así pues, este texto se centrará en las soluciones cliente-servidor, o soluciones distribuidas en una infraestructura general consistente en tres componentes:

- a) El módulo cliente. *Plugin* más o menos pesado, que el usuario ejecutará en su dispositivo electrónico. La tendencia actual es el de las soluciones en las que el cliente accede mediante un simple navegador web (*cliente ligero*). Ya no se utilizan los llamados *clientes pesados*, que eran aquellos que se repartían la inteligencia con el módulo servidor.
- b) Infraestructura distribuida de comunicaciones, sistema que permitirá la comunicación del módulo cliente con el o los módulos servidores a través de cualquier dispositivo y desde cualquier ubicación.
- c) El módulo servidor. Un único o varios módulos que entrañan la inteligencia del ERP para proporcionar la información que el usuario pide a través del módulo cliente. En este componente se incluiría también el lenguaje de programación utilizado para el desarrollo de la solución, la base de datos y el sistema gestor de bases de datos.

La estructura cliente-servidor puede ser de una o varias capas según sigan los modelos:

- Navegador web $\leftarrow \rightarrow$ Aplicación en servidor.
- Navegador web $\leftarrow \rightarrow$ Web intermedia $\leftarrow \rightarrow$ Aplicación en servidor/es.

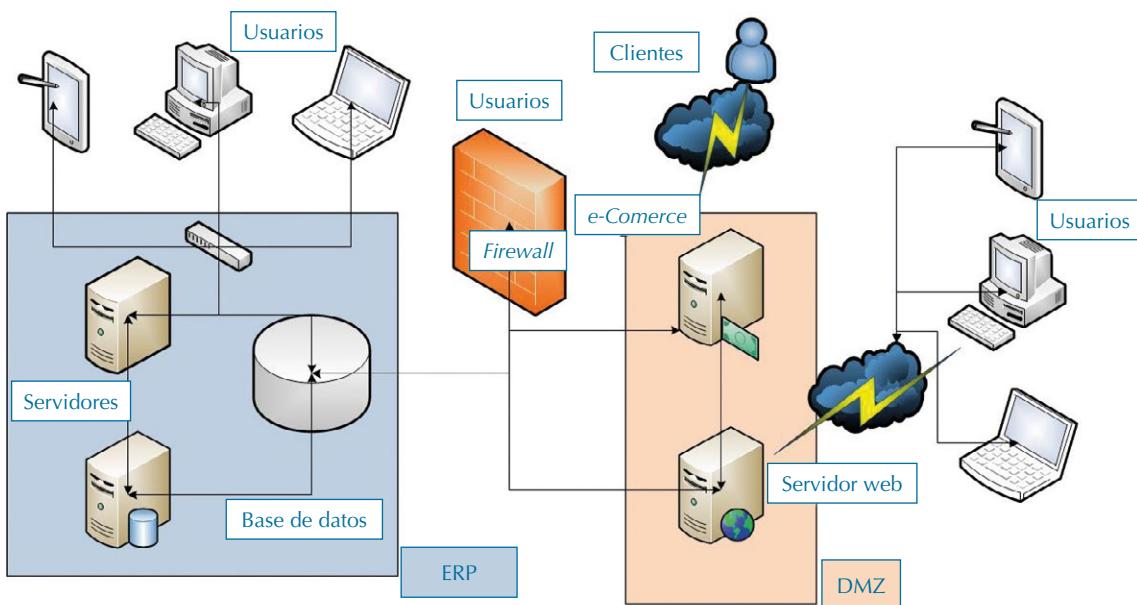


Figura 2.3
Infraestructura general de un ERP.

Además, en un sistema multicapa los servicios suelen ser proporcionados por servidores distintos, con lo que el grado de distribución es mayor. De una manera simple y general, se puede representar la infraestructura de un ERP de la siguiente manera (figura 2.3):

1. Uno o varios servidores, físicos o virtuales que albergan el ERP.
2. Un sistema de comunicaciones protegido por un cortafuego que permite:
 - El acceso de los usuarios desde la red interna.
 - El acceso de los usuarios desde el exterior a la intranet.
 - El acceso de los clientes al módulo de *e-Commerce* desde el exterior (internet).

2.3. Soluciones On-Premise y Cloud. Licenciamiento

Al igual que en cualquier proyecto de instalación de una solución distribuida, en particular una instalación cliente-servidor, se pueden considerar las siguientes opciones de instalación de un ERP:

- Instalación en servidores propios (físicos o virtuales) en la propia sede de la empresa.
- Instalación en servidores físicos o virtuales, propios o alquilados, exclusivos o compartidos, en la sede de un proveedor, es decir utilizando servidores en lo que se conocía hasta hace pocos años como modalidades de *housing* y *hosting*.
- Uso de la infraestructura del proveedor sobre la que este instala y mantiene una instancia de su solución ERP para uso exclusivo de la empresa.

Realmente, considerando no solo la ubicación física sino también el reparto de las responsabilidades sobre cada componente del sistema, se pueden reducir y clasificar en dos grandes tipos de sistemas: los instalados en infraestructuras propias de la empresa y los localizados en la nube.

Tradicionalmente, tanto las soluciones comerciales como las *Open Source* se instalaban utilizando la infraestructura del cliente, modalidad conocida como *On-Premise*, pero actualmente está experimentando un incremento exponencial la tendencia de utilización de sistemas ERP a través de lo que se conoce como la modalidad SaaS (*Software as a Service*) o software como servicio, en español. Esta consiste en que el proveedor proporciona al cliente un acceso a una instancia del ERP instalada en sus máquinas y se compromete a mantenerla y gestionarla. El cliente solo se debe preocupar de acceder al servicio y cumplir con un sistema de retribución basado en el “pago por uso”.

Si la empresa es capaz de vencer ese temor, comúnmente extendido, de poner todos sus procedimientos, datos, resultados e información en manos extrañas, utilizará esta modalidad porque así no tendrá que disponer de una infraestructura física y de personal, y sobre todo, mantenerla.

La externalización de este tipo de servicios tecnológicos actualmente es una opción con costes razonables perfectamente asumibles por las empresas independientemente de su tamaño. De esta manera, las empresas dedican sus recursos más valiosos al negocio en sí para el que se supone que está preparada siguiendo la máxima popular de “zapatero a tus zapatos”.

Serán la cultura empresarial propia y el estudio comparativo entre la opción de tener máquinas, sistemas de comunicaciones, licencias de software, personal técnico, etc., propios inmerso en su estructura del negocio, y la alternativa de hacer el pago en función del uso, de una infraestructura externa habilitada, particularizada y mantenida por el proveedor, los dos factores que influirán en la decisión final.

Es importante tener en cuenta que no siempre las versiones *Cloud* tienen las mismas funcionales que los productos de instalación *On-Premise*.

Actividad propuesta 2.3



Elabora un pequeño informe explicando las diferencias de los conceptos de *Cloud Computing* siguientes: infraestructura como servicio (IaaS), plataforma como servicio (PaaS) y software como servicio (SaaS).

2.3.1. Tipos de licencia

Basándose en el tipo de licencia de uso del software, se pueden considerar dos grandes tipos de sistemas ERP:

- a) ERP propietarios. También llamados privativos, como su propio nombre indica, son *comercializados* por la empresa desarrolladora propietaria del producto y sus *partners*. Tienen un coste de licencia de usuario para el que existen diversos métodos de cálculo que consideran aspectos como el número de usuarios concurrentes o el volumen de datos, entre otros. Además, suelen incluir un sistema de actualizaciones que suponen un coste de mantenimiento recurrente, normalmente de tarificación anual.
- b) ERP libres (*Open Source*). Son desarrollos realizados por una comunidad, *sin ánimo de lucro*, que están disponibles de forma gratuita para quien quiera utilizarlos. Bien es cierto que no siempre es gratuito por completo, porque, alrededor de estas comunidades han aparecido empresas que comercializan servicios o módulos especializados.

Lo fundamental de este tipo de soluciones es que el código fuente está disponible, a diferencia de las soluciones propietarias.

Esta clasificación, por otra parte, es semejante a la que se puede hacer en otros tipos de software. Aunque inicialmente, las soluciones con licencia comercial utilizaban sistemas operativos propietarios como Microsoft Windows y las soluciones *Open Source* estaban desarrolladas para entornos Linux, en la actualidad no está tan marcada esa especialización pues, como ya se ha visto, una de las características de los ERP es que son desarrollos válidos para distintas plataformas y sistemas operativos.



Actividad propuesta 2.4

Repasa los conceptos relacionados con los diferentes tipos de licencias de uso de software *copyright*, *copyleft* y *Creative Commons*, y busca los tipos de licencia bajo los que se distribuyen los ERP más conocidos.

2.4. Tipos, ventajas e inconvenientes de una solución ERP

Cuando los responsables de una empresa se convencen de la necesidad de un ERP para mejorar, abordan un proyecto importante para la viabilidad del negocio que marcará su futuro más o menos inmediato en función del éxito o el fracaso del proyecto.

2.4.1. To do or to buy (or to rent)

A la hora de plantearse la búsqueda de una solución apropiada al negocio, se tiene que escoger entre realizar una aplicación *a medida*, desarrollada de acuerdo con requerimientos propios, o adquirir un *paquete estándar* de entre la gran variedad que existe en el mercado, y adaptarse a él pues estos no suelen ofrecer muchas posibilidades de adaptación.

Se trata de una decisión estratégica clave que después se verá complementada con la elección del producto comercial, en su caso; o con la decisión de si el desarrollo a medida se realiza con recursos internos o se externaliza mediante la subcontratación de una empresa especialista.

El desarrollo a medida permite ajustar la aplicación a las particularidades de la empresa. Realmente, la opción del desarrollo es la más apropiada cuando el negocio tiene un componente de originalidad que dificulta la adaptación de una solución cerrada.

TEN EN CUENTA

- ✓ En el caso del desarrollo a medida, la estimación de coste dependerá de si el desarrollo se aborda con personal interno o se contrata una consultora externa, del alcance del desarrollo y de una variedad tal de condicionantes, que exigen un presupuesto particularizado en cada ocasión.

Por el contrario, las soluciones comerciales disponibles en el mercado proporcionan funcionalidades generalmente demandadas por la mayoría de las empresas, con un menor coste y una mayor rapidez de implementación.

Existen, incluso, soluciones adaptadas a un mercado vertical específico que facilitan aún más el ajuste buscado. Se pueden encontrar las llamadas *soluciones verticales* para una gran variedad de mercados: agrícola, instaladores, despachos, inmobiliario, restauración, etc.

Como ya se ha comentado, se podría añadir, como opción relativamente novedosa, el alquiler de la solución de gestión, en la modalidad de *pago por uso*: la contratación de infraestructura y aplicativo a un proveedor. Cada vez más proveedores ofrecen sus soluciones en esta modalidad.

Respecto a los costes de licenciamiento, se pueden distinguir tres grandes grupos de modelos de precio:

- *Licencia perpetua*. En este caso el pago único por una licencia de uso hace a la empresa cliente propietaria de por vida. No obstante, deberá asegurarse de contratar un sistema de actualizaciones periódicas que mantengan la aplicación actualizada en el tiempo. Hasta hace relativamente pocos años era el modelo más utilizado.
- *Suscripción*. Mediante el abono de una cuota periódica, la empresa cliente se asegura el uso de la aplicación. Es el modelo seguido en la opción SaaS.
- *Open Source*. Licencias gratuitas.

Es, por tanto, importante conocer los distintos tipos de gastos en los que se incurrirá a la hora de tomar una decisión. En la siguiente tabla se resume el entorno económico comparado en el que se distribuyen los distintos gastos:

	COSTES INICIALES	COSTES RECURRENTES		OTROS
LICENCIA PERPETUA	coste inicial de la licencia instalación integración	€\$\$\$\$	Actualizaciones Soporte del fabricante Soporte interno	€\$\$\$ Infraestructura
SUSCRIPCIÓN	instalación integración	€	fee periódico Soporte del fabricante	€\$\$\$\$ Comunicaciones
OPEN SOURCE	instalación integración	€€	Actualizaciones Soporte externo o interno	€€ Infraestructura
Independientes de sistema de licenciamiento: Migración de datos Formación				
		€€€		€\$\$\$\$

Figura 2.4
Entorno de costes de los distintos tipos de licencia.

2.4.2. Importancia de los ERP. La información y el poder

Un mercado global y deslocalizado como el actual, en el que tanto el consumidor como la empresa cliente se han ido acostumbrando a la inmediatez, está obligando a las empresas a adaptar sus sistemas si quieren ser competitivas.

Hoy día no se concibe una empresa que no tenga implantada en mayor o menor medida un ERP. El manejo de la información es una necesidad primordial en cualquier negocio porque permite gestionar los procesos de funcionamiento interno y porque el análisis de esos datos permitirá identificar oportunidades y facilitará la implementación de estrategias empresariales efectivas.

Con los ERP se obtiene información que se utilizará para tomar decisiones operativas, como puede ser la modificación del precio de un producto, y también decisiones estratégicas, como establecer prioridades o quitar una línea de producto.

Tanto si el autor es Bacon, Hobbes o Foucault, la frase “Quien tiene la información, tiene el poder” y sus variaciones como “La información –o el conocimiento– es poder” son de especial aplicación en este ámbito de los sistemas ERP.

RECUERDA

- ✓ Los ERP desempeñan un papel preponderante en el éxito de las grandes, medianas y pequeñas empresas porque permiten disponer de esa información centralizada y casi en tiempo real, manipularla y aprovecharla en la mejora continua imprescindible en cualquier organización de negocio.

Como ya se ha visto, su efectividad se maximiza si se combinan con datos del mercado y de fuentes externas, entrando así en un nivel superior conocido como *Business Intelligence* (BI), en el que se profundizará posteriormente.

Los ERP aumentan la eficiencia de la gestión, mejoran las relaciones con el personal interno y con los agentes externos, facilita el acceso controlado a la información y el conocimiento y reduce los costes empresariales.

La implementación exitosa de un ERP proporcionará a la organización una ventaja competitiva fundamental respecto a su competencia.

2.4.3. Dificultades de implementar un ERP

Es cierto que la implementación de un ERP con garantías de éxito no es una tarea fácil. A lo largo de todas las fases por las que hay que pasar, que se especificarán en próximos capítulos, se presentan muchos y variados inconvenientes. Es relativamente fácil encontrar información al respecto, referida a casi cada tipo y tamaño de negocio, no demasiado tranquilizadora para quien debe tomar la decisión de adoptar este tipo de solución y liderar su implantación en su empresa.

Empezando por la disyuntiva de tener que escoger entre desarrollar un sistema propio, ya sea con personal de plantilla o externo, o la de optar por una solución ya existente en el mercado que se adapte en mayor o menor medida a la actividad propia. Y posteriormente estudiando el tipo de instalación idónea para el propio negocio.

Los gerentes deben plantearse que el éxito no solo depende en la tecnología, sino que hay factores como los que se relacionan a continuación que pueden hacer fracasar la implantación de un sistema ERP:

- Elección inicial respecto al tipo de solución, desacertada.
- Objetivos del proyecto poco claros.
- Expectativas poco realistas.
- Estimaciones erróneas de los recursos necesarios.
- Requisitos del sistema mal definidos.
- Realización insuficiente de pruebas.
- Baja implicación gerencial.

- Mala comunicación entre cliente, desarrolladores y usuarios (la barrera de entrada de la resistencia al cambio y los conflictos entre departamentos).
- Inadecuada gestión del proyecto.

Dos aspectos importantes que se han de tener en cuenta también son la seguridad y el cumplimiento normativo (*compliance*).

Buscando la productividad, los usuarios deben tener acceso al sistema de gestión independientemente de la herramienta utilizada y el lugar donde se encuentre. Esto provoca que el acceso al sistema de gestión esté accesible desde fuera, con el consiguiente incremento de las necesidades de seguridad perimetral. Por ello, muchas pymes que no tienen recursos ni conocimientos suficientes para disponer de una infraestructura propia con altos niveles de seguridad, utilizan las posibilidades del uso del *Cloud Computing*, utilizando tanto la nube pública como la privada o la híbrida, manteniendo en la medida de lo posible los sistemas críticos en la parte privada.

Hay que asumir que, en el caso de decantarse por soluciones SaaS, se está guardando en manos externas el bien más preciado de la empresa: su *know-how*, su información y su capacidad de negocio.

Gestionar el negocio con un ERP implica cambios que deben ser aceptados por todas las partes implicadas de la organización. Se requiere, por tanto, no solo convencimiento y decisión, sino también recursos materiales y humanos, ya sea para la adquisición o para el desarrollo particular, para la implantación y la formación de usuarios, lo que supone una inversión que hay que valorar muy bien.

Sin olvidar que, como ya se ha comentado, el mantenimiento posterior del sistema, en cualesquiera de los casos, comporta costes recurrentes. Respecto al cumplimiento normativo, la implantación paulatina durante los últimos años de estrategias y mecanismos para el control de las actuaciones éticas del personal, de la transparencia empresarial y de los aspectos relacionados con la responsabilidad corporativa, empiezan a estar contempladas también en módulos específicos de los ERP.

A medida que aumenta el número de regulaciones y estándares, la empresa necesita un ERP que ayude a gestionar este entorno en constante evolución y los desarrolladores están proporcionando ya soluciones *ad-hoc*.

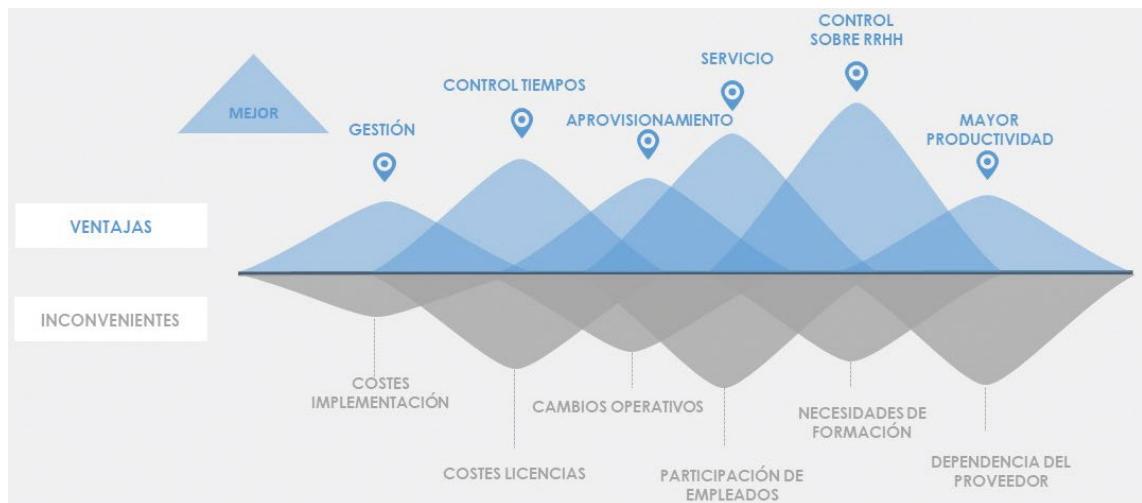


Figura 2.5

Ventajas e inconvenientes de un ERP.

2.5. El mercado actual

A finales de la década de los sesenta y principios de los setenta, nace un mercado de software de gestión empresarial dominado por IBM. En esa época aparecen los fabricantes SAP y J.D. Edwards entre otros.

Posteriormente, en los años ochenta, la revolución de la microinformática hace que más empresas empiecen a utilizarlos para administrar su negocio. Es cuando aparecen la danesa Navision, PeopleSoft, Sage y SP Editores en España (desarrolladores de ContaPlus y FacturaPlus, productos cuyo canal de venta eran los kioscos de prensa).

Estas empresas precursoras han ido experimentando cambios a lo largo del tiempo. Se han ido produciendo adquisiciones, fusiones, absorciones, hasta llegar a la situación actual.

El modelo de negocio tradicional basado en la venta de licencias de software propietario ha experimentado un cambio importante durante la última década, motivado por la proliferación de desarrollos basados en *Open Source*.

En este momento se puede decir que el mercado global está repartido entre dos grandes grupos de fabricantes de software: las empresas que poseen sistemas propietarios y los comercializan y las comunidades de software libre, aunque en este apartado el sistema ha evolucionado hacia empresas que comercializan servicios y mejoras sobre desarrollos *Open Source*.

Sería imposible abarcar la enorme cantidad de soluciones existentes en el mercado, por lo que se trabajará con los tres o cuatro desarrollos más instalados en el mercado español y europeo.

Es tal la oferta existente que cada empresa puede encontrar la más adecuada en un proceso de elección no exento de dificultad, lo que ha dado lugar a la aparición de empresas especializadas en generar informes para ayudar en la elección de la solución más apropiada para cada necesidad empresarial.

Las características que presenta el mercado actual son las siguientes:

- La amplia variedad de ERP verticales.
- La irrupción de los fabricantes de software en el mercado pyme.
- La consagración de las soluciones basadas en software libre (*Open Source*).
- La usabilidad por mimetización gráfica con soluciones de productividad personal conocidas y el uso de navegadores en la parte cliente.
- Las soluciones Cloud y el SaaS.

En los casos de preexistencia de programas equivalentes es importante asegurarse la *migración de los datos* anteriores al nuevo sistema. Por ejemplo, disponer de la contabilidad de los últimos 4-5 años evitará seguir teniendo instalada la solución anterior por motivos fiscales.

TOMA NOTA



La elección de la solución ERP siempre tiene que estar basada en dos premisas fundamentales: la posibilidad de *prueba real* del sistema por la mayor parte de estamentos de la empresa involucrados y el estudio detallado de las opciones de *actualización, soporte y mantenimiento posterior* proporcionado por el fabricante o distribuidor de la solución propietaria, o la empresa comercializadora de los servicios en el caso del *Open Source*.

2.5.1. Soluciones propietarias

De entre las soluciones propietarias se pueden destacar las que comercializan grandes fabricantes como SAP (Business One), Microsoft (Dynamics NAV 365 F&O) u Oracle (Netsuite) sin olvidar a Sage (200 Cloud) en España o a Epicor (iScala) en Hispanoamérica.

Estas herramientas no constituyen la totalidad de la oferta existente, ni siquiera la única solución del fabricante, pero seguramente son las más conocidas del mercado y las tres primeras empresas son, posiblemente, las más reconocidas y de mayor cuota de mercado.

SAP Business One es uno de los más importantes ERP implantados en el sector de fabricación e industria. La compañía alemana es una de las más conocidas en su ámbito. Cuenta con diferentes soluciones adaptadas para cada tipo de cliente, aunque la más conocida es *SAP Business One*. El lenguaje de programación utilizado por SAP es ABAP IV y existe una gran demanda de profesionales especializados en la solución.

Oracle Netsuite es, posiblemente, la solución más extendida entre empresas cuyo modelo de negocio se basa en la venta *online* y los sistemas omnicanal. Este gigante tecnológico es más conocido por su sistema gestor de bases de datos ORACLE. Sin embargo, desde la adquisición de JP Edwards, ha ido ganando protagonismo hasta convertirse, según un estudio de Gartner de 2018, en el sistema ERP líder en el sector. También comercializa *Oracle ERP Cloud*.

Microsoft Dynamics, por su interfaz familiar de Microsoft Office, es el más fácil de utilizar para los usuarios nuevos. Al igual que en el caso de Oracle, la compañía Microsoft, conocida por su sistema operativo Windows, se ha ganado un hueco entre los más utilizados del mundo con su soluciones Dynamics. Según Gartner, actualmente es el ERP con mayor índice de satisfacción de usuarios y mejores características del mercado. Anteriormente conocido como Navision, cambió a Dynamics Nav y, posteriormente, a Microsoft Dynamics 365. Además, incluye funcionalidades de BI.

En general se caracterizan por:

- Sus licencias son de pago y existe un coste recurrente de actualización.
- El código fuente no está disponible.
- Presentan, en general, una cierta dificultad para hacer modificaciones.



Figura 2.6
Matriz de valor de ERP (funcionalidad y facilidad de uso).
Fuente: Nucleus Research.

2.5.2. Soluciones Open Source

En este apartado es oportuno destacar que se trata de aplicaciones que, a priori, no tienen coste pero no siempre son gratuitas. Aunque la licencia básica sí lo es por propia definición del concepto *Open Source*, si no se cuenta con personal con destacados conocimientos técnicos dentro de la organización, será necesario buscar ayuda externa.

Además, han aparecido numerosas empresas que han desarrollado módulos especiales de carácter propietario pero que, sobre todo, proporcionan servicios de instalación, parametrización, formación, soporte y mantenimiento, lo que en definitiva supone un coste.

Por ejemplo, una solución *Open Source* tan conocida como OpenBravo se transformó no hace mucho tiempo en una solución comercial de pago.

En general, se caracterizan por:

- El código fuente está disponible.
- Existen modificaciones accesibles y repositorios abiertos.
- Su implementación exige de mayores conocimientos técnicos.

De entre las más conocidas, se pueden destacar Odoo, Dolibarr, Adempiere, proviene de Compiere, Apache OFBIZ o ERPNext.

2.5.3. Tecnologías más utilizadas

Respecto a las tecnologías que subyacen en las distintas soluciones mencionadas, a título meramente informativo, se presentan en el siguiente cuadro las correspondientes al lenguaje de programación en el que están desarrolladas, la base de datos que utilizan y los entornos en los que se puede instalar la solución.

CUADRO 2.1
Tecnologías utilizadas por los principales desarrollos ERP

Fabricante	Producto	Base de datos	Lenguaje
Microsoft	Dynamics	SQL	C++, C/AL, VB
Oracle	NetSuite	Oracle	C, C++, Java
SAP	Business One	SAP Hana	C, C++
Odoo	Odoo	PostgreSQL	Python
Dolibarr	ERP & CRM	MySQL y PostgreSQL	PHP
Adempiere	ERP	Oracle y PostgreSQL	Java

2.6. La inteligencia de negocio

La inteligencia empresarial (BI), *Business Intelligence*, consiste en el análisis de la información disponible para la obtención de conocimiento sobre la organización, su actividad y su ecosistema.

Mediante un conjunto de herramientas que basan su funcionamiento en la información que se recoge a través del ERP se obtiene una información precisa que facilita la toma de decisiones a nivel operativo, táctico y estratégico. Se trata de una solución complementaria al ERP.

Tiene una estrecha relación con *Big Data* y los conceptos relacionados con este. La combinación de *Business Intelligence* y *Big Data* permite obtener una cantidad inmensamente grande de datos en tiempo real y transformarlos en conocimiento, proporcionando así una ventaja competitiva para los negocios.

Una arquitectura tradicional de un sistema de *Business Intelligence* consiste, a grandes rasgos, en:

1. Una herramienta ETL (Extract, Transform and Load) para recoger datos estructurados de diversas fuentes internas y reorganizarlos según un modelo basado en tablas añadiendo además metadatos.
2. Estos datos transformados por la ETL se almacenan en un *almacén de datos* o *DataWarehouse* (DWH), que no es otra cosa que una base de datos relacional.
3. Las bases de datos OLAP se alimentan de esta información de los sistemas operacionales existentes, y la posterior definición de los KPI posibilitarán obtener cuadros de mando (Dashboard) fiables para la toma de decisiones.

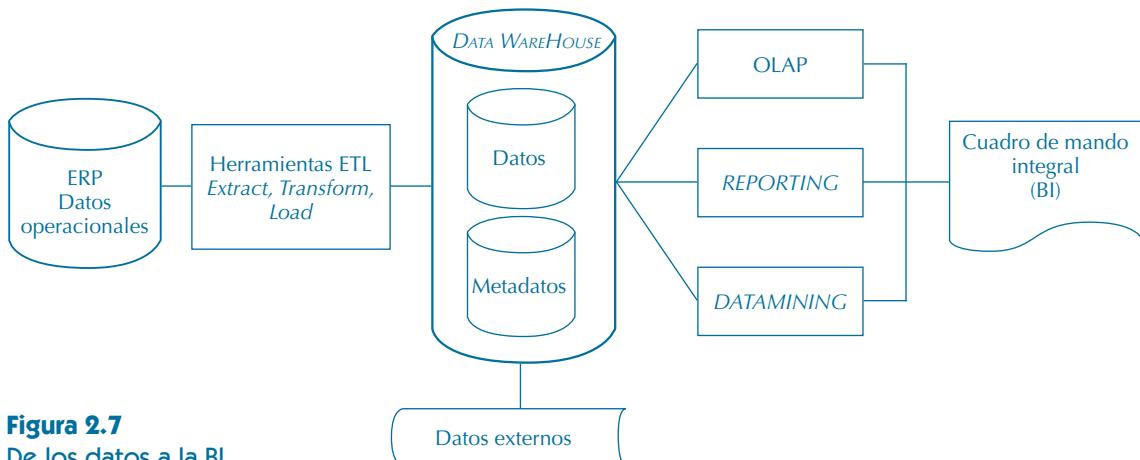


Figura 2.7
De los datos a la BI.

2.6.1. ETL

ETL (*Extract, Transform and Load*) engloba el concepto de herramientas que permiten reunir datos de distintas fuentes, darles un formato común y depurarlos para cargarlos posteriormente en una base de datos o *Data Warehouse*.

A la hora de evaluar las distintas soluciones del mercado, es importante considerar la flexibilidad (plantearse previamente el uso o no del software libre y la no dependencia de un único proveedor), el número de conectores preinstalados, la compatibilidad con el Cloud, la portabilidad y la usabilidad.

2.6.2. OLAP

OLAP (*On-Line Analytical Processing*) o procesamiento analítico de datos es un concepto íntimamente relacionado con las tablas multidimensionales, o cubos.

Se trata de un paso más del concepto de tabla bidimensional normalmente utilizada que se nutre de los sistemas de datos operacionales mediante procesos ETL.

Es la técnica de tratamiento analítico, con acceso normalmente de solo lectura, de los datos que utiliza tres o más dimensiones y que permite presentar resultados de todas las combinaciones posibles.

2.6.3. DataWarehouse

Su traducción literal sería “almacén de datos”. Y es lo que su propio nombre indica: un enorme almacén de datos altamente estructurado.

El concepto fue definido por William H. Inmon, conocido como el padre de DataWareHousing. “Una colección de datos orientada a un tema específico, integrado, variante en el tiempo y no volátil, que soporta el proceso de toma de decisiones”.

Contiene todas las transacciones realizadas por las distintas aplicaciones utilizadas para la gestión del negocio.

2.6.4. DataMining

Textualmente “minería de datos”. Son procesos estadísticos que intentan encontrar patrones ocultos a simple vista para establecer relaciones de causa–efecto entre grupos de datos relacionados, con el objetivo de describir e incluso predecir comportamientos.



Figura 2.8
Relación
ERP y BI.

2.6.5. KPI

KPI (*Key Performance Indicator*). Se corresponde en español con *índicador clave de desempeño*. Ha debido quedar claro a estas alturas que, para poder administrar un negocio, hay que poder medir objetivos concretos en períodos de tiempo determinados. Y para medir hace falta definir los llamados indicadores de medida, es decir, las magnitudes que se quieren evaluar.

¿Qué se puede evaluar?

Multitud de cosas dependiendo de lo que se pretenda: pueden ser magnitudes como el total de ventas en un periodo estacional, el porcentaje de proyectos aceptados, el rendimiento o rentabilidad media de determinado tipo de operación de venta, la ratio de clientes nuevos y el de perdidos, la productividad del personal, etc.

Serán los responsables de la organización los que definan los KPI de cualquier nivel (operacional, de gestión, estratégicos, etc.) que quieren utilizar para administrar su negocio y las herramientas existentes les permitirán controlarlos.

2.6.6. BSC

BSC (*Balanced ScoreCard*) es el cuadro de mando integral que proporciona, mediante una representación gráfica de los KPI, una visión desde una perspectiva concreta o desde la perspectiva global del negocio.

Los sistemas BI, en general, suelen tener un panel de control, habitualmente llamado cuadro de mando (*scorecard*), que permiten obtener de manera gráfica métricas basadas en los KPI que revelan la marcha del negocio y la tendencia.

Actividad propuesta 2.5



- a) Define c-Commerce, e-Commerce y e-Business.
- b) Explica las diferencias entre e-Commerce y e-Business.

Supuestos prácticos

1. Esta práctica consiste en estudiar las características generales de la solución ofrecida por un proveedor de ERP del mercado español, al objeto de poder comparar posteriormente con el trabajo realizado por otros compañeros.

Se trata de elaborar un informe que incluya las especificaciones técnicas, plataformas, tipos de licencia, necesidades de infraestructura, opciones de instalación y soporte a la misma, módulos y características, modalidades de soporte técnico... y todo cuanto se considere interesante tener en cuenta para que el informe se pudiera utilizar para tomar una decisión.

Algunos ejemplos podrían ser los siguientes: SAP Business One, Oracle Netsuite, Microsoft Dynamics NAV, a3 ERP (Wolters Kluwer), Unit4, ADempiere, OpenBravo (español), Odoo, Dolibarr, etc. Pero no se trata de usar los argumentos comerciales que los fabricantes publican en sus páginas web, sino elaborar un informe propio con un enfoque técnico acerca de la solución escogida.

2. Aunque todas las empresas son diferentes, en general tienen en común que se gestionan por departamentos. En el caso de organizaciones muy pequeñas se reduce su número, incluso hasta desaparecer, agrupando las distintas funcionalidades necesarias para el desarrollo del negocio.

- a) Considera una empresa de tamaño apropiado y elabora una relación de los departamentos más habituales (al menos ocho).
- b) Explica la función de cada uno de ellos.
- c) Relaciona los con el/los módulo/s de un ERP (que se seleccione para este ejercicio), que sean apropiados para llevar su gestión de forma automatizada.
3. El fabricante Odoo permite el uso de su ERP en la modalidad SaaS. En la opción Pruébalo gratis de su página web www.odoo.com/es se pueden activar una serie de aplicaciones de la solución (hasta diez).
Explica detalladamente en un informe, en qué consiste esta modalidad de uso, en particular detallando módulos que ofrece, opciones de contratación posibles, servicios ofrecidos, número de usuarios incluidos, duración, precios.

Resumen

- En la actualidad, está generalizado el uso de los ERP por parte de las empresas.
- Los ERP son de aplicación en cualquier empresa, independientemente del tipo de negocio y el tamaño.
- Los ERP proporcionan una ventaja competitiva, pues concentran en un repositorio central toda la información relativa al negocio y sus transacciones.
- Las características principales de un ERP son la modularidad y la integración.
- Los módulos que componen un ERP se pueden clasificar en dos grandes grupos: *Back-Office* y *Front-Office*, según gestionen procedimientos internos o tengan relación con clientes y proveedores, respectivamente.
- Existe gran variedad de módulos normalmente relacionados con los departamentos o funciones departamentales de una empresa.
- Las soluciones ERP siguen una estructura cliente-servidor.
- Respecto al tipo de licencia, pueden clasificarse en propietarios o comerciales y libres u *Open Source*.
- Con relación a la forma de instalación, se pueden clasificar en soluciones *On-Premise* o *Cloud*.
- Aunque sus ventajas son innegables, la implantación de una solución ERP tiene sus riesgos pues hay factores que pueden hacerla fracasar.
- Con conceptos y herramientas, que responden a acrónimos como ETL, OLAP, *Data Warehouse*, *Data Mining*, KPI, ScoreCard..., se obtiene la inteligencia de negocio o BI (*Business Intelligence*).
- El mercado actual de soluciones propietarias está liderado por empresas como Microsoft, SAP, Oracle, etc.
- El mercado de las soluciones *Open Source* es liderado por Odoo.

ACTIVIDADES DE AUTOEVALUACIÓN

1. Los objetivos básicos de un ERP son (señala todas las correctas):
 a) El acceso fácil a la información de la empresa.
 b) La optimización de los procesos de negocio.
 c) La facilidad que proporciona para presentar impuestos locales.
 d) La implantación del e-Commerce.
2. Las opciones “A medida” y “Empaquetado” de un ERP se refieren al tipo de:
 a) Desarrollo.
 b) Licencia.
 c) Instalación.
 d) Implantación.
3. La base de datos de un ERP es:
 a) Única.
 b) Única para cada módulo instalado.
 c) Una para los módulos de *Back-Office* y otra para los de *Front-Office*.
 d) Depende del tipo de instalación (*On-Premise* o *SaaS*).
4. Respecto a los ERP tipo *SaaS*, podemos decir:
 a) La compañía que contrata aprovecha su experiencia en ERP para la adaptación de este a su empresa.
 b) La compañía que contrata el ERP no interviene en las labores de implantación y mantenimiento del mismo.
 c) La compañía que contrata el ERP se encarga del hospedaje de este.
 d) La compañía que contrata el ERP es la que realiza el mantenimiento del mismo así como su operación técnica diaria.
5. Señala la respuesta correcta:
 a) Un ERP monopuesto requiere una máquina cliente más potente que una solución cliente-servidor.
 b) Un ERP cliente/servidor no permite gestionar de forma centralizada el control del ERP.
 c) Un ERP monopuesto depende fundamentalmente de la velocidad de red.
 d) Un ERP cliente-servidor es difícil de mantener.
6. Las soluciones ERP en la modalidad de *cloud computing* utilizan (señala todas las correctas):
 a) La nube pública.
 b) La nube privada.
 c) La nube crítica.
 d) Ninguna de las opciones es cierta.
7. De los siguientes, ¿cuáles son módulos *Back-Office*? (señala todas las correctas):
 a) Producción.
 b) Almacén/Inventario.
 c) TPV.
 d) OLAP.

8. ¿Cuál de los siguientes no es un factor de éxito o fracaso de un ERP?:

- a) Implicación de la dirección.
- b) Resistencia al cambio.
- c) Importe del proyecto.
- d) Falta de medios.

9. Según su modo de trabajo, los módulos (o aplicaciones) de un ERP se pueden clasificar en:

- a) Básicos y opcionales.
- b) Verticales u horizontales.
- c) A medida y empaquetados.
- d) *Back-Office* y *Front-Office*.

10. La función del departamento de compras es:

- a) Guardar el producto en el almacén.
- b) Gestionar el abastecimiento de productos o de los materiales necesarios para crear el producto.
- c) Llevar el producto directamente a la cadena de producción.
- d) Todas son correctas.

SOLUCIONES:

1. **a b c d**

2. **a b c d**

3. **a b c d**

4. **a b c d**

5. **a b c d**

6. **a b c d**

7. **a b c d**

8. **a b c d**

9. **a b c d**

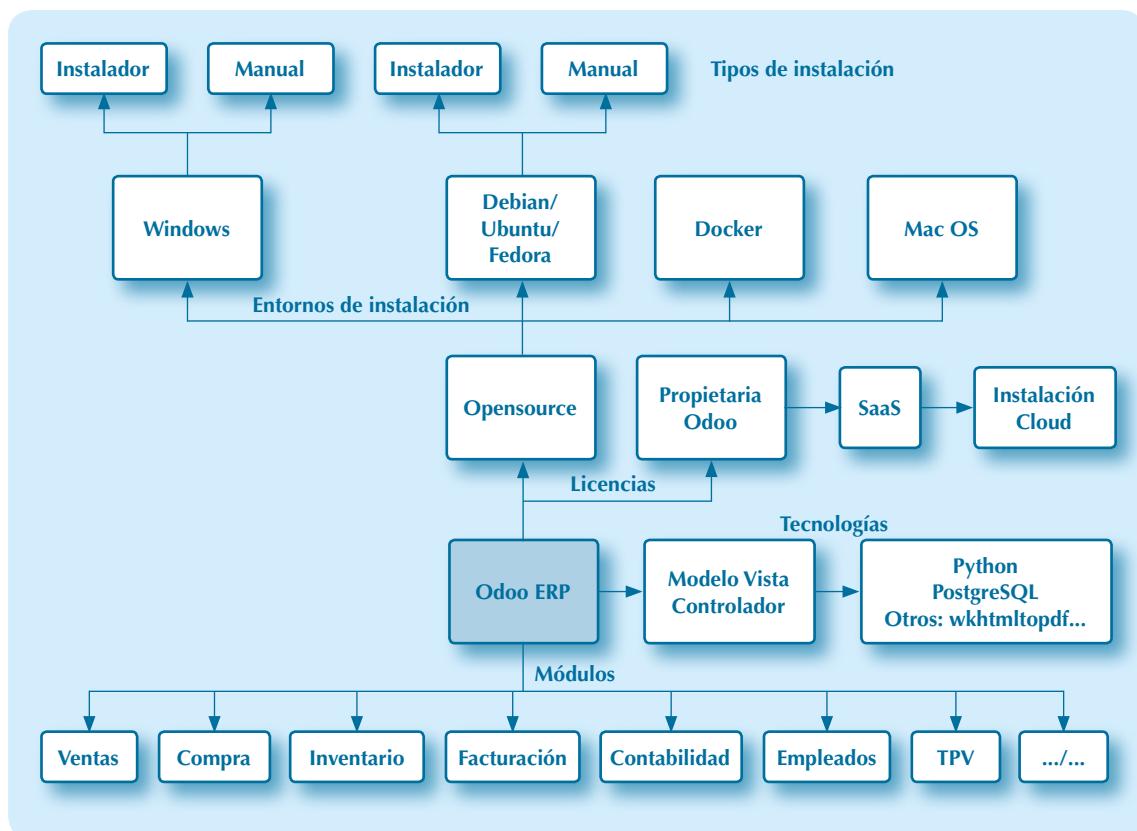
10. **a b c d**

Instalación de un ERP

Objetivos

- ✓ Conocer los requerimientos previos de una instalación en función de su tipo.
- ✓ Distinguir entre instalaciones monopuesto y cliente-servidor.
- ✓ Realizar una instalación de una solución SaaS.
- ✓ Realizar una instalación mediante paquetes (*packaged installers*).
- ✓ Realizar una instalación manual de Odoo ERP en entorno Linux.
- ✓ Conocer someramente los programas y paquetes auxiliares necesarios para el funcionamiento de Odoo ERP.
- ✓ Conocer la estructura general de carpetas creada con la instalación de Odoo ERP.
- ✓ Entender las opciones y condicionantes de los distintos módulos ofertados por Odoo.

Mapa conceptual



Glosario

AGPL (Afferro General Public License). Licencia Pública General de Afferro. Es una licencia CopyLeft derivada de la GPL de GNU que añade la obligación de distribución del software que se destina a dar servicios a través de una red.

BBDD. Bases de Datos.

GPL (General Public License). Licencia Pública General o CopyLeft. Es una licencia que asegura que el software esté abierto para todos los usuarios, que pueden obtener el código fuente, editar lo, modificarlo y redistribuirlo.

LGPL (Lesser General Public License). Licencia Pública General Reducida. Basada en GPL. Licencia mediante la cual las bibliotecas pueden enlazarse con programas no-GPL.

MPL (Mozilla Public License). Licencia híbrida BSD y GNU-GPL de la Fundación Mozilla que permite una cierta reutilización comercial no libre del software.

ORM (Objects Relational Mapping). Mapeo Relacional de Objetos. Modelo de programación que mapea las estructuras de una base de datos relacional sobre una estructura lógica de entidades propias de la Programación Orientada a Objetos.

Packaged Installers. Paquete de instalación asistida que pone el fabricante a disposición del cliente para que pueda realizar la instalación del producto.

RPC (Remote Procedure Call). Llamada a procedimiento remoto. Permite ejecutar código en otra máquina remota de forma transparente a las comunicaciones.

RRHH. Recursos Humanos.

3.1. Introducción

A partir de este momento, este texto se centrará en el proceso de instalación práctica de un ERP en diferentes escenarios y los avances teóricos se realizarán sobre la base de los desarrollos prácticos que se proponen.

Desde un punto de vista docente, parece más interesante para futuros desarrolladores, que son el público objetivo de este libro, profundizar en las instalaciones de productos *Open Source* y sobre sistemas operativos no gráficos en la parte servidor, pues requiere de un mayor cuidado en los detalles y permite adentrarse en las tripas del ERP y modificarlas. No obstante, se trabaja también con soluciones privativas y soluciones *Cloud*.

De hecho, se empezará con una instalación *Cloud* para poder familiarizarse con la operativa y, sobre ella, describir los módulos que, repartidos en cuatro grandes grupos (financieros, comercial y marketing, producción y compras, y recursos humanos), normalmente contienen una solución ERP junto con sus funciones.

Entender las vicisitudes a las que hay que enfrentarse durante el proceso de instalación en un entorno determinado e integrar las distintas tecnologías utilizadas es importante para el desarrollador.

3.2. Requisitos previos de hardware y software. Escenario práctico

Los requerimientos mínimos y recomendados de una solución ERP dependen en gran medida del entorno en el que se va a instalar, de los usuarios que van a hacer uso de la aplicación y del tamaño de las bases de datos. No es fácil, a priori, especificar unos requerimientos generales.

Para cada proyecto de instalación de una solución ERP es necesario estudiar con detalle las magnitudes antes mencionadas. En este texto, utilizando las posibilidades que brinda la virtualización, se pretende construir un escenario con servidores y clientes multiplataforma.

Aunque existen otras soluciones, se utilizará Oracle VM VirtualBox como software de virtualización y sistemas operativos actuales Windows 10, Ubuntu 18.04 y 20.04, y Windows Server 2016 y 2019. Lógicamente, dependiendo de las posibilidades del lector, se pueden utilizar otras

versiones, pues los procedimientos aquí explicados son perfectamente aplicables a otros entornos semejantes con pequeñas variaciones.

Antes de empezar se relacionan las siguientes recomendaciones:

1. Dimensionar procesador, RAM y disco de las máquinas virtuales de acuerdo con la capacidad de la máquina física utilizada.
2. Seguir una nomenclatura para los nombres de las máquinas utilizadas. Por ejemplo, *srv01w* puede ser el nombre de un servidor con sistema operativo Windows Server, mientras *srv03x* se referiría a un tercer servidor Ubuntu.
3. Lo mismo con los equipos de escritorio: *dkp01w* y *dkp02x*.
4. Conviene utilizar siempre los mismos usuarios, por ejemplo, utilizando las iniciales del nombre y su categoría jerárquica en el sistema que se instala: *jjgl.adm* o *jjgl-user01*. (utilizando “.” o “–” dependiendo del sistema operativo).
5. Respecto a la cuenta de correo, si fuera necesaria, se recomienda utilizar el propio nombre del usuario y cualquiera de los servicios webmail, como Gmail.
6. Optimizar las máquinas virtuales teniendo en cuenta para lo que se van a utilizar. Hay servicios y utilidades que no hay porqué instalar.
7. Como se trata de un entorno de estudio y no se va a poner en producción, utilizar siempre la misma contraseña, fácil, para todos los usuarios, independientemente de su rango, pues la seguridad no es importante en este escenario.

En el siguiente apartado se van a detallar tres tipos de instalaciones. En esta ocasión se van a realizar sobre máquinas virtuales con sistemas operativos Microsoft Windows 10 y Ubuntu Desktop 18.04, con idea de que el lector pueda reproducir los procesos de instalación en un entorno no demasiado exigente en lo que se refiere a prestaciones de procesador, RAM y almacenamiento.

En próximos capítulos, en los que se profundizará en aspectos relacionados con la seguridad de acceso y los permisos de usuario, se propondrán entornos de trabajo más apropiados y cercanos a la realidad, basados en sistemas operativos servidor, como Microsoft Windows Server 2019 y Ubuntu Server 20.04.

3.2.1. Odoo ERP

Respecto al software, en este texto se trabajará fundamentalmente con la solución ERP de Odoo.

Odoo S.A., empresa de origen belga, es un ejemplo de lo comentado en apartados anteriores en el sentido de que comunidades que han desarrollado software *Open Source* han evolucionado hacia empresas que, aun continuando con el soporte a la comunidad, han desarrollado módulos de pago, ofrecen servicios de instalación, configuración y soporte y comercializan servicios *Cloud*.

En este, como en otros casos, coexisten ambas organizaciones.

Odoo ERP está disponible en dos versiones: Odoo Enterprise (SaaS con acceso al código restringido a clientes y socios empresariales de Odoo) y Odoo Community (*Open Source*).

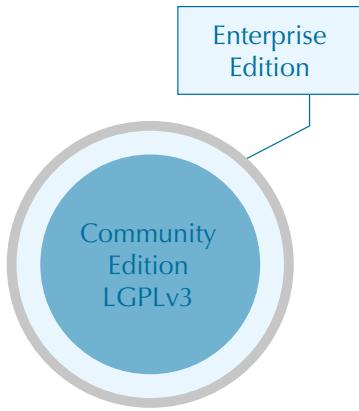


Figura 3.1
Ediciones de Odoo.

Recurso web

www

El código base de Odoo está alojado en Github en <https://github.com/odoo/odoo>.

Respecto a la ayuda disponible, además de la gran cantidad de páginas particulares y de empresas que proporcionan soporte, se puede encontrar en <https://help.odoo.com>, foro de ayuda oficial de Odoo, marco en el que participan tanto desarrolladores como algunos empleados de Odoo.

Además, existe un entorno de pruebas automatizadas denominado Runbot al que se puede acceder desde la dirección <http://runbot.odoo.com/runbot>. En él pueden ejecutarse instancias de las últimas versiones estables.

Odoo ERP es una de las soluciones *Open Source* más conocidas y utilizadas. Dispone de una comunidad de desarrolladores sin ánimo de lucro que proporciona gran cantidad de información, soporte y ayuda en la Odoo Community Association (OCA), accesible en <https://odoo-community.org/> y que comparte el código fuente de los módulos y aplicaciones desarrollados por sus miembros en <https://github.com/OCA>.

Desde 2015 la mayoría de los módulos de Odoo disponen de licencia de código abierto LGPLv3 (todos los de la versión Community). Sin embargo, la versión Enterprise se encuentra bajo licencia comercial distribuida por la empresa belga Odoo S.A.

Basado en una arquitectura de tipo cliente-servidor en la que los clientes son navegadores web que acceden al servidor mediante RPC, Odoo, antes OpenERP hasta su versión 7 y anteriormente TinyERP, utiliza la siguiente tecnología:

- Lenguaje de programación: Python.
- Base de datos: PostgreSQL.
- Lenguajes Web: HTML5, JavaScript.
- Motor de plantillas XML: QWeb (para generar fragmentos y páginas HTML).



PARA SABER MÁS

Este ERP nació en 2005, desarrollado en Python y PostgreSQL bajo condiciones *Open Source*. Su nombre inicial, Tiny ERP proviene de su sencillez. Entre 2007 a 2011 su facturación creció un 1549 %. Pasó a llamarse OpenERP, siguiendo con su filosofía libre y abierta. En 2014 experimentó una considerable mejora con una interfaz mucho más cuidada, una mayor integración de los datos y más estabilidad, y pasó a llamarse Odoo.

Respecto a su modelo de negocio, Odoo cuenta con dos versiones: Enterprise (requiere licencia de pago) y Community (*Open Source*).

Consulta los siguientes videos para saber más:

- Qué es Odoo (Loredana Pepe) QR1
- Cinco razones para elegir Odoo ERP (Eduardo Paz Lloveras) QR2



QR1



QR2

Odoo implementa el Modelo Vista Controlador (MVC), patrón de arquitectura de software en el que el Modelo son los datos, la Vista es la interfaz de usuario y el Controlador la inteligencia o lógica de negocio que controla la interacción con el usuario. En particular:

- Modelo: tabas PostgreSQL.
- Vistas: archivos XML.
- Controlador: objetos Python.

3.3. Modalidades de instalación

Como ya se ha visto, existen dos opciones: la opción *Cloud* y la *On-Premise*.

Como primera opción, se instalará en este apartado una solución SaaS accedida desde una máquina cliente con cualquier sistema operativo. Posteriormente, se realizarán dos instalaciones *On-Premise*: una instalación local en una máquina Windows y otra en una máquina Ubuntu. En este último caso, se hace sin entorno gráfico, pues es el modo de trabajo habitual cuando se trata de servidores Linux.



PARA SABER MÁS

Consulta la tabla comparativa de las funcionalidades que ofrecen ambas versiones en la página web de Odoo:



Por tanto, salvo en la instalación SaaS, se utilizarán los instaladores (*Packaged Installers*) de la versión Community. Estas son instalaciones de las denominadas *automáticas*, que permiten poca variación en la configuración.

Odoo proporciona instaladores empaquetados (tanto de la versión Community como de la Enterprise) para Windows, distribuciones basadas en *deb* para Debian, Ubuntu, etc., y distribuciones basadas en *RPM* para Fedora, CentOS, etc.).

Pero este tipo de instalaciones no son siempre la forma escogida para una instalación real, ya que en los sistemas productivos normalmente hay que adaptarse a la infraestructura y será necesario optar por instalaciones personalizadas. En este caso, es recomendable utilizar la instalación *manual*, con el código fuente, descargándolo desde <https://github.com/odoo/odoo>.

WWW

Recurso web

Existe también imagen básica para Docker en <https://github.com/odoo/docker> e información en https://registry.hub.docker.com/_/odoo/

3.3.1. Instalaciones monopuesto

Se llaman así a las instalaciones más simples: aquellas en las que todos los componentes necesarios de la aplicación están en la misma máquina que, por lo general, utiliza un único usuario o, al menos, un único usuario simultáneamente. Las instalaciones monopuesto se han quedado obsoletas. Aún hay algunas empresas que las ofrecen pero, por norma general, estas instalaciones se han sustituido por las soluciones *Cloud* en las que el usuario solo necesita un navegador y un acceso a internet.

Actualmente, la mayor parte de las soluciones ERP proporcionan un servidor que es accesible desde varios dispositivos y por varios usuarios. El hecho de que se instalen en un equipo con sistema operativo *desktop* no las convierte en instalaciones monopuesto. En realidad, se trata de instalaciones cliente-servidor en las que se ejecutan tanto el servidor como el cliente en la misma máquina.



Actividad propuesta 3.1

Busca una solución ERP que proporcione una instalación de tipo monopuesto.

3.3.2. Instalación de solución *Cloud*: Odoo ERP

En este apartado se describirá la instalación y configuración de la edición en línea basada en la nube, de Odoo ERP.

Para conseguir una instalación propia del ERP de Odoo lo primero que hay que hacer es darse de alta en la página www.odoo.com/es_ES y acceder al apartado “Pruébalo gratis”.

La página permitirá seleccionar una serie de módulos (hasta 10 por ser una versión de prueba) antes de pedir los datos personales. Se puede comprobar aquí la modularidad de los ERP en general y de este en particular.

Figura 3.2
Licencias para educación.

Una vez seleccionados los módulos es el momento de proporcionar los datos personales y aceptar el Acuerdo de Suscripción y la Política de Privacidad.

Odoo ofrece a los estudiantes la posibilidad de disponer de una licencia por diez meses. Para ello, a la hora de poner el nombre de la empresa se debe añadir el prefijo “edu-” e indicar que eres un estudiante.

TEN EN CUENTA

- ✓ El nombre de empresa que se ponga en el formulario será el nombre de la BBDD asociada a la instancia.

Si no ha habido problemas, con estas pocas operaciones se habrá configurado de forma básica una instancia del ERP de Odoo en la modalidad de licencia SaaS y se podrá acceder a la pantalla de aplicaciones. Para ello, bastará con responder al correo de activación que se ha recibido en la cuenta proporcionada.

La página de acceso al ERP desde cualquier dispositivo con acceso a internet y un navegador es *nombre-empresa.odoo.com/web/login*



Figura 3.3
Pantalla principal prueba con los iconos de los módulos escogidos.

Es importante destacar que en el ícono de *Ajustes/Opciones generales* existe un apartado donde se puede activar el *modo desarrollador*, necesario para todas las operaciones de configuración que se van a abordar en próximos apartados.

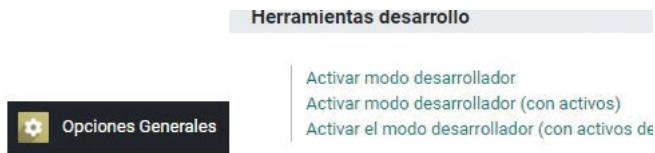


Figura 3.4
Activación de modo desarrollador.

La variante *modo desarrollador (con activos)* añade a las funciones del *modo desarrollador* la posibilidad de depurar el código JavaScript y CSS que se envía al navegador. De momento no será necesaria su activación, bastará con el *modo desarrollador*.

Los menús de configuración de las distintas aplicaciones son sustancialmente diferentes según esté activado o no. Al activar el modo desarrollador se activan las *Open Developer Tools* que proporcionan un menú avanzado dirigido a administradores y desarrolladores para poder

disponer de más opciones de configuración, ver los campos de las tablas, acceder al código de formularios y vistas, editar el apartado de acciones de menú, etc.

Odoo ofrece un amplio apartado de documentación de usuario y para desarrolladores, así como una sección de *e-learning* interesante.

Durante las siguientes conexiones a la página web de Odoo con el usuario utilizado para el registro, se podrá gestionar la base de datos creada. Se profundiza sobre este aspecto en el capítulo 8.

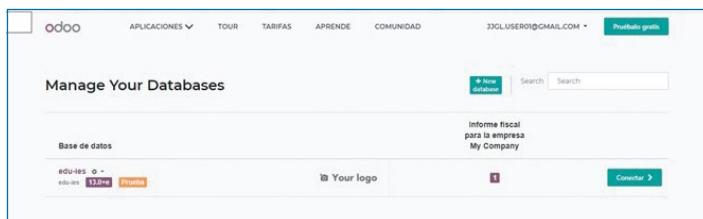


Figura 3.5
Gestión de las BBDD.



Actividad propuesta 3.2

Configura la instancia con los datos básicos: nombre, empresa, dirección completa, teléfono, CIF, email, etc., y añade un logo en el apartado correspondiente.

3.3.3. Instalación para Windows con instalador

Para obtener el instalador para el entorno deseado, basta con registrarse en la página https://www.odoo.com/es_ES/page/download y descargar el paquete correspondiente.

En esta ocasión, una vez descargado el instalador de la versión Odoo 13.0 para Windows, el procedimiento es muy simple: basta con pulsar *Next* en las pantallas que van apareciendo al ejecutar el asistente de instalación, eso sí, tomando la precaución de ir anotando las opciones por defecto que ofrece respecto al servidor, puerto, usuario y contraseña de la base de datos PostgreSQL o las modificadas en su caso.

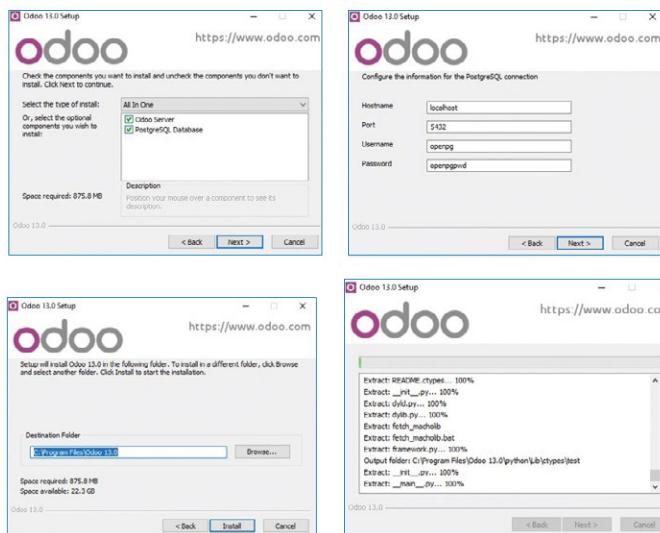


Figura 3.6
Instalación
Windows.

Finalizado el proceso de extracción y de instalación de los servicios necesarios, el ERP quedará instalado en la máquina.

Por defecto, el puerto por el que Odoo escuchará es el 8069. Así, en este caso, una conexión a `localhost:8069` permitirá la conexión al ERP desde la máquina en la que se encuentra instalado, mientras que una conexión a `dirección-ip-máquina:8069` permitirá el acceso desde otra máquina de la red.

La creación de la BBDD durante la primera ejecución o la elección de una BBDD cuando existan varias se hace desde la dirección `localhost:8069/web/database/selector` o `dirección-ip-servidor:8069/web/database/selector`.

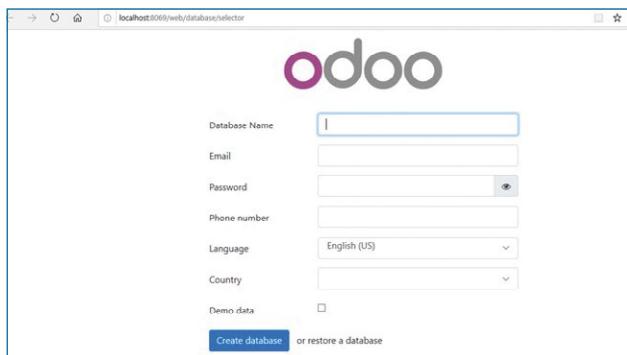


Figura 3.7
Creación de la base de datos.

Existe la opción de cargar la BBDD con un conjunto de datos para demostración (opción *Demo data*). Puede ser una opción interesante para familiarizarse con la solución con algunos datos relativos a usuarios, clientes, proveedores, productos, etc., ya precargados.

Al finalizar el proceso de creación, aparecerá la ventana de aplicaciones de Odoo. En esta ocasión se puede observar que se muestra la opción de instalación de más de cincuenta módulos.

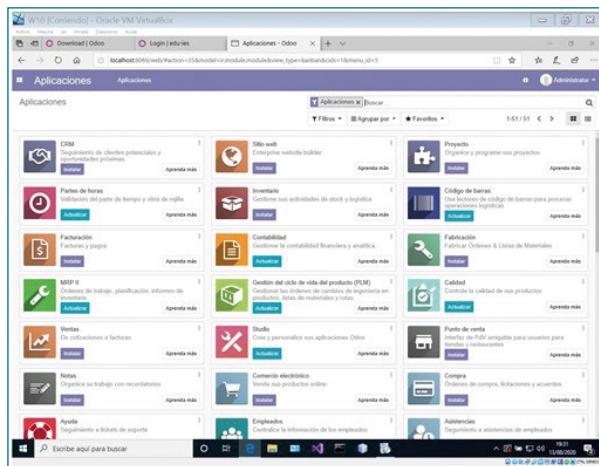


Figura 3.8
Resultado final.

Este número de aplicaciones (y módulos) ofertados es distinto, dependiendo de que se instale la versión para Windows o Linux o la versión SaaS.

Uno a uno, se podrán instalar todos los módulos que se deseen. Es recomendable trabajar en *modo desarrollador*.



Actividad propuesta 3.3

Crea una nueva base de datos en la instalación sobre Windows y carga los datos de prueba. Consigue un listado de contactos en formato pdf.

3.3.4. Instalación en Ubuntu desktop mediante paquete .deb

También se puede descargar un paquete “.deb” para la instalación en entorno Ubuntu desde la página oficial de descargas de Odoo.

El paquete “.deb” de Odoo 13.0 actualmente es compatible con Ubuntu 18.04 y Debian Buster o superiores.

Siendo un entorno Linux, para la instalación bastaría en principio con ejecutar, una vez descargado el paquete, el comando:

```
# dpkg -i <ruta_al_paquete_instalación>
```

```
jjgl-adm@dkp01x: ~
jjgl-adm@dkp01x:~$ clear
jjgl-adm@dkp01x:~$ sudo dpkg -i /home/jjgl-adm/Descargas/odoo_13.0.latest_all.deb
```

Figura 3.9
Instalación de paquete .deb.

Pero esto fallará por las dependencias faltantes. Odoo necesita que tanto la BBDD PostgreSQL como el lenguaje Python estén instalados previamente.

```
jjgl-adm@dkp01x: ~
jjgl-adm@dkp01x:~$ sudo dpkg -i /home/jjgl-adm/Descargas/odoo_13.0.latest_all.deb
Seleccionando el paquete odoo previamente no seleccionado.
(Leyendo la base de datos ... 132075 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar .../odoo_13.0.latest_all.deb ...
Desempaquetando odoo (13.0.20200814) ...
dpkg: problemas de dependencias impiden la configuración de odoo:
  odoo depende de python3-babel; sin embargo:
    El paquete 'python3-babel' no está instalado.
  odoo depende de python3-decorator; sin embargo:
    El paquete 'python3-decorator' no está instalado.
  odoo depende de python3-docutils; sin embargo:
    El paquete 'python3-docutils' no está instalado.
  odoo depende de python3-feedparser; sin embargo:
    El paquete 'python3-feedparser' no está instalado.
  odoo depende de python3-gevent; sin embargo:
    El paquete 'python3-gevent' no está instalado.
  odoo depende de python3-html2text; sin embargo:
    El paquete 'python3-html2text' no está instalado.
  odoo depende de python3-jinja2; sin embargo:
    El paquete 'python3-jinja2' no está instalado.
  odoo depende de python3-langsass; sin embargo:
    El paquete 'python3-langsass' no está instalado.
```

Figura 3.10
Errores de dependencias.

Para ello, previamente se hará:

1. *Instalación de Python:* Ubuntu 18.04 ya tiene instalada la versión 3 de Python lo que se puede comprobar, por ejemplo, con:

```
$ sudo apt list --installed | grep python
```

Pero necesita de la instalación de paquetes adicionales para corregir dependencias ausentes:

```
$ sudo apt --fix-broken install
$ sudo apt-get install python3-pip
```

2. *Instalación de servidor de PostgreSQL.*

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

```
jjgl-adm@dkp01x: ~
selecting default shared_buffers ... 128MB
selecting default timezone ... Europe/Madrid
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

Success. You can now start the database server using:

/usr/lib/postgresql/10/bin/pg_ctl -D /var/lib/postgresql/10/main -l logfile start

Ver Cluster Port Status Owner Data directory Log file
10 main 5432 down postgres /var/lib/postgresql/10/main /var/log/postgresql/postgresql-10-main.log
update-alternatives: utilizando /usr/share/postgresql/10/man/man1/postmaster.1.gz para proveer
/usr/share/man/man1/postmaster.1.gz (postmaster.1.gz) en modo automático
Configurando postgresql (10+190ubuntu0.1) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para ureadahead (0.100.0-21) ...
Procesando disparadores para systemd (237-3ubuntu10.42) ...

jjgl-adm@dkp01x:~$
```

Figura 3.11
Instalación exitosa.

Además, es recomendable instalar otros paquetes auxiliares:

- *wkhtmltopdf* para la generación correcta de los documentos pdf (exactamente la versión 0.12.5 es la recomendada para trabajar con encabezados y pies de página para Odoo 10.0 y superiores). En el caso de Ubuntu 18 en una máquina de 64 bits, después de descargar el paquete correspondiente desde <https://github.com/wkhtmltopdf/wkhtmltopdf/releases/tag/0.12.5>, se ejecuta:

```
$ sudo apt-get install /<ruta_al_paquete_instalación>/wkhtmltox_0.12.5-1.bionic_amd64.deb
```

- *xlwt*. Paquete de Python para poder exportar a formato xls.

```
$ sudo pip3 install xlwt
```

- *num2words*. Paquete de Python para la conversión de números a texto.

```
$ sudo pip3 install num2words
```

Se ejecuta `$ sudo -H pip3 install...` si se obtiene aviso de deshabilitación de la caché por problemas de propietario y permisos.

Ahora sí que será posible la instalación de Odoo 13:

```
# dpkg -i <ruta-paquete-instalación>
```

De esta manera, el paquete de instalación instala Odoo como servicio, lo configura para que se ejecute al inicio y configura el usuario de PostgreSQL para Odoo.

La carpeta Odoo estará en `/usr/lib/python3/dist-packages`

Dependiendo de las versiones utilizadas del sistema operativo Ubuntu y de Odoo, la instalación se complicará más o menos. Conforme han ido apareciendo nuevas versiones de Odoo, el proceso de instalación se ha ido mejorando.

Como último paso, si se quiere incluir Odoo en los repositorios para obtener las futuras actualizaciones junto con el resto del sistema con `apt-get upgrade`, es necesario ejecutar la siguiente secuencia (con usuario `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
```

TOMA NOTA



Los prompt `#` y `$` son utilizados de forma indistinta en el texto. Es necesario recordar que las instalaciones deben hacerse con suficientes privilegios por lo que o se utiliza el Superusuario (`#`) o se utiliza sudo (`$`).

Si al ejecutar

```
$ dpkg -i <ruta-paquete-instalacion>
```

faltaran dependencias, aparecerá el error:

"dpkg: error al procesar el paquete odoo (--install):
problemas de dependencias - se deja sin configurar."

Entonces, los siguientes pasos:

```
# apt-get install -f
# dpkg -i <ruta_al_paquete_instalación>
```

deben arreglar el problema.

3.3.5. Instalación manual en Ubuntu server con fuentes obtenidas de github

En determinadas circunstancias, sobre todo en instancias que se van a poner en producción, es necesario tener un mayor control sobre la instalación. En estos casos es recomendable realizar lo que se conoce por una instalación manual en vez de las automáticas descritas en los apartados anteriores.

Este tipo de instalación proporciona un mayor control sobre la configuración y la parametrización de los servicios, la ubicación de los diferentes componentes del software, el acceso al código fuente y por tanto a las posibilidades de modificación y de creación de nuevos módulos.

A partir de ahora se trabajará con la versión Community que es la *Open Source*.

El código fuente de Odoo Community se puede obtener de <https://github.com/odoo/odoo> como descarga de archivo *zip* o a través de *git*.

Los siguientes pasos constituyen una forma de hacer la instalación de Odoo 13.0 en un equipo con sistema operativo Ubuntu Server 18.04 configurando el servicio para que arranque en el inicio:

a) Instalación

Paso 1. Crear en el sistema el usuario *odoor* con permisos administrativos. Este será el usuario utilizado para la instalación manual. Su directorio de inicio puede ser */opt/odoor*, que es donde se suelen instalar servidores en entornos de producción. Una forma de crearlo es mediante:

```
$ sudo useradd -m -d /opt/odoor -U -r -s /bin/bash odoor
$ sudo passwd odoor      para establecer contraseña
$ adduser odoor sudo
```

Paso 2. Con usuario *odoor* obtener el código de Odoo. Se hace clonando el repositorio de Odoo en GitHub. Comprobar que git está instalado (por ejemplo, con # *git -version*).

```
# git clone https://github.com/odoo/odoo.git --depth 1 --branch 13.0
--single-branch
```

Con --depth 1 solo se descargará el último commit de la versión 13.0 en el directorio / opt/odoor/ que es el home del usuario *odoor*.

```
odoor@srv01x:~/odoor
odoor@srv01x:~$ git clone https://github.com/odoo/odoo.git --depth 1 --branch 13.0 --single-branch
Cloning into 'odoor'...
remote: Enumerating objects: 28567, done.
remote: Counting objects: 100% (28567/28567), done.
remote: Compressing objects: 100% (23363/23363), done.
remote: Total 28567 (delta 8297), reused 12385 (delta 4314), pack-reused 0
Receiving objects: 100% (28567/28567), 121.96 MiB | 5.53 MiB/s, done.
Resolving deltas: 100% (8297/8297), done.
Checking out files: 100% (25609/25609), done.
odoor@srv01x:~$ ls -l
total 4
drwxrwxr-x 10 odoor odoor 4096 Oct 17 13:35 odoor
odoor@srv01x:~$ cd odoor
odoor@srv01x:~/odoor$ ls -l
total 100
-rw-rw-r-- 1 odoor odoor 803 Oct 17 13:34 CONTRIBUTING.md
-rw-rw-r-- 1 odoor odoor 433 Oct 17 13:34 COPYRIGHT
-rw-rw-r-- 1 odoor odoor 43529 Oct 17 13:34 LICENSE
-rw-rw-r-- 1 odoor odoor 1491 Oct 17 13:34 MANIFEST.in
-rw-rw-r-- 1 odoor odoor 2113 Oct 17 13:34 README.md
-rw-rw-r-- 1 odoor odoor 1734 Oct 17 13:34 SECURITY.md
drwxrwxr-x 337 odoor odoor 12288 Oct 17 13:34 addons
drwxrwxr-x 3 odoor odoor 4096 Oct 17 13:34 debian
drwxrwxr-x 10 odoor odoor 4096 Oct 17 13:34 doc
drwxrwxr-x 11 odoor odoor 4096 Oct 17 13:35 odoobin
-rw-rw-r-- 1 odoor odoor 180 Oct 17 13:34 odoobin
-rw-rw-r-- 1 odoor odoor 1331 Oct 17 13:35 requirements.txt
drwxrwxr-x 4 odoor odoor 4096 Oct 17 13:35 setup
-rw-rw-r-- 1 odoor odoor 876 Oct 17 13:35 setup.cfg
-rw-rw-r-- 1 odoor odoor 1698 Oct 17 13:35 setup.py
odoor@srv01x:~/odoor$
```

Figura 3.12
Obtención
del código
de Odoo desde
GitHub.

Paso 3. Comprobar que Python3 está instalado en su versión 3.6 o superior (no es necesario comprobarlo en Ubuntu 18.04) y que pip3 no está instalado. Con el comando:

```
$ python3 -- version
```

Se comprueba si python3 está instalado y si la versión es la adecuada (Python 3.6 o superior). Ubuntu 18.04 trae instalada Python 3.6.9

La instalación del gestor de paquetes escritos en Python pip3 se hace con:

```
$ sudo apt-get install python3-pip
```

```
odoor@srv01x:~/odoor$ sudo apt-get install python3-pip
[sudo] password for odoor:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu build-essential cpp cpp-7 dh-python
  dpkg-dev fakeroot g++-7 gcc gcc-7-base libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan4 libatomic1 libbinutils
  libc-dev-bin libc6-dev libcc1-0 libcilkrt5 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libgcc-7-dev libgomp1 libis119 libitm libisano0 libmpc3 libmpx2
  libpython3-dev libpython3.6-dev libquadmath0 libstdc++-7-dev libtsan0 libubsan0
  linux-libc-dev make manpages-dev python-pip-whl python3-crypto python3-dev python3-distutils
  python3-keyring python3-keyrings.alt python3-lib2to3 python3-secretstorage
  python3-setuptools python3-wheel python3-xdg python3.6-dev
Suggested packages:
  binutils-doc cpp-doc gcc-7-locales debian-keyring g++-multilib g++-7-multilib gcc-7-doc
  libstdc++-7-dbg gcc-multilib autoconf automake libtool flex bison gdb gcc-doc
  gcc-7-multilib libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg libasan4-dbg liblsan0-dbg
  libtsan0-dbg libubsan0-dbg libcilkrt5-dbg libmpx2-dbg libquadmath0-dbg glibc-doc bzr
  libstdc++-7-doc make-doc python-crypto-doc gnome-keyring libkf5wallet-bin
  girl1.2-gnomekeyring-1.0 python-secretstorage-doc python-setuptools-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu build-essential cpp cpp-7 dh-python
  dpkg-dev fakeroot g++-7 gcc gcc-7-base libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan4 libatomic1 libbinutils
  libc-dev-bin libc6-dev libcc1-0 libcilkrt5 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libgcc-7-dev libgomp1 libis119 libitm libisano0 libmpc3 libmpx2
  libpython3-dev libpython3.6-dev libquadmath0 libstdc++-7-dev libtsan0 libubsan0
  linux-libc-dev make manpages-dev python-pip-whl python3-crypto python3-dev python3-distutils
  python3-keyring python3-keyrings.alt python3-lib2to3 python3-pip python3-secretstorage
  python3-setuptools python3-wheel python3-xdg python3.6-dev
0 upgraded, 57 newly installed, 0 to remove and 5 not upgraded.
Need to get 91.2 MB of archives.
After this operation, 253 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Figura 3.13
Instalación
de Python3-pip.

```
$ pip3 -- version
```

Se comprueba la versión es la pip 9.0.1 from Python 3.6.

Paso 4. Instalación de PostgreSQL

```
$ sudo apt-get install postgresql postgresql-client
```

```
selecting default timezone ... Etc/UTC
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

Success. You can now start the database server using:

/usr/lib/postgresql/10/bin/pg_ctl -D /var/lib/postgresql/10/main -l logfile start

Ver Cluster Port Status Owner Data directory Log file
10 main 5432 down postgres /var/lib/postgresql/10/main /var/log/postgresql/postgresql-10-main.log
update-alternatives: using /usr/share/postgresql/10/man/man1/postmaster.1.gz to provide /usr/share/man/man1/postmaster.1.gz (postmaster.1.gz) in auto mode
Setting up postgresql (10+190ubuntu0.1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
Processing triggers for systemd (237-3ubuntu10.42) ...
odoor@srv01x:~/odoor$
```

Figura 3.14
Instalación
de PostgreSQL.

Para dotar de contraseña al usuario `postgres`:

```
$ su
$ su postgres
$ psql
=# alter user postgres with password 'postgres';
=# \q
```

Odoo no permite la conexión con el usuario por defecto `postgres`, por lo que es necesario añadir el usuario utilizado hasta ahora, `odoo`, como supersuario con privilegios para crear bases de datos. Para ello:

```
$ sudo -u postgres createuser -s odoo
$ createdb odoo
```

```
odoo@srv01x:~/odoo$ psql
psql (10.14 (Ubuntu 10.14-0ubuntu0.18.04.1))
Type "help" for help.

odoo=# \l
              List of databases
   Name    | Owner     | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----+
  odoo   | odoo      | UTF8    | C.UTF-8 | C.UTF-8 |
 postgres | postgres  | UTF8    | C.UTF-8 | C.UTF-8 | =c/postgres
 template0 | postgres  | UTF8    | C.UTF-8 | C.UTF-8 | =c/postgres=CTc/postgres+
 template1 | postgres  | UTF8    | C.UTF-8 | C.UTF-8 | =c/postgres=CTc/postgres+
(4 rows)

odoo=#
```

Figura 3.15
Listado de BBDD.

Paso 5. Dependencias y fichero `requirements.txt`. Ahora será necesario instalar las herramientas de desarrollo y dependencias nativas siguientes:

```
sudo apt install python3-dev libxml2-dev libxsllt1-dev libldap2-dev
libsasl2-dev libtiff5-dev libjpeg8-dev libopenjp2-7-dev zlib1g-dev
libfreetype6-dev liblcms2-dev libwebp-dev libharfbuzz-dev libfribidi-dev
libxcb1-dev libpq-dev
```

Las dependencias de Odoo se enumeran en el archivo ubicado en el directorio raíz de Odoo llamado `requirements.txt`. Por ello, es el momento de instalar todas las dependencias que aún no estén instaladas mediante:

```
$ cd odoo
$ sudo -H pip3 install setuptools wheel
$ sudo -H pip3 install -r [path]requirements.txt
```

Paso 6. Instalación de utilidades. Como en el caso de la instalación del apartado 3.3.4, será necesario instalar las utilidades `wkhtmltopdf`, `xlwt` y `num2words` que permitirán la exportación

de datos a documentos pdf, xls y convertir formatos. Además, el paquete xfonts-75dpi que necesita wkhtmltopdf. Se hace aquí la primera de ellas:

```
$ wget https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download
/0.12.5/wkhtmltox_0.12.5-1.bionic_amd64.deb
$ sudo apt-get install xfonts-75dpi
$ sudo dpkg -i wkhtmltox_0.12.5-1.bionic_amd64.deb
```

```
odoo@srv10x:~$ sudo apt-get install xfonts-75dpi
Reading package lists... done
Building dependency tree
Reading state information... done
The following packages were automatically installed and are no longer required:
adwaita-icon-theme-at-spi2-core avahi-daemon dconf-gsettings-backend dconf-service geoclue-2.0 glib-networking-services gsettings-desktop-schemas gstreamer1.0-plugins-base gtk-update-icon-cache iio-sensor-proxy libatk-bridge2.0-0 libatk1.0-0 libcairo2.0-0 libavahi-client3 libavahi-glib libbrotli libcairo-gobject2 libcairo2 libcdparanoia0 libcolord2 libcroco3 libcurl2 libgd-pixbuf2.0-0 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-common libgeoclue-2.0 libgl1-mesa-glx libgtk-3-0 libgtk-3-bin libgtk-3-common libhyphen0 libjson-glib-1.0-0 libjson-glib-1.0-common libmbus0 liborc-0.4-0 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libproxy1v5 libgmi-glib libqt5network5 libqt5positioning5 libqt5printsupport5 libqt5qlm5 libqt5quick5 libqt5sensor5 libqt5svg2-2 librsvg2-common libsoup-gnome2.4-1 libsoup2.4-1 libthai-data libthai0 libwayland-cursor0 libwayland-egl libwoff1 libxcb-icccm4 libxcb-image0 libxcb-keysyms1 libxcb-randr0 libxcb-shm0 libxcb-xinerama0 libxcb-xkb1 libcomposite1 libxkbcommon-x11-0 libxkbcommon0 linux-headers-4.15.0-112-generic linux-modules-4.15.0-112-generic linux-modules-extra-4.15.0-112-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
xfonts-75dpi
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
1 not fully installed or removed.
Need to get 3368 kB of archives.
After this operation, 3638 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu bionic/universe amd64 xfonts-75dpi all 1:1.0.4+nmul [3368 kB]
Fetched 3368 kB in 1s (2637 kB/s)
Selecting previously unselected package xfonts-75dpi.
(Reading database ... 164640 files and directories currently installed.)
Preparing to unpack .../xfonts-75dpi_1%3a1.0.4+nmul_all.deb ...
Unpacking xfonts-75dpi (1:1.0.4+nmul) ...
Setting up xfonts-75dpi (1:1.0.4+nmul) ...
Setting up wkhtmltox (1:0.12.5-1.bionic) ...
Processing triggers for fontconfig (2.12.6-0ubuntu2) ...
odoo@srv10x:~$ ^C
odoo@srv10x:~$ sudo dpkg -i wkhtmltox_0.12.5-1.bionic_amd64.deb
```

Figura 3.16
Instalación de la utilidad wkhtmltopdf.

Si todo ha ido bien, se puede probar y lanzar el servidor de Odoo de forma manual (con el usuario propietario de la base de datos postgres creada y desde el directorio de Odoo) con el comando:

```
$ python3 odoo-bin --addons-path=addons -d "nombre_base_de_datos_sin_comillas" -i "nombre_base_de_datos_sin_comillas"
```

b) Configuración

Para no tener que lanzarlo de forma manual, en esta fase se creará, con el usuario *odoo* utilizado durante la instalación, el fichero de configuración *odoo.conf* en el home de la aplicación (en este caso */opt/odoo*) con el siguiente contenido mínimo:

```
[options]
admin_passwd = postgres

xmlrpc = True
```

```

xmlrpc_port = 8069

db_host = 127.0.0.1
db_port = 5432
db_user = odoo
db_password = odoo

addons_path = /opt/odoo/odoo/addons

```

A partir de este momento se puede ejecutar Odoo manualmente con:

```
$ /opt/odoo/odoo/odoo-bin -c /opt/odoo/odoo/odoo.conf
```

Además, para disponer de un fichero de registros se crea el directorio odoo dentro de */var/log* y se añade la opción --logfile:

```
$ /opt/odoo/odoo/odoo-bin -c /opt/odoo/odoo/odoo.conf --logfile
/var/log/odoo/odoo.log
```

De esta manera se arranca odoo manualmente y se añaden los registros en *odoo.log* (no aparecerá entonces información en el terminal).

Para que en */var/log/odoo/odoo.log* se registren los logs de odoo es buena práctica el crear una carpeta odoo a la que el grupo del usuario odoo tenga acceso con permiso de escritura:

```

$ sudo mkdir odoo
$ sudo chgrp odoo odoo/
$ sudo chmod g+w odoo/

```

c) Inicio automático

Para asegurar que el servidor de Odoo se lanzará cuando se inicie el servidor, lo que es muy importante en sistemas en producción, hay que configurarlo como servicio del sistema.

Para ello se crea el fichero */etc/systemd/system/odoo.service* con el contenido mínimo siguiente:

```

[Unit]
Description=Odoo 13
After= postgresql.service

[Service]
Type=simple
User=odoo
Group=odoo
ExecStart=/opt/odoo/odoo/odoo-bin -c /opt/odoo/odoo/odoo.conf --log-
file /var/log/odoo/odoo.log

[Install]
WantedBy=multi-user.target

```

Este fichero se arranca [para/reinicia] con:

```
$ sudo systemctl start [stop/restart] odoo.service
```

Y para que se ejecute al inicio:

```
$ sudo systemctl enable odoo.service
```



Actividad propuesta 3.4

- Explica el concepto nightly build.
- Obtén el código de odoo de nightly build.

3.3.6. El problema de las dependencias en las distintas modalidades de instalación de Odoo

Ya se ha comentado que Odoo necesita que tanto Python como PostgreSQL estén instalados en el sistema. Además, depende de paquetes y librerías varias, que en el caso de las distintas distribuciones Linux, se encuentran tanto en repositorios de la propia distribución como en repositorios de Python. Una forma de controlar estas dependencias y de no ir resolviéndolas conforme van apareciendo es la de conseguir la información detallada del paquete construido (.deb en Ubuntu) desde el repositorio oficial <https://nightly.odoo.com/>.

Archivo	Última actualización
.. /	
Packages	09-Dec-2020 02:52
Release	09-Dec-2020 02:52
Release.gpg	09-Dec-2020 02:52
Sources	09-Dec-2020 02:52
odoo_13.0.20191010.dsc	10-Oct-2019 05:17
odoo_13.0.20191010.tar.xz	10-Oct-2019 05:17
odoo_13.0.20191010_all.deb	10-Oct-2019 05:18
odoo_13.0.20191010_amd64.changes	10-Oct-2019 05:19
odoo_13.0.20191011.dsc	11-Oct-2019 05:17
odoo_13.0.20191011.tar.xz	11-Oct-2019 05:17
odoo_13.0.20191011_all.deb	11-Oct-2019 05:18
odoo_13.0.20191011_amd64.changes	11-Oct-2019 05:18
odoo_13.0.20191020.dsc	20-Oct-2019 05:12
odoo_13.0.20191020.tar.xz	20-Oct-2019 05:12
odoo_13.0.20191020_all.deb	20-Oct-2019 05:13
odoo_13.0.20191020_amd64.changes	20-Oct-2019 05:13
odoo_13.0.20191021.dsc	21-Oct-2019 05:13
odoo_13.0.20191021.tar.xz	21-Oct-2019 05:13
odoo_13.0.20191021_all.deb	21-Oct-2019 05:14
odoo_13.0.20191021_amd64.changes	21-Oct-2019 05:14

Figura 3.17
Índice de la versión 13 en nightly.odoo.com.

En el archivo de texto *Packages* de la carpeta *deb* de la versión que se va a instalar se encuentra toda la información al respecto.

Por ejemplo, para la versión 13 el contenido de este fichero es:

```

Package: odoo
Version: 13.0.20201209
Architecture: all
Maintainer: Odoo S.A. <info@odoo.com>
Installed-Size: 572979
Depends: python3-babel, python3-dateutil, python3-decorator, python3-docutils, python3-feedparser, python3-gevent, python3-html2text, python3-jinja2, python3-libsass, python3-lxml, python3-mako, python3-mock, python3-ofxparse, python3-passlib, python3-pil, python3-polib, python3-psutil, python3-psycopg2, python3-pydot, python3-pyparsing, python3-pypdf2, python3-qrcode, python3-reportlab, python3-requests, python3-serial, python3-tz, python3-usb, python3-vatnumber, python3-vobject, python3-werkzeug, python3-xlsxwriter, python3-zeep, python3:any, adduser, lsb-base, postgresql-client, python3-chardet, python3-suds, python3-xlrd
Recommends: postgresql, python3-ldap
Conflicts: openerp, openerp-server, openerp-web, tinyerp-server
Replaces: openerp, openerp-server, openerp-web, tinyerp-server
Filename: ./odoo_13.0.20201209_all.deb
Size: 61786916
MD5sum: d397fb9cde5c997a47a17103d8e33fe9
SHA1: b9083211d61eecce26958dd8c0561349d08191e0
SHA256: fa6bb8ebbe84273a3bba88c73fc6afd960d87ea62cdfe491568ff600511fea2
Section: net
Priority: optional
Homepage: http://www.odoo.com/
Description: Open Source Apps To Grow Your Business Odoo, formerly known as OpenERP, is a suite of open-source business apps written in Python and released under the LGPLv3 license. This suite of applications covers all business needs, from Website/Ecommerce down to manufacturing, inventory and accounting, all seamlessly integrated. Odoo's technical features include a distributed server, flexible workflows, an object database, a dynamic GUI, customizable reports, and an XML-RPC interface. Odoo is the most installed business software in the world. It is used by 2.000.000 users worldwide ranging from very small companies (1 user) to very large ones (300 000 users).

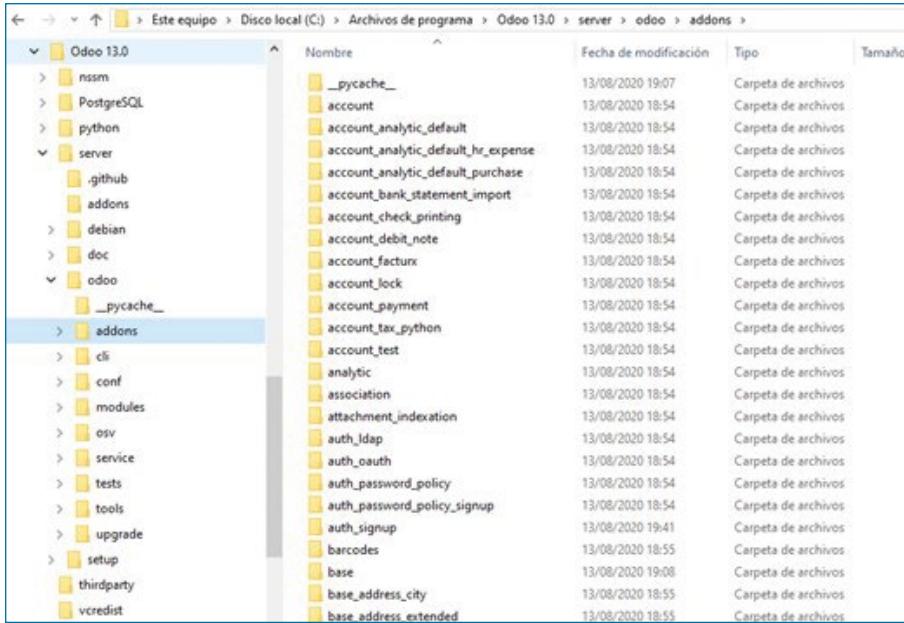
```

3.3.7. Estructura general de carpetas de la aplicación

Utilizando la instalación realizada en la máquina Windows 10, una simple visualización mediante el *Explorador de Archivos* proporcionará una visión de la estructura de carpetas creadas en la instalación de Odoo 13.

Posteriormente se profundizará sobre esta estructura y, en particular, se trabajará con la carpeta *odoo/addons* que contiene una carpeta por cada uno de los módulos de Odoo (figura 3.19).

Respecto a la BBDD PostgreSQL, mediante *pgAdminIII*, cliente gráfico multiplataforma para gestionar PostgreSQL, instalada por defecto en Ubuntu 18.04 Desktop, se puede obtener una vista de su estructura.

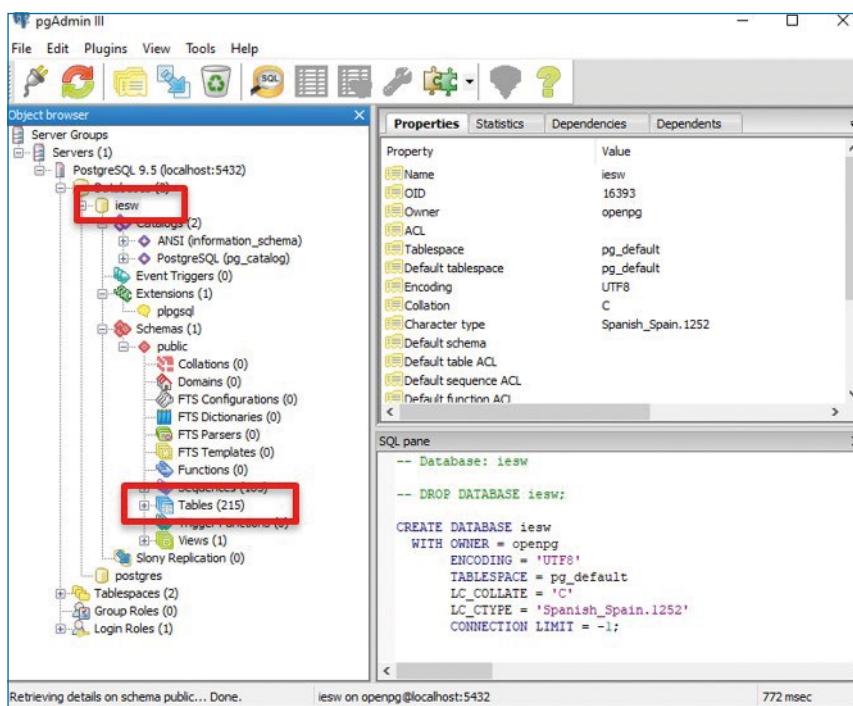


The screenshot shows a file explorer window with the following path: Este equipo > Disco local (C:) > Archivos de programa > Odoo 13.0 > server > odoo > addons. The 'addons' folder is selected. The table lists the contents of this folder:

Nombre	Fecha de modificación	Tipo
__pycache__	13/08/2020 19:07	Carpeta de archivos
account	13/08/2020 18:54	Carpeta de archivos
account_analytic_default	13/08/2020 18:54	Carpeta de archivos
account_analytic_default_hr_expense	13/08/2020 18:54	Carpeta de archivos
account_analytic_default_purchase	13/08/2020 18:54	Carpeta de archivos
account_bank_statement_import	13/08/2020 18:54	Carpeta de archivos
account_check_printing	13/08/2020 18:54	Carpeta de archivos
account_debit_note	13/08/2020 18:54	Carpeta de archivos
account_facturx	13/08/2020 18:54	Carpeta de archivos
account_lock	13/08/2020 18:54	Carpeta de archivos
account_payment	13/08/2020 18:54	Carpeta de archivos
account_tax_python	13/08/2020 18:54	Carpeta de archivos
account_test	13/08/2020 18:54	Carpeta de archivos
analytic	13/08/2020 18:54	Carpeta de archivos
association	13/08/2020 18:54	Carpeta de archivos
attachment_indexation	13/08/2020 18:54	Carpeta de archivos
auth_ldap	13/08/2020 18:54	Carpeta de archivos
auth_oauth	13/08/2020 18:54	Carpeta de archivos
auth_password_policy	13/08/2020 18:54	Carpeta de archivos
auth_password_policy_signup	13/08/2020 18:54	Carpeta de archivos
auth_signup	13/08/2020 19:41	Carpeta de archivos
barcodes	13/08/2020 18:55	Carpeta de archivos
base	13/08/2020 19:08	Carpeta de archivos
base_address_city	13/08/2020 18:55	Carpeta de archivos
base_address_extended	13/08/2020 18:55	Carpeta de archivos

Figura 3.18
Carpeta addons.

Observar el nombre de la BBDD (que es el proporcionado durante la instalación) y el número de tablas. Además de estas tablas básicas, conforme se vayan añadiendo nuevos módulos, irán apareciendo nuevas tablas (figura 3.19):



The screenshot shows the pgAdmin III interface with the following details:

- Object browser:** Shows the database structure. A red box highlights the 'Tables (215)' node under the 'public' schema.
- Properties pane:** Displays properties for the 'iesw' database, including Name: 'iesw', OID: '16393', Owner: 'openpg', Tablespace: 'pg_default', Encoding: 'UTF8', Collation: 'C', Character type: 'Spanish_Spain.1252', Default schema: 'public', Default table ACL: 'pg.default', Default sequence ACL: 'pg.default', and Default function ACL: 'pg.default'.
- SQL pane:** Shows the SQL code for creating the 'iesw' database:

```
-- Database: iew
-- 
-- DROP DATABASE iew;

CREATE DATABASE iew
  WITH OWNER = openpg
    ENCODING = 'UTF8'
    TABLESPACE = pg_default
    LC_COLLATE = 'C'
    LC_CTYPE = 'Spanish_Spanish.1252'
    CONNECTION LIMIT = -1;
```

Figura 3.19
La base de datos de Odoo.



PARA SABER MÁS

Se puede utilizar una versión más moderna de la utilidad: pgAdmin IV. Prueba su instalación. Por defecto, el servidor PostgreSQL solo permite conexiones desde localhost. Es lo aconsejado para un servidor en producción para no comprometer la seguridad del sistema. Pero, si se quiere utilizar pgAdminIV desde otra máquina de la red, es necesario hacer las siguientes modificaciones:

- En el archivo /etc/postgresql/[version]/main/pg_hba.conf introducir una línea como la siguiente: *host all all 192.168.1.0/24 md5* para conexión desde cualquier equipo de la red.
- En el archivo /etc/postgresql/[version]/main/postgresql.conf, cambiar la línea *listen_addresses ='localhost'* y cambiar por *'*'*, lo que hará que PostgreSQL escuche conexiones en el puerto 5432.

No obstante, en Ubuntu, se puede trabajar desde la consola ejecutando *psql* con el usuario *postgres* o el usuario *odoo* creado al efecto.



```
jjgl-admin@dkp01x:~$ sudo su postgres
postgres@dkp01x:/home/jjgl-admin$ psql
psql (10.14 (Ubuntu 10.14-0ubuntu0.18.04.1))
Type "help" for help.

postgres=#
```

Figura 3.20
Consola psql.

Algunos comandos básicos de psql son:

<code>postgres=# \du</code>	relación de usuarios
<code>postgres=# \dt</code>	listado de tablas
<code>postgres=# \l</code>	listado de bbdd
<code>postgres=# connect "bbdd"</code>	conectar a una bbdd
<code>postgres=# \q</code>	para salir de psql

Y una relación más detallada de los comandos se pueden obtener ejecutando:

<code>postgres=# help</code>	ayuda general
<code>postgres=# \?</code>	ayuda sobre comandos psql
<code>postgres=# \h</code>	ayuda sobre comandos SQL

3.4. Tipos de módulos

Hay que recordar aquí que, por regla general, una empresa fabrica y/o comercializa un producto y/o presta un servicio. Para ello compra a proveedores, proporciona un valor añadido con

recursos humanos propios o externos y materiales, calcula el coste y en función de este y de distintos gastos imputables, vende a un cliente por un precio que debe proporcionar un beneficio. Todas estas actividades deben poder controlarse con el ERP elegido.

Además de la clasificación vista en el capítulo 2 (apartados 2.3.2 y 2.3.3) respecto a la tipología interna o externa de los procesos que automatiza, desde las primeras publicaciones que definían y desarrollaban la idea de lo que era o debía ser un ERP, se ha coincidido con más o menos matices, en la idea de clasificar los módulos de un ERP, *atendiendo a su funcionalidad*, en cuatro grandes grupos:

- a) *Módulos para gestionar los procesos de fabricación*: los que controlan la planificación de la producción si la hubiera, el aprovisionamiento de materiales, el equipamiento y máquinas, el inventario y la logística.
- b) *Módulos para gestionar los procesos financieros y contables*: para el control de gastos, inversiones e ingresos, la gestión de los flujos financieros, la generación de informes y además para el cumplimiento legal contable y fiscal.
- c) *Módulos para gestionar los procesos de marketing y ventas*: los que gestionan las acciones de publicidad y marketing para llegar al cliente, permiten elaborar presupuestos y órdenes de venta, planifican las ventas, crean las distintas listas de precios y controlan la distribución y facturación.
- d) *Módulos para gestionar los procesos de RRHH* que organizan y controlan la contratación del personal, las retribuciones, la asistencia, los permisos, etc.

Es cierto que esta clasificación no es estanca y que a veces un módulo de una solución de un fabricante determinado podría catalogarse en más de una de estas categorías o, dicho de otra forma, proporciona la gestión de procesos que pertenecen a más de una de estas categorías. No obstante, se trata de una clasificación acertada, y aceptable en términos generales.

3.4.1. Módulos básicos y específicos

Los módulos *básicos* indispensables en casi cualquier organización, independientemente del tamaño, son los que gestionan la operativa de compras y ventas.

Una empresa, por muy pequeña que sea, debe, como mínimo, gestionar internamente este apartado y procurar obtener una información de salida sobre su actividad que sea exportable fácilmente a sus proveedores de servicios externos que le proporcionen, por ejemplo, la contabilidad, la gestión de personal, el marketing y publicidad, etc.

Se puede añadir a este planteamiento de mínimos un módulo de CRM para llevar una, ya justificada en apartados anteriores, necesaria gestión de las relaciones con el cliente.

Cuando el tamaño de la empresa lo vaya demandando, la asunción de funcionalidades por parte de departamentos internos dotados de personal propio, como puede ser un departamento financiero, se generará la necesidad de añadir nuevos módulos al ERP inicial.

Después, la casuística tan amplia de tipos de actividades empresariales necesitará de la utilización de módulos más *específicos*. Así, por ejemplo, una empresa de prestación de servicios técnicos de soporte informático necesitará añadir a su ERP un módulo de control de tiempo de asistencia (*ticketing* o *helpdesk*) que una empresa de fabricación de comida no necesitará y, por el contrario, esta sí necesitará un módulo que contemple la trazabilidad de los alimentos que utiliza en sus recetas.

Actividad propuesta 3.5



- Estudia si tiene sentido una solución de tipo ERP para un profesional independiente, como un médico.
- ¿Existen operaciones de venta? ¿Tendría sentido el uso de un módulo de compras? ¿Qué otros módulos pueden considerarse de aplicación a este entorno?
- Averigua si existen soluciones verticales y compara sus particularidades con un ERP estándar.

3.4.2. Instalación de aplicaciones y módulos en Odoo

En el caso de Odoo, como en otros productos similares, existe una estructura troncal que se instala inicialmente (estructura básica del ERP). El resto de las aplicaciones ofertadas se pueden ir añadiendo una a una conforme se constata su necesidad.

Las aplicaciones de Odoo proporcionan una funcionalidad central alrededor de la cual otros módulos (a veces llamados addons o complementos) agregan o modifican características. Se podría decir que en Odoo las aplicaciones son bloques de módulos. Cada ficha que aparece en la pantalla de *Aplicaciones* proporciona información relevante acerca de la misma. Hecha esta aclaración, en este texto se utilizarán de ahora en adelante indistintamente ambas opciones.

Es importante tener en cuenta que los módulos ofrecidos por Odoo suelen ser de fácil instalación, pues si el módulo ya tiene su carpeta descargada en la carpeta *addons* nombrada antes, generalmente basta con pinchar el botón *Instalar* que aparece en su ficha.

Pero también es cierto que existen unas dependencias intermodulares que pueden complicar el proceso. En esas circunstancias el módulo que se está intentando instalar, si no las resolviera automáticamente, avisará de las dependencias que tiene y que deben ser instaladas previamente.

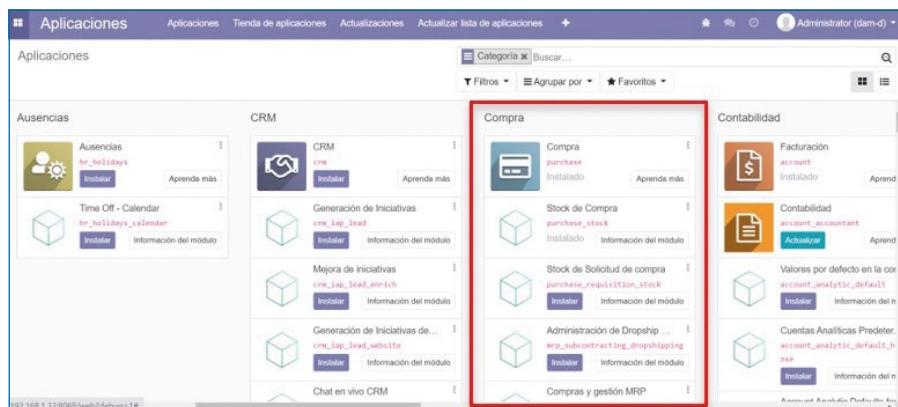


Figura 3.21
Aplicación Principal y Módulos
(extensiones de la aplicación).

Aplicaciones						
	Nombre de módulo	Nombre técnico	Autor	Sitio web	Última versión	Estado
CRM	crm	Odoo S.A.	https://www.odoo.com/page/crm		13.0.1.0	Instalado
Sitio web	website	Odoo S.A.	https://www.odoo.com/page/website-builder		13.0.1.0	No instalado
Proyecto	project	Odoo S.A.	https://www.odoo.com/page/project-management		13.0.1.1	No instalado
Partes de horas	timesheet_grid	Odoo S.A.	https://www.odoo.com/page/timesheet-mobile-app?utm_source=odoo&utm_medium=module		13.0.1.0	Desinstalable
Inventario	stock	Odoo S.A.	https://www.odoo.com/page/warehouse		13.0.1.1	A instalar
Código de barras	stock_barcode	Odoo S.A.	https://www.odoo.com/page/warehouse?utm_source=odoo&utm_medium=module		13.0.1.0	Desinstalable
Facturación	account	Odoo S.A.	https://www.odoo.com/page/billing		13.0.1.1	Instalado
Contabilidad	account_accountant	Odoo S.A.	https://www.odoo.com/page/accounting?utm_source=odoo&utm_medium=module		13.0.1.0	Desinstalable
Fabricación	mrp	Odoo S.A.	https://www.odoo.com/page/manufacturing		13.0.2.0	No instalado

Figura 3.22
Relación de módulos instalados.

En la pantalla de Aplicaciones, el botón  proporciona información más detallada acerca de los módulos instalados, sus versiones y si son desinstalables o no, entre otras cosas.

Además, hay una gran cantidad de módulos que no se descargan y preconfiguran en la instalación inicial. Para instalarlos habrá que seguir un procedimiento particular para cada módulo.



Figura 3.23
Obtención
de información
general
y dependencias
de un módulo de Odoo.

Supuestos prácticos

1. Para este ejercicio se debe crear una nueva base de datos de Odoo ERP en la modalidad SaaS.

Para ello, hay que acceder a la página www.odoo.com y dar de alta una cuenta de usuario. A continuación, crear una base de datos nueva con un único módulo inicial: CRM.

Recuerda que el nombre de la BBDD debe comenzar por “edu-” para que la instancia proporcionada en modo trial por Odoo tenga una duración de 10 meses.

- a) Configura una compañía (nombre, dirección, teléfono, CIF...).
- b) Cambia la imagen de la compañía mediante la introducción de un logo.
- c) Crea un usuario con una cuenta de Gmail (o similar) distinta e invítalo a conectar.
- d) Establece una conversación directa entre el administrador y dicho usuario con el intercambio de varios mensajes.
- e) Crea un nuevo canal llamado ies-canal.
- f) Crea una nueva empresa (completando todos los datos).
- g) Crea un nuevo contacto que pertenezca a la empresa anteriormente creada.
- h) Instala los módulos Ventas, Compra, Almacén y TPV.

2. Para el desarrollo de este supuesto y los de los siguientes capítulos, se necesitará utilizar máquinas virtuales. Para este en particular, una máquina virtual con sistema operativo Ubuntu Desktop (recomendado: 2 núcleos, 2 GB de RAM y 25 GB de disco). Instala Odoo Community siguiendo el procedimiento sugerido en la página web de odoo.com para una instalación con paquete .deb. El resultado final debe ser un manual de instalación que cualquier técnico con limitados conocimientos técnicos pueda utilizar.

- a) Crea la máquina virtual con SO Ubuntu Desktop.
- b) Descarga el paquete .deb de Odoo.
- c) Sigue los pasos de instalación recomendados en: [https://www.odoo.com/documentation/\[versión\]/setup/install.html](https://www.odoo.com/documentation/[versión]/setup/install.html).
- d) Documenta todo el proceso que se ha seguido desde el punto 2, explicando lo realizado con la ayuda de capturas de pantalla.
- e) Explica los problemas que se han tenido que solventar.

Carga de módulos:

- f) Instala al menos los módulos CRM, Proyectos y Empleados. Comprueba las posibles dependencias.
- g) Consigue una evidencia de la instalación y añádela al manual de instalación.
- h) Realiza una conexión desde el host (el equipo anfitrión) con Windows o Linux al equipo Ubuntu Desktop y comprueba el funcionamiento correcto del servidor Odoo instalado.

3. En este ejercicio se va a practicar la gestión de PostgreSQL con *psql*. *psql* permite gestionar PostgreSQL desde el terminal. Se puede escribir consultas de manera interactiva y ver los datos resultantes. Dispone de un *Shell* que permite la escritura de scripts.
 - a) Comprueba los usuarios que se han creado en el apartado 3.3.5 del libro y sus características. ¿Son usuarios del sistema?
 - b) Prueba los principales comandos de *psql*.
 - c) Crea una tabla cualquiera con 3 campos e índices necesarios.
 - d) Usando los scripts necesarios, gestiona la información para introducir datos, actualizarlos, borrarlos y visualizarlos en pantalla.
4. Se van a instalar módulos de la contabilidad española en la máquina del supuesto práctico 2. En esta ocasión serán módulos que no incluye la instalación que se ha hecho previamente de Odoo.

Comprueba que el módulo de Contabilidad es de pago para la versión SaaS. Una vez configurada la información acerca de la compañía, se debe acceder a la OCA en GitHub - OCA/l10n-spain: Odoo Spain Localization y estudiar su contenido.

En realidad, para este ejercicio basta con informarse de lo que añade y que se instale:

 - Adaptación de los clientes proveedores y bancos para España (*l10n_es_partner*).
 - Topónimos españoles (*l10n_es_toponyms*).

Deben comprobarse los requerimientos para la instalación de estos módulos y sus dependencias por si hay que instalar previamente otros módulos, lo que suele ser práctica habitual en la instalación de módulos y aplicaciones de terceros.
5. Se va a practicar la descarga y la instalación *pgAdmin4*. *pgAdmin4* es una aplicación web, *Open Source*, escrita en Python que permite la gestión gráfica de BBDD PostgreSQL.
 - a) Configura PostgreSQL de la instancia de Odoo del ejercicio 3.2 para poder acceder remotamente (por ejemplo, desde el host).
 - b) Una vez instalado *pgAdmin4*, crea una conexión a una base de datos de Odoo.
 - c) Realiza una consulta que muestre los siguientes datos incluidos en la tabla *res_partner*:
 - Nombre: nombre de la empresa.
 - Dirección: calle.
 - C.P.: código postal.
 - Localidad: ciudad.
 - Teléfono: teléfono de la empresa.
 - Límite: crédito concedido.

Resumen

- Existen unos requisitos previos para la instalación de un ERP.
- Se ha optado por el estudio práctico con la solución *Open Source* y comercial (SaaS) de Odoo en un entorno virtual.
- El proceso de instalación, dependiendo del entorno, es diferente.
- Odoo implementa el Modelo Vista Controlador (MVC).
- Odoo ERP es una de las soluciones *Open Source* más conocidas y utilizadas. Dispone de una amplia comunidad de desarrolladores que comparte el código fuente de las aplicaciones desarrolladas por sus miembros.
- El código de Odoo está escrito en Python y la BBDD que utiliza por defecto es PostgreSQL.
- Utiliza HTML5 y JavaScript para la construcción de vistas.
- Las instalaciones monousuario están prácticamente desaparecidas.
- Las instalaciones SaaS suelen ser licencias comerciales, aunque hay algunas de uso libre y gratuito, limitadas en cuanto a módulos disponibles, usuarios y/o tamaño de la BBDD. El proveedor, si no factura por el software porque sea *Open Source*, sí que puede facturar por los servicios de aprovisionamiento de máquinas, instancias de la aplicación, disponibilidad y almacenamiento *Cloud*.
- La instalación mediante paquetes es simple en entorno Windows y algo más complicada en entorno Linux.
- La instalación manual de Odoo permite disponer del código fuente y, por tanto, abordar acciones de modificación de componentes y creación de módulos.
- En Odoo, para las labores de configuración es indispensable trabajar en “Modo desarrollador”.
- La carpeta addons contiene todas las aplicaciones *Open Source* de Odoo.
- Respecto a las funcionalidades que ofrece, se puede establecer una clasificación general de aplicaciones en cuatro categorías:
 - Módulos para gestionar los procesos de fabricación si los hubiera.
 - Módulos para gestionar los procesos financieros y contables.
 - Módulos para gestionar los procesos de marketing y ventas.
 - Módulos para gestionar los procesos de RRHH.
- Algunos módulos de Odoo, como los de otras soluciones ERP, sobrepasan los límites de estas categorías implementando funcionalidades de varias de ellas de forma integrada.

ACTIVIDADES DE AUTOEVALUACIÓN

1. Indica las tecnologías utilizadas por Odoo (señala todas las correctas):

- a) Python.
- b) C#.
- c) JavaScript.
- d) Java.

2. Los tipos de instalación que ofrece Odoo son:

- a) SaaS, Packaged Installers, Código fuente, Docker.
- b) SaaS, Docker, Código fuente.
- c) Packaged Installers, Código fuente, SaaS.
- d) Windows, Debian, Fedora.

3. La activación del modo desarrollador (señala todas las correctas):

- a) Permite depurar código.
- b) Proporciona menús de configuración diferentes.
- c) Activa las Herramientas de Desarrollo.
- d) Proporciona permisos de superusuario.

4. Respecto a PostgreSQL señala la respuesta incorrecta:

- a) El usuario postgres creado durante el proceso de instalación tiene contraseña vacía.
- b) Debe instalarse antes de instalar Odoo.
- c) El prompt de psql es =#
- d) Escucha por el puerto 5432 por defecto.

5. De acuerdo con la instalación realizada en este capítulo, la carpeta addons del servidor en Ubuntu se encuentra en:

- a) /home/odoo
- b) /etc/bin/odoo
- c) /opt/odoo
- d) /opt/odoo/odoo

6. Para cambiar el puerto de acceso a la BBDD de PostgreSQL, se debe modificar el fichero:

- a) odoo.conf
- b) odoo.service
- c) pg_hba.conf
- d) odoo-bin

7. La conexión a la BBDD PostgreSQL se puede hacer mediante pgAdminIV (señala todas las correctas):

- a) Desde localhost.
- b) Desde cualquier máquina de la red.
- c) Desde cualquier máquina de la red modificando postgresql.conf.
- d) Desde cualquier máquina de la red modificando pg_hba.conf y postgresql.conf.

8. En Ubuntu, los mensajes del servidor se van grabando en:

- a) /home/user
- b) /etc/bin
- c) /var/log
- d) /bin

9. El ejecutable de Odoo es:

- a) odoo.conf
- b) odoo.service
- c) odoo-bin
- d) odoo.exe

10. ¿En qué lenguaje de programación se codifican los módulos en Odoo?:

- a) Python
- b) Java
- c) C++
- d) C#

SOLUCIONES:

1. **a** **b** **c** **d**

2. **a** **b** **c** **d**

3. **a** **b** **c** **d**

4. **a** **b** **c** **d**

5. **a** **b** **c** **d**

6. **a** **b** **c** **d**

7. **a** **b** **c** **d**

8. **a** **b** **c** **d**

9. **a** **b** **c** **d**

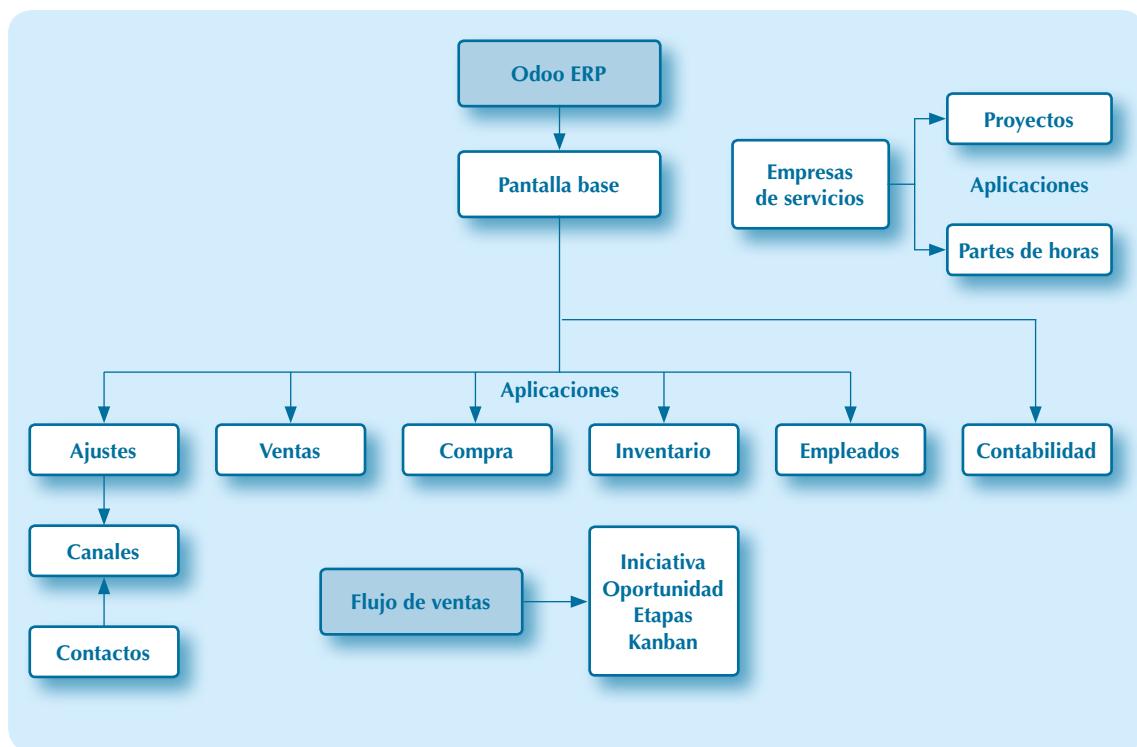
10. **a** **b** **c** **d**

Configuración inicial y funcionamiento de Odoo ERP

Objetivos

- ✓ Realizar los primeros pasos de configuración en una instalación de Odoo ERP.
- ✓ Obtener una visión general del cliente web (partes de la pantalla).
- ✓ Entender la importancia de proporcionar los datos básicos iniciales.
- ✓ Conocer de forma básica la aplicación Contactos.
- ✓ Planificar actividades con la aplicación CRM.
- ✓ Instalar y revisar las aplicaciones más habituales.
- ✓ Entender los conceptos de *iniciativa, oportunidad y cierre de negocio*.
- ✓ Conocer las particularidades de la gestión de las empresas de servicios.
- ✓ Instalar y conocer el funcionamiento básico de las aplicaciones más habituales en empresas de servicios.

Mapa conceptual



Glosario

Accounting. Contabilidad.

Cash Flow. Flujo de caja.

Kanban. Palabra japonesa que define un sistema visual de información que clasifica en estados de situación las tareas en las que se descompone un proyecto.

Packaged Installers. Paquete de instalación asistida que pone el fabricante a disposición del cliente para que pueda realizar la instalación del producto.

Pipeline. Canalización. Flujo de ventas.

PO - Purchase Order. Orden de compra.

RFQ - Request for Quotation. Solicitud de presupuesto.

Widescreen. Monitor de pantalla ancha. Su relación de aspecto es 16:9 en vez de el 4:3 tradicional.

4.1. Introducción

Una vez instalada, en cualquier entorno, la instancia del servidor de Odoo 13 y creada la base de datos, es conveniente realizar una primera parametrización con los datos básicos de la compañía antes de empezar a utilizarla.

Para esta parte se utilizará la versión *Cloud* (que como ya se ha comentado, es una versión *Enterprise*) por lo que algunas pantallas pueden diferir de las que aparezcan usando la versión *Community*.

4.2. Configuración inicial

La pantalla principal de la aplicación tendrá un aspecto muy parecido a este:



Figura 4.1
Pantalla principal
de Odoo.

En ella se ubicarán los iconos de las aplicaciones que se hayan escogido para su instalación inicialmente.

En la parte superior derecha se disponen unos iconos que permitirán, por ejemplo, conocer las actividades realizadas últimamente, las previstas para realizar en el día de hoy y siguientes clasificadas por aplicación, las conversaciones abiertas en los distintos canales de comunicación que ofrece Odoo, así como información de la cuenta que ha accedido (con opciones de ayuda, documentación, preferencias).

Conviene dedicar unos minutos a navegar para entender el diseño del ERP y sus módulos y ver las estructuras de las pantallas y la disposición de los distintos menús.

Cada ícono de la zona central abre una aplicación que dispondrá normalmente de un menú horizontal entre cuyas opciones estará la opción de ajustes o configuración propia, a la que también se puede acceder como parte de la aplicación *Ajustes* existente en la pantalla principal.



Figura 4.2
Menú horizontal principal.

A lo largo del capítulo se profundizará en estos y otros aspectos para dar una visión global de la aplicación desde el punto de vista del usuario.

4.2.1. Datos básicos de la compañía

Los datos básicos, como el nombre, dirección, formas de contacto, logo, CIF, cuentas bancarias, se pueden configurar en la primera pantalla de la opción de *Ajustes*.

Es importante proporcionar estos datos antes de emprender la primera actividad porque, entre otras cosas, muchas de las tablas que se vayan creando con esas acciones serán denominadas internamente incluyendo en su nombre referencias a ese nombre de compañía.

También el idioma y la localización influirán en aspectos como el formato de fechas (parametrización regional) y la aplicación de contabilidad.

Además, estos datos se integrarán en los modelos de documentos de negocio que se utilizarán posteriormente, en cuerpos, cabeceras y pies de página.

Esta pantalla tiene los apartados siguientes:

- Usuarios (2).
- Compañías.
- Idiomas.
- Documentos de negocio.
- Estadísticas.
- Contactos.
- Conversaciones.
- Compras dentro de la aplicación.
- Productos.
- *Odoocloud Notification* (OCN).
- Integraciones.
- Vista del mapa.
- Herramientas de desarrollo.
- Acerca de.

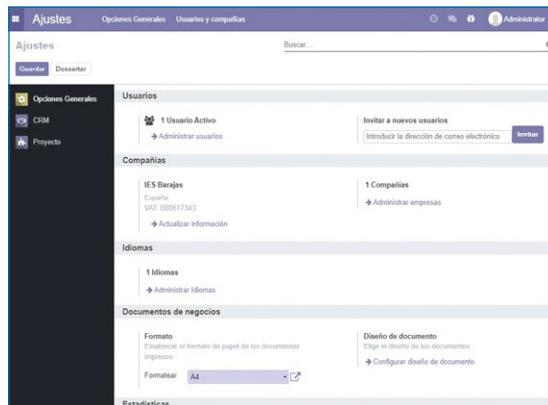


Figura 4.3
Ajustes generales.

Actividad propuesta 4.1



- Explica en qué consiste el modo desarrollador.
- Averigua las diferencias entre el modo desarrollador y el modo desarrollador con activos.

4.2.2. La comunicación interna

El módulo *Conversaciones* es una potente herramienta de comunicación interna y externa, que permite a los usuarios entablar conversaciones a través de chat con colegas y grupos para el seguimiento del trabajo compartido de forma cómoda en un solo lugar.

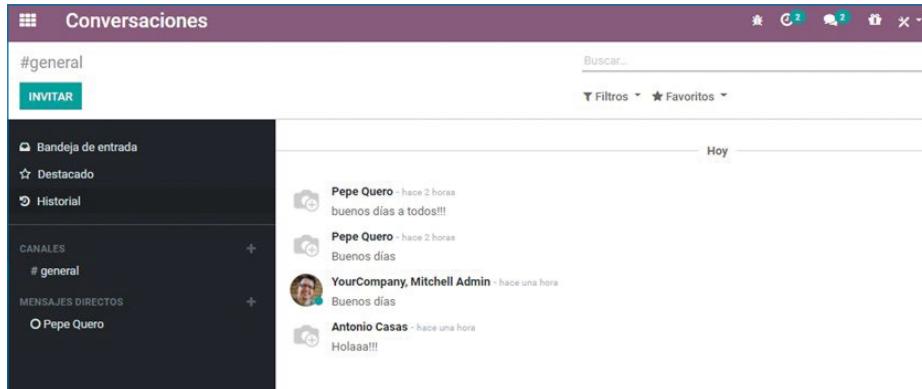


Figura 4.4
Conversaciones.

Además, al visualizar un documento de negocio como, por ejemplo, un presupuesto, en la parte inferior de la pantalla (dependiendo de la configuración de la pantalla –en pantallas *widescreen*–, podría aparecer en el lado derecho) aparece una sección con las conversaciones, comentarios y peticiones que el personal se ha intercambiado durante su preparación, lo que constituye un historial de la operación.

Al inicio se crean dos categorías: Canales y Mensajes directos.

Producto	Descripción	Cantidad	Precio unitario	Impuestos	Total
Virtual Interior Design	Virtual Interior Design	24.000	75,00	IVA 21% (Base)	1.800,00 €
Virtual Home Staging	Virtual Home Staging	30.000	38,25	IVA 21% (Base)	1.147,50 €

Base Imponible: 2.947,50 €
Impuestos: 631,99 €
Total: 3.566,48 €

Figura 4.5
Vista en pantalla *widescreen*.



Actividad propuesta 4.2

- Estudia las comunicaciones que se pueden establecer mediante Conversaciones y Canales.
- Crea un canal y prueba su utilización.

4.2.3. Contactos e importación de datos

La aplicación *Contactos* permite centralizar toda la información relativa a los distintos interlocutores de la empresa: clientes, compañeros, proveedores, etc., clasificándolos en dos grandes categorías: Compañías y Particulares.

En versiones anteriores, la pestaña de *Venta* y *compra* permitía también la catalogación como cliente o proveedor de cualquier contacto. En Odoo 13.0 y posteriores se han eliminado los campos “Es un cliente” y “Es un proveedor” en el formulario de contacto. En su lugar, hay dos campos nuevos: “Rango de cliente” y “Rango de proveedor”, pero no se muestran.

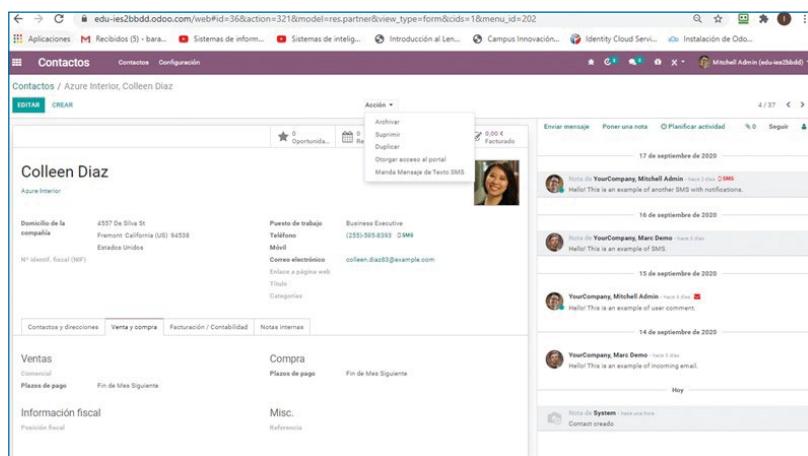


Figura 4.6
Vista de contacto.

Si el contacto se añade desde el módulo de Ventas, se catalogará como cliente; y si se hace desde el módulo de Compra, se guardará como proveedor. Esto es así porque cualquier entidad relacionada de alguna manera con el sistema, ya sea personal de la empresa, cliente o proveedor, se almacena en el mismo modelo (tabla de base de datos).

Hay que tener en cuenta que en el mundo comercial actual es cada vez más frecuente que un contacto pueda ser a la vez proveedor y cliente.

Puede utilizar la función de importación para preestablecer estos campos (Rango de cliente, Rango de proveedor).

Con *Contactos* se puede codificar, gestionar e importar la información de contactos que estarán disponibles para el resto de las aplicaciones instaladas.

Odoo permite importar contactos (y cualquier tipo de elemento comercial como productos, pedidos, movimientos bancarios, etc.) desde ficheros .csv y .xls.

Dispone de plantillas que se pueden adaptar. Además, en caso de que no pueda mapear una columna de forma automática porque no se ajuste a ninguno de los campos que tiene establecidos por defecto Odoo, se podrá asignar de forma manual escogiendo el campo apropiado de la lista.

RECUERDA

- ✓ Se pueden importar tablas de bases de datos SQL, teniendo en cuenta que, si se importan datos de diferentes tablas, habrá que recrear las relaciones entre ellas de forma manual.

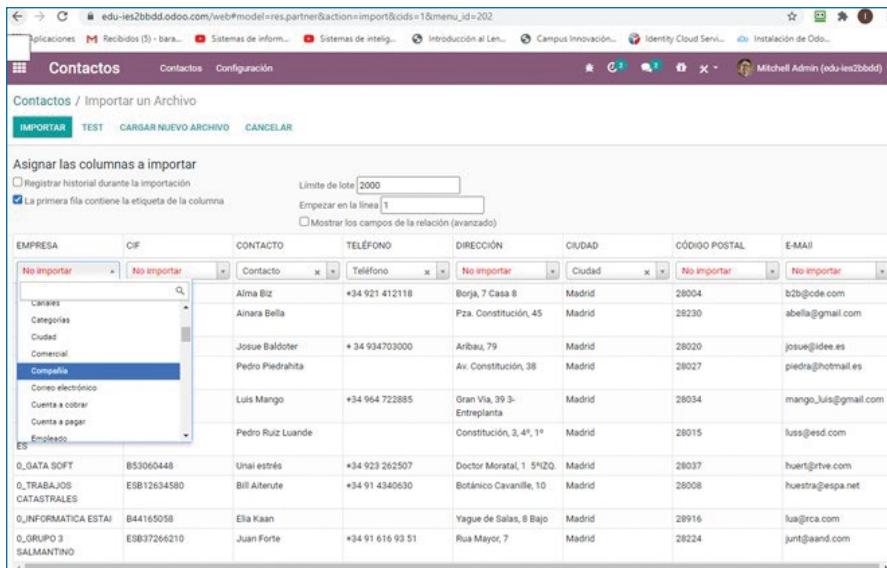


Figura 4.7
Importación de datos.

En la página web de Odoo se pueden encontrar modelos de archivos *.csv* con la estructura que utiliza la aplicación en los objetos categorías, productos, fabricantes, clientes, órdenes de compra, etc.

4.2.4. Planificando actividades (CRM)

Con la aplicación CRM se pueden organizar las actividades y tareas diarias y aplicarlas a los distintos proyectos u operaciones en los que se participa.

Esta aplicación proporciona en su pantalla de inicio una vista Kanban de todas estas operaciones si está activado el filtro “Flujo de ventas”.

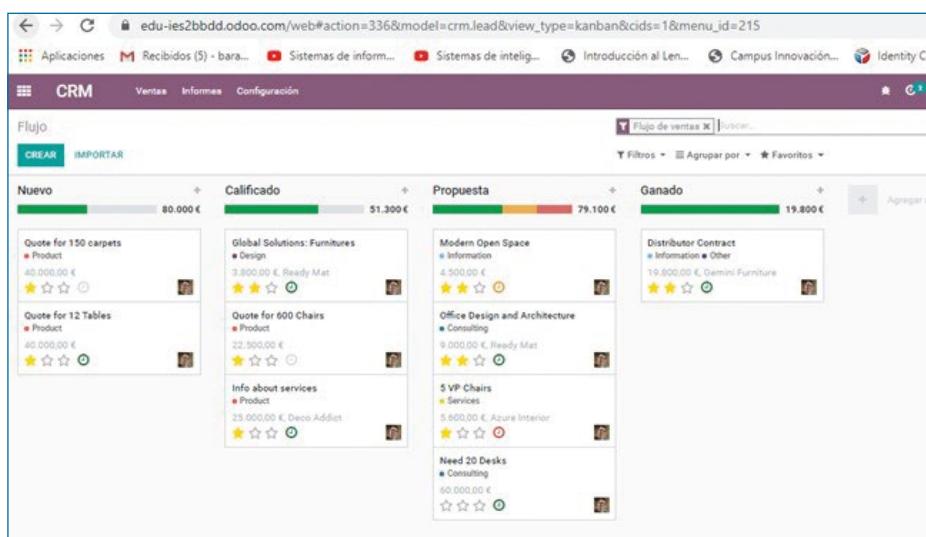


Figura 4.8
Pipeline.

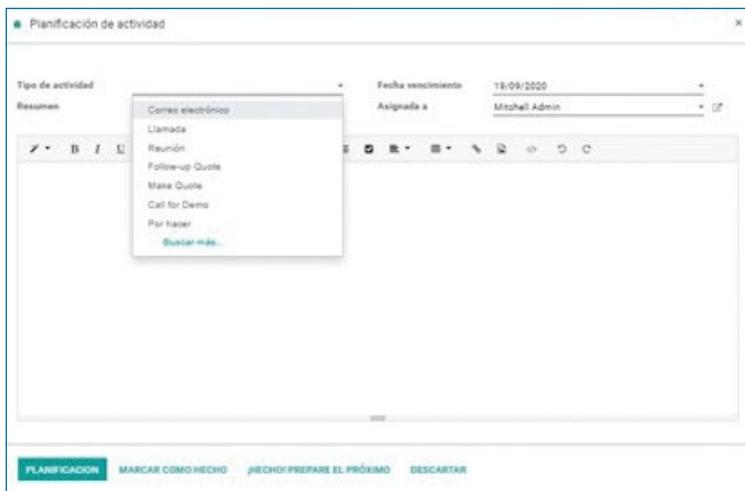


Figura 4.9
Planificación de actividad.

Desde aquí se puede entrar en cualquiera de ellas y añadir una actividad planificada con la opción “Planificar actividad”. Entre las opciones que aparecen, se puede agendar y asignarla a cualquiera de los colegas involucrados en la operación. Una nota nueva se añadirá al historial de la operación como recordatorio de la fecha prevista para llevar a cabo la operación planificada.

Además de las actividades que vienen por defecto, se pueden crear desde la opción correspondiente del menú de CRM.

La creación de nuevas actividades permite configurarlas para que salten automáticamente después de otra actividad en un plazo fijado o para que realicen otra actividad después de su finalización, entre otras cosas, lo que permite automatizar, de acuerdo con los procedimientos internos practicados, el seguimiento de cualquier operación comercial.

4.3. Principales aplicaciones de Odoo

Como ya se ha comentado, las aplicaciones proporcionan determinadas funcionalidades relativas a un tipo de proceso empresarial específico, mientras que los módulos añaden características a esas aplicaciones principales.

Junto al módulo base, que incorpora las funcionalidades básicas troncales del ERP, en la instalación inicial de Odoo se instalan parte de funcionalidades de comunicación interna, calendario, ajustes y contactos.

En este apartado se describirán algunas de las aplicaciones generalmente consideradas más importantes, como *Ventas*, *Compra*, *Inventario*, *Empleados* o *Contabilidad*. Pero esta no constituye una relación exhaustiva, pues seguramente otras como *Facturación*, *Punto de venta*, *Calidad*, *Ausencias*, *Nóminas* o *Comercio electrónico* tienen también su importancia para casi cualquier organización.

No obstante, para cada empresa, la relación de aplicaciones importantes será distinta. Una empresa fabricante necesitará aplicaciones como *Fabricación*, *Inventario* o *Gestión del ciclo de vida*, mientras que para una empresa de servicios profesionales las aplicaciones de *Proyecto*, *Suscripciones* o *Partes de horas* serán, seguramente, imprescindibles.

4.3.1. Aplicación Ventas

La venta es la funcionalidad más importante de una empresa. El éxito de una empresa radica en la capacidad de su departamento comercial para hacer que el cliente quiera el producto que ofrece frente al de otros competidores.

La aplicación *Ventas* automatiza parte de las tareas que realiza la fuerza de ventas. La elaboración de presupuestos a clientes puede ser considerada la actividad principal del departamento. Posteriormente, su seguimiento por parte de los comerciales y el estudio de los resultados obtenidos durante un periodo determinado y los informes correspondientes, serán las acciones relacionadas con este módulo.

Permite elaborar presupuestos para las oportunidades detectadas y hacer un seguimiento de todo el proceso, así como realizar variaciones sobre productos, cupones, órdenes de venta y contratos, entre otros. Gestiona las listas de precios y los descuentos. Proporciona plantillas personalizables para elaborar presupuestos y presenta una integración total con el módulo de CRM. Dispone de una gran variedad de informes prediseñados basados en KPI y soporta la firma electrónica.

Está integrado con otros módulos de Odoo como *CRM*, *Firma electrónica*, *Sitios web* y *Contabilidad*.

Para elaborar un nuevo presupuesto para un cliente basta con pulsar la opción *Crear*. A partir de este formulario se podrá:

- Añadir un cliente o utilizar los datos de un cliente ya existente en la BBDD.
- Añadir los productos definidos en la aplicación *Compra*.
- Modificar los precios manualmente o por medio de fórmulas matemáticas introducidas directamente en el campo de precio que la aplicación calculará automáticamente.
- Añadir secciones al documento para separar grupos de productos y comentarios.

Líneas del pedido	Productos opcionales	Otra Información	Firma del Cliente			
Producto	Descripción	Cantidad	Precio unitario	Impuestos	Subtotal	
+ Muebles						
+ [DESK0004] Customizable Desk (CONFIG) (Aluminum, Black)	1,000	756,00	(IVA 21% (Bienes))	756,00	756,00	
+ [DESK0005] Customizable Desk (CONFIG) (Custom, White)	1,000	750,00	(IVA 21% (Bienes))	750,00	750,00	
+ [E-COM01] Storage Box	[E-COM01] Storage Box	1,000	79,00	(IVA 21% (Bienes))	79,00	79,00
+ Servicios						
+ Virtual Home Staging	Virtual Home Staging	1,000	38,25	(IVA 21% (Servicios))	38,25	38,25

Figura 4.10

Diseño de presupuesto con secciones.

Productos	Cantidad	Precio unitario	Impuestos	Importe
Muebles				
[DESK0004] Customizable Desk (CONFIG) (Aluminum, Black)	1,000	756,00	(IVA 21% (Bienes))	756,00
[DESK0005] Customizable Desk (CONFIG) (Custom, White)	1,000	750,00	(IVA 21% (Bienes))	750,00
[E-COM01] Storage Box	1,000	79,00	(IVA 21% (Bienes))	79,00
Total				1.585,00 €
Services				
Virtual Home Staging	1,000	38,25	(IVA 21% (Servicios))	38,25
Total				38,25 €

Figura 4.11

Formato de envío.

Una vez elaborado, se podrá imprimir en formato pdf o enviar por correo electrónico directamente. En este caso se puede obtener una previsualización para comprobar cómo lo verá el cliente.

Al confirmar el envío, Odoo dará la oportunidad de modificar el cuerpo del mensaje que tiene por defecto configurado en la plantilla.



PARA SABER MÁS

Se puede agregar un tiempo de seguridad en la pestaña de *Inventario del Producto* para añadir tiempo necesario desde que se recibe un producto del proveedor hasta que está disponible para entregar al cliente.

4.3.2. Aplicación Facturación

Permite la generación automática de facturas considerando el estado de órdenes de venta y de entrega y su envío electrónico a los clientes. También factura las horas de trabajo, contratos y permite un seguimiento y los pagos *online*.

Proporciona informes personalizables mediante *dashboards* con tablas dinámicas.

La aplicación está integrada con *Ventas, Compra, Proyectos, Contabilidad*.

4.3.3. Aplicación Compra

Diseñada para la gestión de compras. Automatiza el proceso de compra a proveedores desde la petición de ofertas, la recepción y comparativa de ofertas recibidas y la generación de las órdenes de compra, hasta la recepción de la factura y la confirmación de la recepción del material adquirido.

Permite automatizar las tareas de adquisición de productos y servicios necesarias en todas las empresas, establecer una política de compras adecuada y controlar las existencias en el almacén o almacenes utilizados.

Incluye también procesos relacionados con producto y almacén como la automatización de los procesos de compra en función del *stock* y, como todos los módulos, sus propios informes, como estadísticas de compra que miden comparativamente a proveedores mediante listas de precios, descuentos, retrasos en la entrega.

La aplicación está integrada con otras como *Inventario, Facturación y Contabilidad*.

Permite automatizar procesos, dar acceso a los proveedores para que puedan obtener directamente pedidos, hacer entregas, etc.

Con esta aplicación se gestionan las solicitudes de presupuesto (RFQ) y las órdenes de compra (PO).

A partir de un formulario se podrá:

- Añadir un proveedor o utilizar los datos de uno ya existente en la BBDD.
- Añadir los productos del catálogo de este proveedor.
- Modificar los precios manualmente o por medio de fórmulas matemáticas introducidas directamente en el campo de precio que la aplicación calculará automáticamente.
- Especificar los términos y condiciones.

En la parte superior derecha se muestra el estado actual de la solicitud.

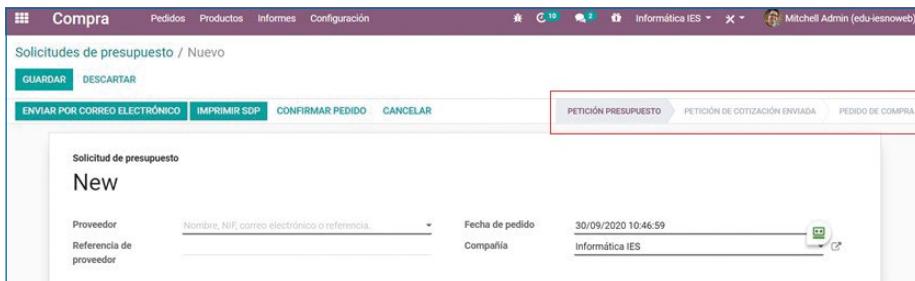


Figura 4.12
Solicitud de oferta.

La aplicación permite establecer un sistema de autorizaciones para, por ejemplo, que un responsable deba aprobar pedidos realizados por el departamento por valor superior a una cierta cantidad, desde la opción de *Configuración/Ajustes* de la propia aplicación.

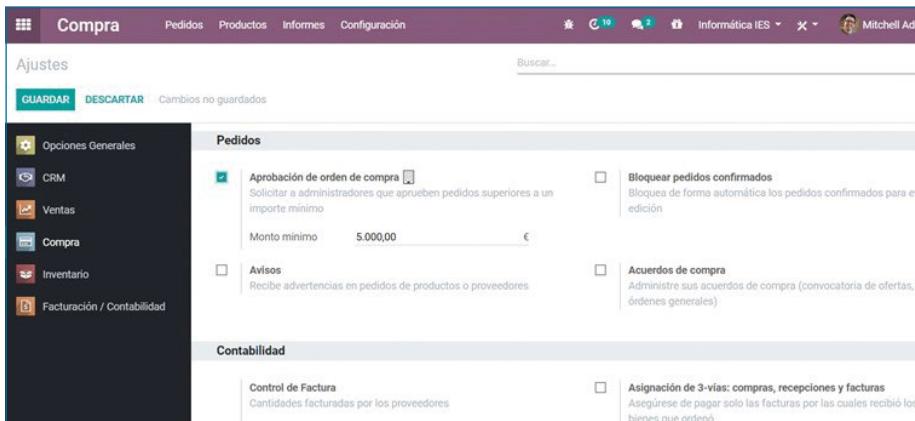


Figura 4.13
Ajustes Compra.

Aparecerá una nueva fase en la zona de “estado actual” antes mencionada: *Para aprobar*, y al administrador le aparecerá la opción de *Aprobar orden*. Cuando la operación se ha concretado y el material se ha recibido, se puede marcar como “hecha” ya sea parcialmente, si se ha recibido menos cantidad de la inicialmente pedida, o de forma total.

En el primer caso se puede generar automáticamente una nueva orden de pedido pendiente con la cantidad restante.

También se pueden configurar los plazos de entrega en el formulario del producto.

4.3.4. Aplicación Inventario

Los productos almacenables deben estar controlados para evitar tanto la acumulación excesiva como la posible falta en un momento determinado. El control del *stock* es imprescindible en cualquier empresa.

Permite el inventario de doble entrada, la solicitud automatizada de presupuestos, la trazabilidad de las compras, etc.

Además, proporciona informes en tiempo real y permite el uso de *dashboards*.

Está integrada con las aplicaciones *Ventas*, *Compra* y *Contabilidad*.

La gestión del inventario ayudará a prevenir la escasez de un producto y a mantener una adecuada cantidad de este.

Odoo permite, con su módulo de *Inventario*, gestionar los pedidos recibidos (recepciones) o en proceso de recepción y las entregas realizadas (órdenes de entrega) o en proceso de entrega. En definitiva, proporciona una visión general de las operaciones en curso.



Figura 4.14

Visión operaciones de compra.

Se trata de un módulo intrínsecamente relacionado con los módulos de *Ventas* y *Compra*.

Al escoger un producto durante la elaboración de un presupuesto con el módulo de *Ventas*, en el campo *Cantidad* se dispondrá de un aviso respecto al *stock* actual del producto presupuestado. Así, si fuera necesario, cambiando a la aplicación *Compra* se podría generar un pedido al proveedor para aprovisionar el material necesario para cumplir con una petición confirmada por el cliente.



TOMA NOTA

Esta aplicación permite gestionar todo el ciclo de una operación de compra-venta.

Mediante los ajustes manuales de inventario se puede hacer coincidir el inventario con las cantidades reales disponibles (considerando las mermas por robo, errores cometidos, deterioros) en la opción de *Actualizar la cantidad*.



Figura 4.15
Ajuste manual
de inventario.

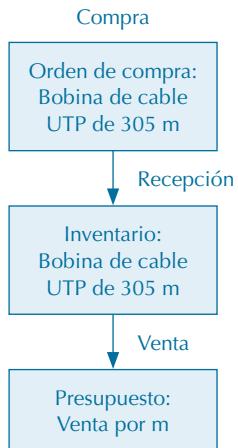
INVENTARIO					
Actualizar la cantidad		Ubicaciones Internas > Producto > Ubicación			
INVENTARIO A FECHA		Lote/Nº de serie	Cantidad disponible	Valor	Compañía
» [DESK0004] Customizable Desk (CONFIG) (Aluminium, Black) (1)			60,00	30.000,00	
» [DESK0005] Customizable Desk (CONFIG) (Custom, White) (1)			65,00	0,00	
» [DESK0006] Customizable Desk (CONFIG) (Custom, Black) (1)			70,00	0,00	
» [E-COM07] Large Cabinet (2)			350,00	280.000,00	
» WH/Stock (1)			-150,00	-120.000,00	
» [E-COM07] Large Cabinet	WH/Stock		-150,00	-120.000,00 €	Informática IES
Aregar linea					
» WH/Stock/Shelf 1 (1)			500,00	400.000,00	
» [E-COM07] Large Cabinet	WH/Stock/Shelf 1		500,00	400.000,00 €	Informática IES
Aregar linea					
» [E-COM08] Storage Box (1)			18,00	1.260,00	
» [E-COM10] Pedal Bin (1)			22,00	220,00	

Figura 4.16
Informes de inventario.

La generación de informes de inventario se encuentra en la opción *Informes* del menú de la aplicación. Se pueden obtener informes ya prediseñados sobre el inventario, movimientos, los tiempos de entrega, retrasos, valoración y hacer análisis del almacén.

Para productos de los que se necesita controlar otras unidades de medida existe la opción de *Ajustes/Unidades de medida* que permite convertir el total de material comprado en unidades de venta diferentes.

PROCEDIMIENTO SIN UNIDADES DE MEDIDA



Implica cálculo manual tanto en el presupuesto como en la actualización de stock

UTILIZANDO UNIDADES DE MEDIDA

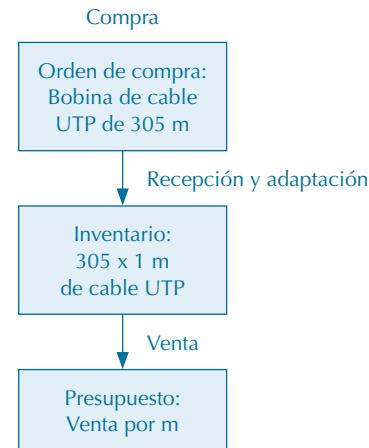


Figura 4.17
Unidades de medida.

4.3.5. Aplicación Empleados

Uno de los módulos de *Back-Office* más utilizados en las empresas es el que gestiona todo lo relacionado con el componente humano.

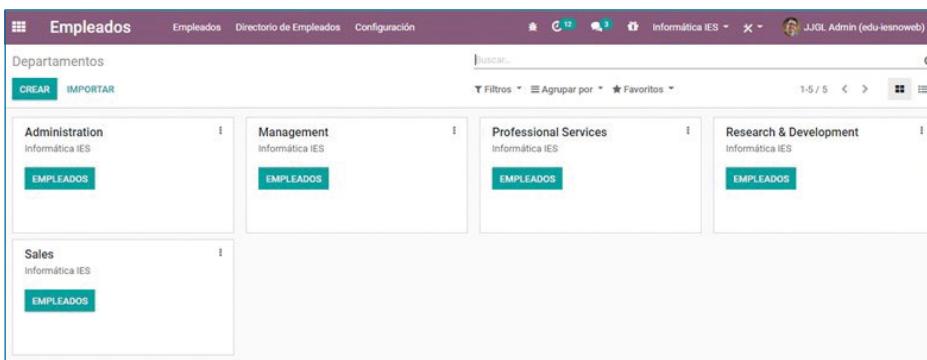


Figura 4.18
Departamentos.

La gestión de los recursos humanos de una empresa, grande o pequeña, es vital para su éxito porque la satisfacción laboral de los empleados es sinónimo de productividad.

La aplicación *Empleados* de Odoo permite gestionar en tiempo real el capital humano de un empresa: desde el proceso de contratación, la elaboración de nóminas, la gestión de ausencias legales, médicas y vacacionales hasta el control de presencia y las valoraciones del desempeño.

La empresa se divide en departamentos, accesibles desde la opción *Configuración* de la aplicación.

Una vez creados los distintos departamentos se podrá ir añadiendo los empleados mediante la opción *Empleados* del menú superior de la aplicación o desde el botón *Empleados* de la vista Kanban de *Departamentos*.

Figura 4.19
Formulario
de empleado.

El formulario proporciona campos para la introducción de una amplia gama de datos relativos al empleado, así como una vista que reproduce la estructura jerárquica (organigrama) en la que está incluido.

Permite llevar la gestión de RRHH, administrando a los empleados con funcionalidades de selección de talento y reclutamiento y evaluación de personal, control de horarios, ausencias y vacaciones, hojas de trabajo, control de gastos.

La aplicación está integrada con módulos como *Reclutamiento*, *Valoración*, *Gastos*, *Permisos*.



PARA SABER MÁS

La aplicación Empleados utiliza herramientas sociales y gamificación.

4.3.6. Aplicación Contabilidad

Controlar y registrar todo tipo de ingresos, gastos y otras operaciones económicas es fundamental para que una empresa conozca su salud financiera en tiempo real. Este sistema de control, compuesto por una serie de procedimientos estándares, se conoce generalmente como *contabilidad*.

Su objetivo es proporcionar información relevante y fiable sobre la situación económica de la empresa y su evolución durante un periodo determinado de tiempo. Además, facilita las liquidaciones de los distintos impuestos comprometidos con el Estado y una planificación avanzada de pagos e inversiones.

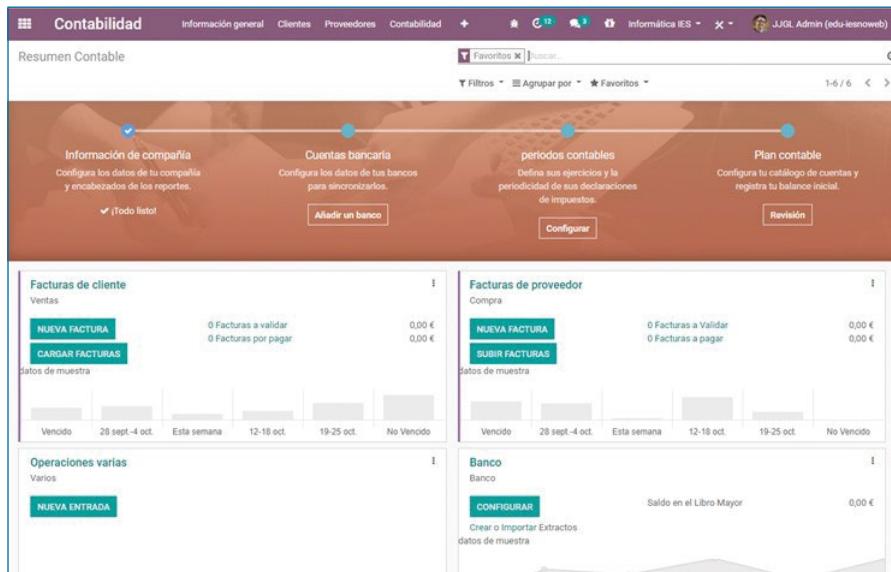


Figura 4.20
Asistente de configuración.

Determinados conceptos básicos son imprescindibles para poder manejar la aplicación *Contabilidad* de Odoo.

La forma más habitual de llevar una contabilidad es la que se conoce como *contabilidad de partida doble* en la que cada transacción afecta al menos a dos cuentas contables, moviéndose cantidades de una a otra reflejando las operaciones sucesivas que se han producido, lo que se conoce por *conciliación*.

En España, las empresas están obligadas a llevar una contabilidad del tipo de partida doble, según establece el Plan General de Contabilidad.

Todas las transacciones se registran cronológicamente en diarios. Los informes principales son el *Balance de pérdidas y ganancias*, que muestra el resultado de una empresa durante un periodo, y el *Balance general*, que es una instantánea de la situación de la empresa en un momento determinado.

Las estructuras de las cuentas, los libros, los informes y los impuestos varían de un país a otro, pero la forma de funcionar es la misma. La aplicación *Contabilidad* de Odoo está integrada con otras aplicaciones, como *Ventas*, *Compra*, *Inventario*, *Gastos*, *PoS*.

Importantísimo módulo que necesita de la previa configuración local y que:

- Gestiona las cuentas contables y realiza la conciliación mediante la integración con bancos.
- Gestiona cobros, pendientes, y gran variedad de informes como previsión de ingresos por futuros pagos acordados, ganancias, balances.
- Permite llevar la contabilidad completa y proporciona contabilidad analítica. Soporta OLAP.

Al activar el botón *Revisión* del cuarto punto el sistema, se configurará adoptando el plan contable de acuerdo con la configuración regional y el país declarado durante la instalación inicial. En esta versión, establecida la configuración regional, *Contabilidad* se instala adaptada al Plan General Contable español mediante la instalación del paquete de localización fiscal correspondiente.

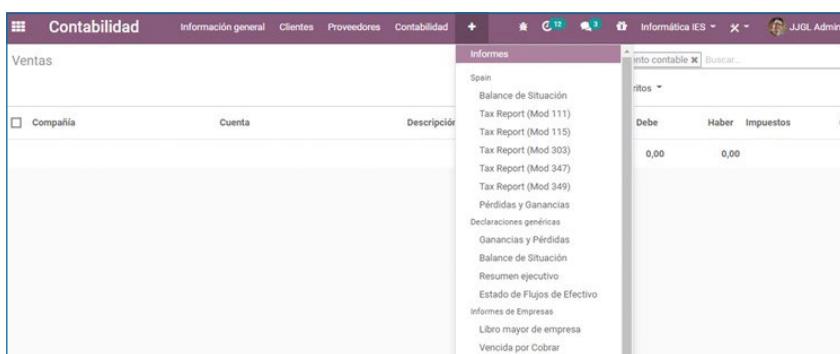


Figura 4.21
Adaptación española.

TEN EN CUENTA

- ✓ En versiones anteriores de Odoo es necesario instalar los módulos adaptados de la contabilidad española.

4.3.7. Aplicación Punto de venta (PoS o TPV)

La aplicación *Punto de venta (TPV)* proporciona, mediante el uso de un navegador en cualquier dispositivo (desde un teléfono móvil a una caja registradora) y sin necesidad de instalación, las funcionalidades que necesita una tienda de cara al público. Funciona con y sin conexión con la ubicación central de ERP, sin perder transacciones hasta la siguiente sincronización y puede trabajar con programas de fidelización. La aplicación está integrada con otras como *Ventas*, *Comercio electrónico*, *Inventario*, *Marketing* y *Contabilidad*.

4.4. Etapas del ciclo de ventas: iniciativa, oportunidad y presupuesto

A medida que una empresa crece, el control de clientes y de las operaciones de venta asociadas a él se hace más necesario.

Odoo utiliza los conceptos de *iniciativa*, *oportunidad de venta* y *flujo de ventas*.

Se puede decir que las iniciativas se corresponden con clientes potenciales (*leads* en inglés) y las oportunidades con aquellos de los que ya se dispone de todos los datos relevantes e intereses para entablar una relación comercial.

Generar una *iniciativa* es el proceso por el cual se obtienen los datos relevantes posibles de un contacto (que se han obtenido de alguna manera: por la web, llamada recibida, BBDD, etc.), con el fin de establecer una comunicación con él para intentar convertirlo en un cliente.

Las iniciativas hay que activarlas en la pantalla de *configuración/ajustes*. Su activación provocará la aparición de la opción *Iniciativas* en el menú de CRM.

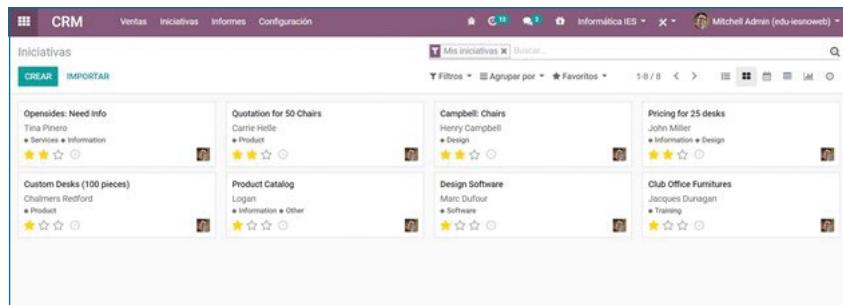


Figura 4.22
Iniciativas.

Una *oportunidad de venta* es la aparición de un momento propicio para realizar una acción encaminada a conseguir el objetivo final: una venta.

Las *oportunidades de venta* se crean desde la pantalla *Flujo* (*pipeline* en inglés) de la aplicación CRM y en su *vista formulario de venta* se configuran todas sus características.

Esta pantalla *flujo de ventas* muestra un diagrama Kanban de las *oportunidades de venta* (para que aparezcan deben tener una etiqueta de color en la opción *etiquetas*) en las que se está trabajando y la etapa en la que se encuentra cada una de ellas.

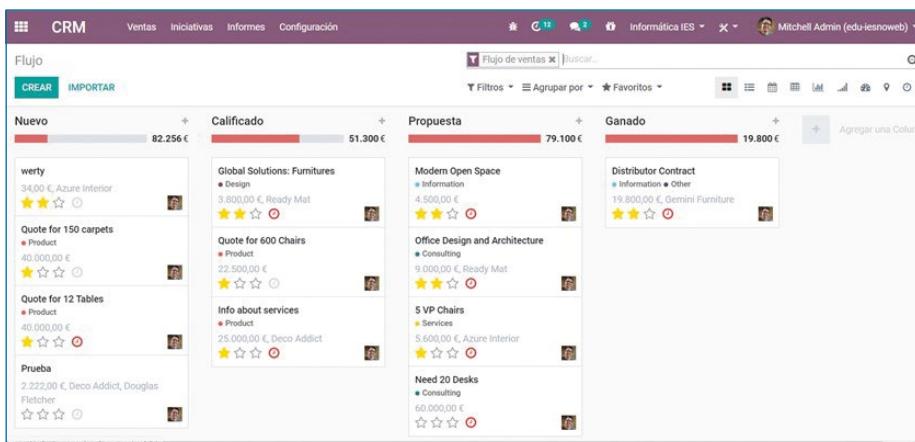


Figura 4.23
Flujo de ventas.

Odoo calcula automáticamente, en función de datos históricos, la probabilidad de ganar esa oportunidad. Los ítems que tiene en cuenta se pueden ver y modificar en *Configuración/Ajustes/Puntuación de Iniciativas pronosticadas*, aunque también el comercial encargado puede modificar este dato manualmente en función de sus sensaciones.



SABÍAS QUE...

Para una *oportunidad de venta* es necesario establecer las distintas etapas por las que va a pasar (flujo de venta) desde su aparición hasta la culminación y las acciones comerciales (*tipos de actividad*) más apropiadas a su estado que se abordarán en cada una de ellas.

Las etapas se pueden modificar y definir en la opción de *configuración/etapas*.

Las iniciativas también pueden cargarse importando un fichero .xlsx o csv.

4.5. La gestión de las empresas de servicios profesionales

Un caso que merece ser estudiado aparte es el de las empresas que proporcionan servicios. No comercializan productos tangibles, sino que su negocio consiste en la prestación de determinados servicios profesionales como por ejemplo las empresas de logística, suministros, servicios de formación, de turismo, etc.

Un servicio viene definido por una serie de características:

- *Intangibilidad*. No pueden ser tocados ni manipulados. Lo único que puede hacer el comprador antes de adquirirlos es averiguar la reputación que ha obtenido a lo largo del tiempo (fiabilidad).
- *Inseparabilidad*. Son creados y consumidos al mismo tiempo, se ofrecen *in situ* y su calidad se mide una vez prestado.
- *Heterogeneidad*. No suelen estar estandarizados; son personalizables, difícilmente repetibles y de calidad diferente según el cliente que lo valora.
- *No perdurabilidad (o carácter perecedero)*. No se pueden almacenar ni inventariar: si no se está realizando en un momento determinado, implica pérdida (no hay posibilidad de equilibrio entre una demanda variable y una oferta continuada del servicio).

A grandes rasgos, se puede distinguir entre empresas que proporcionan servicios más o menos *sostenidos* (también llamados homogéneos, porque no sufren grandes cambios de costes, personal ni de recursos materiales) como, por ejemplo, las suministradoras de gas o electricidad, o las aseguradoras; y las empresas que se basan en *proyectos*, cuyos servicios tienen duraciones, costes, y actividades diferentes como, por ejemplo, empresas de ingeniería, instaladoras y desarrolladoras de software.

En lo que se refiere a estas últimas, que suelen ser organizaciones de un alto nivel de especialización, su área de gestión crítica es la relativa a la planificación y gestión del trabajo de sus empleados y la gestión de proyectos. Es posible que no se necesite gestionar un almacén pero sí controlar fehacientemente las horas facturables de los recursos que utilice en cada proyecto o contrato con un cliente, por lo que todas las aplicaciones relacionadas con RRHH, personal y control de tiempo serán fundamentales.

Los recursos humanos constituyen en este tipo de empresas una de las partidas de coste más importante, lo que revela la necesidad de su control para convertir el mayor porcentaje posible de horas de trabajo en horas facturables a un cliente.

Seguramente el CRM, que es una de las partes más importantes de cualquier empresa y un apropiado catálogo de servicios (productos intangibles), será muy necesario para el éxito de la empresa.

No hay que descartar aplicaciones de *gestión de compras*, porque en la gran mayoría de los casos las empresas de servicios también necesitan proveedores externos.

Es más, a veces, las empresas de servicios proporcionan también bienes junto con servicios. Por ejemplo, un distribuidor informático especializado en la integración de soluciones de almacenamiento profesional que venda también las cabinas de almacenamiento y los discos necesarios, además de proporcionar los servicios especializados de instalación, configuración e integración.

Existen empresas que solo ofrecen servicios con sus propios productos y otras en las que el cliente puede comprar los productos a otra empresa y ellas solo se encargan de los servicios de instalación y configuración, por ejemplo.

Si se piensa en empresas que proporcionan servicios de soporte y asistencia, aparece enseguida la necesidad de una *aplicación de “ticketing”* para controlar el tiempo facturable a cada cliente e incluso una de *gestión documental* para almacenar las soluciones aplicadas a las incidencias.



Actividad propuesta 4.3

Relaciona los siguientes conceptos: empresas de servicios, outsourcing, externalización, tercerización, subcontratación y empresas de trabajo temporal.

4.5.1. Aplicación Proyectos

Esta aplicación permite gestionar los distintos proyectos en los que están ocupando su tiempo los empleados de la compañía.

Con ella se pueden organizar y planificar las distintas etapas de un proyecto, descomponer este en tareas, marcar hitos, así como administrar los recursos materiales y humanos que intervienen en el mismo. Además, permite el seguimiento para el control del desarrollo del proyecto con gráficos, diagramas y vistas.

Figura 4.24
Tareas de un proyecto.

Las tareas se pueden clasificar en distintos estados para obtener una vista Kanban de todas ellas ubicadas en sus correspondientes estados, que es el tablero básico utilizado por la aplicación *Proyectos*.

El proceso de creación de un nuevo proyecto supone la definición de los estados y las tareas que conforman el proyecto. Odoo proporciona ejemplos según el tipo de proyecto.

Figura 4.25
Asistente Kanban.

4.5.2. Aplicación Partes de horas

Esta aplicación permite utilizar hojas de tiempo para que cada empleado anote el tiempo que dedica a la realización de cada tarea de los distintos proyectos en los que esté participando y añada las descripciones de las acciones realizadas.

Así, la empresa puede conocer el trabajo de cada empleado en los distintos proyectos y, en función de su catalogación como tiempo facturable o no facturable, imputárselo posteriormente al cliente final e incluir su precio en la factura.

Partes de horas está integrada con las aplicaciones *Proyecto*, *Ventas*, *Empleados* y *Contabilidad*.

La opción de *Configuración* del menú principal permite establecer la unidad mínima de imputación de tiempo, la política de facturación o el sistema de recordatorios para empleados y responsables.



PARA SABER MÁS

Dispone de una App para iOS y Android con temporizador integrado que permite que cada empleado controle su dedicación incluso sin conexión con el ERP.

Al *Añadir una línea* en la aplicación se va introduciendo toda la información relacionada con la actividad realizada, el tiempo empleado, etc.

Posteriormente, de forma periódica, el responsable podrá validar cada hoja e imputar su coste al proyecto y al cliente final.

Partes de horas

Ajustes

GUARDAR DESCARTAR

Opciones Generales, CRM, Ventas, Compra, Inventario, Contabilidad, Proyecto, Partes de horas, Empleados, Punto de venta

Unidad de codificación: Horas

Round timesheets: Duración mínima: 15 minutos, Redondear: 15 minutos

Parte de horas atractivo: Sincronizar el tiempo dedicado con nuestras aplicaciones web/móviles. Botones para descargar en Google Play y App Store.

Facturación: Time Billing, Prestar servicios y facturar el tiempo empleado. Configura tus servicios.

Política de facturación: Registrar tiempo empleado y facturar según: Todos los partes de hora registrados, Solo partes de horas aprobados.

Control de partes de horas: Recordatorio para empleados, Activar un recordatorio periódico mediante correo electrónico para usuarios de partes de horas. Frecuencia: 1 día, Despues de la semana. Recordatorio para gerentes, Activar un recordatorio periódico mediante correo electrónico para administradores de partes de horas. Frecuencia: 3 días, Despues de la semana.

Ausencias

Figura 4.26
Ajustes Partes de horas.

Añadir una línea

Proyecto: AGR - S00022
Tarea: After-Sales Services
Empleado: DPC - S00023, Desarrollo, Office Design, Prueba 2

Fecha: 11/10/2020
Tiempo dedicado: 00:00 Horas

GUARDAR DESCARTAR

Figura 4.27
Introducción de datos.

Partes de horas					Parte de horas	A validar	Informes	Configuración	JJGL Admin (edu-lesnoweb)
					Empleado > Proyecto x		Buscar...		
					Filtros	Agrupar por	Favoritos		
Todos los partes de horas									
IMPORTAR									
<input type="checkbox"/> Fecha	Empleado	Descripción	Proyecto	Tarea					Duración (Horas)
▼ Juan José Admin (21)									67:00
▼ AGR - S00022 (4)									20:00
<input type="checkbox"/> 11/10/2020	Juan José Admin	Assembling	AGR - S00022	[SERV_89744] Senior Architect (Invoice on Timesheets)					05:00
<input type="checkbox"/> 01/10/2020	Juan José Admin	Design	AGR - S00022	[SERV_89744] Senior Architect (Invoice on Timesheets)					05:00
<input type="checkbox"/> 30/09/2020	Juan José Admin	Fine tuning	AGR - S00022	[SERV_89744] Senior Architect (Invoice on Timesheets)					05:00
<input type="checkbox"/> 29/09/2020	Juan José Admin	Delivery	AGR - S00022	[SERV_89744] Senior Architect (Invoice on Timesheets)					05:00
Agregar linea									
▼ After-Sales Services (8)									14:00
<input type="checkbox"/> 02/10/2020	Juan José Admin	Building	After-Sales Services	S00022: [SERV_585189] Customer Care (Prepaid Hours)					01:00
<input type="checkbox"/> 29/09/2020	Juan José Admin	Client meeting	After-Sales Services	S00022: [SERV_585189] Customer Care (Prepaid Hours)					01:00
<input type="checkbox"/> 28/09/2020	Juan José Admin	Requirements analysis	After-Sales Services	S00022: [SERV_585189] Customer Care (Prepaid Hours)					01:00
<input type="checkbox"/> 25/09/2020	Juan José Admin	Wood chopping	After-Sales Services	S00022: [SERV_585189] Customer Care (Prepaid Hours)					01:00
<input type="checkbox"/> 22/09/2020	Juan José Admin	Assembling	After-Sales Services	S00022: [SERV_585189] Customer Care (Prepaid Hours)					01:00
<input type="checkbox"/> 21/09/2020	Juan José Admin	Research	After-Sales Services	S00022: [SERV_585189] Customer Care (Prepaid Hours)					01:00
<input type="checkbox"/> 21/09/2020	Juan José Admin	Design analysis	After-Sales Services	S00023: [SERV_585189] Customer Care (Prepaid Hours)					04:00
<input type="checkbox"/> 14/09/2020	Juan José Admin	Document analysis	After-Sales Services	S00023: [SERV_585189] Customer Care (Prepaid Hours)					04:00
Agregar linea									
► Office Design (3)									24:00
► Prueba 2 (1)									01:00
► Research & Development (5)									08:00
► Marc Demo (11)									52:00

Figura 4.28 Informe horas empleadas.

Supuestos prácticos

1. Configuración de un equipo de venta.

- a) En la instalación SaaS del supuesto 3.1, crea un nuevo equipo de ventas denominado "Equipo-IES", configura como jefe de equipo al usuario utilizado durante la instalación y añade dos integrantes más. Define como meta del equipo 245 000 € de ventas. Crea las siguientes fases ("pipeline", flujo de ventas o flujo) para el equipo de ventas "Equipo-IES" según los datos siguientes:

Etapas:

- Nueva.
 - 1^a llamada.
 - Propuesta por correo.
 - 2^a llamada.
 - Reunión.
 - Pendiente de decisión.
 - Ganada.

- b) Configura las actividades para el equipo de ventas “Equipo-IES” de la siguiente manera:

- Nueva: siguiente actividad: 1^a llamada • Número de días: 1.
- 1^a llamada: siguiente actividad: propuesta por correo • Número de días: 1.
- Propuesta por correo: siguiente actividad: 2^a llamada • Número de días: 3.
- 2^a llamada: siguiente actividad: reunión • Número de días: 5.
- Reunión: siguiente actividad: pendiente de decisión • Número de días: 1.

Es posible que haya que crear algún “Tipo de actividad”.

2. En este ejercicio se va a configurar las actividades de venta en la instalación SaaS.

- a) En el apartado de *Productos*, crea dentro de la categoría *Saleables* (Vendibles o Productos a la Venta) las categorías Hardware, Software y Servicios. Dentro de categoría Software, crea las categorías Sistemas y Aplicaciones.
- b) Crea una oportunidad: Tartessos S.A. es una empresa interesada en comprar equipos portátiles. Han pedido un presupuesto de 5 unidades del modelo MacBook Pro (13-inch, M1, 2020) que tiene un PVP de 1 200 € + IVA y 5 licencias de Office 365 para Mac con un coste de 80 € + IVA. Crea los productos en su categoría, añádeles una foto y aumenta el stock inicial a 20 unidades de cada uno.

Se sabe que Tartessos tiene un presupuesto de 6 000 €. Da de alta el nuevo cliente, la oportunidad y establece las próximas acciones para ganar esta oportunidad.

La persona de contacto de Tartessos es el Sr. Antonio López y se supone que se conoce la cuenta de correo profesional –se puede utilizar otra cuenta propia o la de un compañero–. El resto de los datos serán inventados respetando el formato necesario.

Registra dos llamadas en fechas distintas, envíale después una propuesta adecuada al Sr. López y planifica para dentro de una semana una reunión para cerrar el trato.

c) Añade una cita en el calendario con las siguientes características:

- Nombre: Reunión de Equipo-IES.
- Asistentes: Equipo de ventas Equipo-IES.
- Descripción: Reunión semanal de toma de contacto.
- Duración: 1:15 minutos.
- Todos los lunes a las 09:30 en la sala de juntas, durante las próximas 12 semanas.

3. En este ejercicio se trabaja con la máquina instalada en el supuesto práctico 3.2.

- a) Instala los módulos siguientes: Ventas, Compra, Inventario, Facturación, Marketing por e-mail y TPV.
- b) Añade dos usuarios internos a la aplicación (empleados).
- c) Crea 3 clientes, 3 proveedores, 3 categorías de producto y 7 productos repartidos entre las distintas categorías.
- d) Importación de productos. Realiza la importación de productos utilizando los ficheros csv (disponibles en la zona de recursos digitales) creando las categorías

de producto apropiadas. (Es posible que no puedas importar de una vez todos los campos e incluso que algunos no los puedas importar. Resuelve este problema). Esta instancia la utilizará una empresa que dispone de dos almacenes claramente diferenciados (un almacén en la dirección central de la compañía y otro en una tienda al público que utilizará el módulo TPV). A la hora de hacer los pedidos, de hacer traspasos o de vender, se debe tener en cuenta de qué almacén sale o entra la mercancía.

- e) Crea una tarifa estándar de venta en la que los márgenes que se apliquen sean del 10 % para la categoría hardware y 15 % para software.
- f) Crea una segunda tarifa estacional para aplicarla entre el 1 y el 28 de febrero de cada año en la que, a los precios de la tarifa estándar se le aplique un 10 % de descuento.
- g) Realiza una operación de venta a dos clientes. Para ello habrá que elaborar dos presupuestos con dos y tres productos almacenados en la Central, respectivamente. Supón que ambos clientes aceptan y sigue cada proceso por todos los pasos intermedios hasta el envío de las facturas y el registro del pago de ambas. Acepta después una devolución de un producto incluido en una de ellas y sigue el proceso hasta la recepción del producto y la emisión de la factura rectificativa (de abono al cliente) comprobando el resultado en el stock del almacén.
- h) Haz una transferencia de productos entre ambos almacenes. Haz un informe de inventario antes y después de realizarla y comprueba que ha funcionado correctamente.
- i) Diseña el sitio web de la empresa incluyendo la tienda *online* con todos los productos que comercializa la empresa. Incluye un plano de Google Maps con la ubicación de la tienda.
- j) Permite las compras desde la web con forma de pago único: la transferencia bancaria.
- k) Busca las plantillas de los mails que se envían al cliente durante el proceso de compra y personalízalas cambiando el texto y el aspecto gráfico. Comprueba que los envíos de mail se realizan correctamente.

Resumen

- Una vez instalada una instancia de Odoo, hay que realizar una primera parametrización con los datos básicos de la compañía como el nombre que será utilizado por el sistema durante la creación de tablas asociadas a las distintas aplicaciones.
- La instalación de las aplicaciones y módulos necesarios se puede hacer de una vez al principio o paulatinamente cuando se advierta su necesidad.
- El módulo Conversaciones es una potente herramienta de comunicación interna y externa.
- Odoo proporciona un potente sistema de importación de datos.

- Con la aplicación CRM se pueden organizar las actividades y tareas diarias y aplicarlas a los distintos proyectos u operaciones en los que se participa.
- Existe un gran número de Aplicaciones. Las más usadas son Ventas, Compra, Inventario, Empleados o Contabilidad. Otras como Facturación, Punto de venta, Calidad, Ausencias, Nóminas o Comercio electrónico también son habitualmente instaladas en general.
- La aplicación Ventas automatiza parte de las tareas que realiza la fuerza de ventas.
- La aplicación Facturación permite la generación automática de facturas.
- La aplicación Compra automatiza el proceso de compras a proveedores.
- La aplicación Inventario permite controlar las existencias de los productos almacenables y gestionar las entregas y recepciones de material.
- La aplicación Empleados permite gestionar en tiempo real el capital humano de un empresa: desde el proceso de contratación, la elaboración de nóminas, la gestión de ausencias legales, médicas y vacacionales hasta el control de presencia y las valoraciones del desempeño.
- La aplicación Contabilidad proporciona información relevante y fiable sobre la situación económica de la empresa y facilita las liquidaciones de impuestos.
- La aplicación TPV proporciona las funcionalidades que necesita una tienda abierta al público.
- Una iniciativa es la recopilación de datos de un cliente potencial.
- Una oportunidad de venta es un momento propicio para hacer un negocio.
- La gestión de servicios necesita de un tratamiento especial desde el punto de vista de la comercialización. Existen aplicaciones específicas.
- La aplicación Proyectos permite planificar las distintas etapas de un proyecto, descomponerlo en tareas y administrar los recursos empleados.
- La aplicación Partes de horas gestiona las hojas de tiempo de cada empleado: reconocen la parte de jornada empleada en los distintos proyectos en los que interviene un recurso.

ACTIVIDADES DE AUTOEVALUACIÓN

1. La función del departamento de compras es:

- a) Guardar el producto en el almacén.
- b) Gestionar el abastecimiento de productos o de los materiales necesarios para crear el producto.
- c) Llevar el producto directamente a la cadena de producción.
- d) Todas las opciones son correctas.

2. ¿Se puede crear un canal privado con varios usuarios en el módulo Conversaciones?:
- a) Sí.
 - b) No.
 - c) Solo entre usuarios del mismo departamento.
 - d) Solo entre usuarios web.
3. ¿Es posible distinguir clientes potenciales y oportunidades?:
- a) Sí, en realidad se hace por defecto.
 - b) Sí, activando la opción “Clientes potenciales” en Configuración.
 - c) No.
 - d) Sí, pero tiene que hacerlo Administrador.
4. Si se convierte un cliente potencial en una oportunidad, ¿se pueden combinar dos oportunidades?:
- a) Se hace automáticamente si el contacto coincide.
 - b) No se puede.
 - c) Sí, respondiendo a una sugerencia del propio programa.
 - d) Sí, pero tiene que hacerlo el jefe de comerciales.
5. El idioma y la localización de los datos básicos influyen:
- a) En el formato de fecha.
 - b) En la aplicación de Contabilidad.
 - c) En los modelos de documentos de negocio.
 - d) Todas las opciones son correctas.
6. Para una empresa de servicios profesionales, son adecuadas las aplicaciones (señala todas las correctas):
- a) Punto de Venta (PoS).
 - b) Partes de horas.
 - c) Proyectos.
 - d) Fabricación.
7. Son Aplicaciones consideradas de *Back-Office* (señala todas las correctas):
- a) Empleados.
 - b) Comercio electrónico.
 - c) Inventario.
 - d) TPV.
8. Son características que definen un servicio (señala todas las correctas):
- a) Exportabilidad.
 - b) Modularidad.
 - c) Inseparabilidad.
 - d) Heterogeneidad.
9. Dentro de los informes de almacén el que permite comprobar el *stock* actual es:
- a) Inventario previsto.
 - b) Valoración de inventario.
 - c) Movimientos de productos.
 - d) Informe de inventario.

10. ¿Cómo se puede importar contactos a Odoo? (señala todas las correctas):

- a) Desde la aplicación Ventas.
- b) Desde ficheros de tipo csv.
- c) Desde ficheros de tipo doc.
- d) Desde ficheros de tipo xls.

SOLUCIONES:

1. **a** **b** **c** **d**

2. **a** **b** **c** **d**

3. **a** **b** **c** **d**

4. **a** **b** **c** **d**

5. **a** **b** **c** **d**

6. **a** **b** **c** **d**

7. **a** **b** **c** **d**

8. **a** **b** **c** **d**

9. **a** **b** **c** **d**

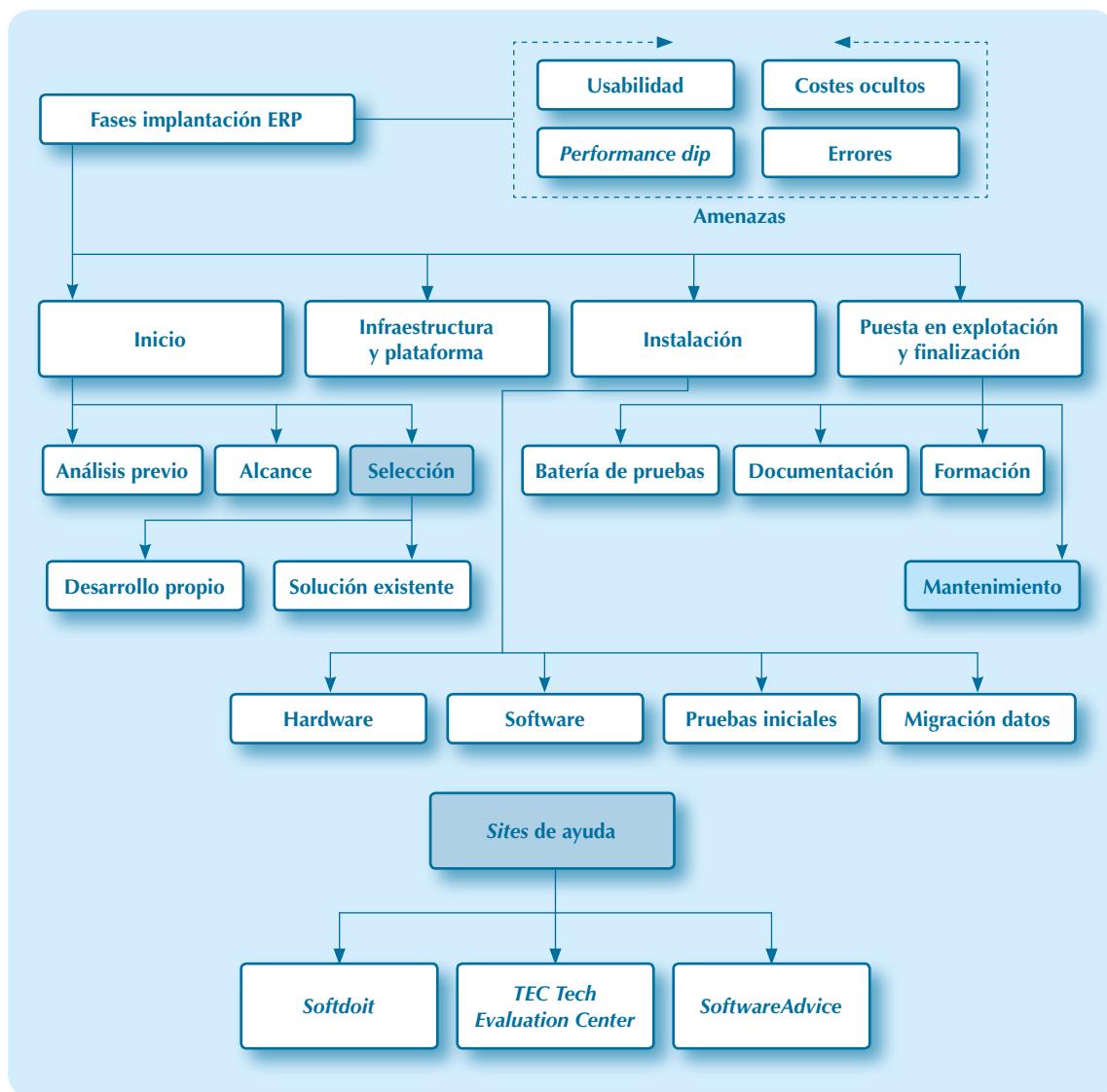
10. **a** **b** **c** **d**

Implantación de un ERP en una empresa

Objetivos

- ✓ Entender las dificultades inherentes al proceso de elección de un ERP.
- ✓ Conocer algunos portales de ayuda para la elección de software.
- ✓ Emprender el proceso de implementación de un ERP.
- ✓ Dividir el proceso en fases.
- ✓ Entender la necesidad de seguir un procedimiento ordenado de acciones.
- ✓ Asumir la importancia de la usabilidad de un ERP.
- ✓ Evitar la aparición de costes ocultos.
- ✓ Entender la necesidad de la evolución constante tras la implementación.

Mapa conceptual



Glosario

BPR (Business Process Reengineering). Reingeniería de procesos de negocio. Proyecto de redefinición de los procesos de negocio implantados en una empresa.

Caso de éxito. Documento comercial basado en una estructura de planteamiento, nudo y desenlace que explica los pormenores de una actuación exitosa de la empresa en un cliente.

Hosting. Alojamiento de servicios en un centro de datos o CPD.

Housing. Alquiler para el alojamiento físico de infraestructura propia, en un CPD.

Key user. Usuario clave. Usuario con destrezas tecnológicas y conocimiento del negocio que cumple un papel decisivo en un proyecto.

Pago por uso. Modelo económico basado en el pago de un servicio valorado en función del consumo de los recursos que se utilizan y durante el tiempo en el que se contrata. Utilizado habitualmente en el *Cloud Computing*.

Renting. Alquiler a largo plazo de bienes muebles. Contrato que, a cambio de una cuota fija mensual, permite disponer de unos bienes durante un periodo de tiempo y devolverlos a la finalización del contrato. Fiscalmente se trata como gasto y no como inversión amortizable.

Site. Sitio web. Conjunto de páginas web enlazadas.

Usabilidad. Definida en la norma ISO 9241-11 como: “el grado en que un producto puede ser usado por usuarios específicos para conseguir metas específicas con efectividad, eficiencia y satisfacción”.

Whitepaper. Libro blanco. Documento con estructura problema/solución que explica un concepto o tema concreto. Tiene su origen en los publicados por gobiernos para explicar su postura ante un tema determinado.

5.1. Introducción

La estructura de los mercados actuales exige a la mayoría de las empresas de cualquier tamaño y condición su adaptación a entornos tecnológicos que le permitan competir con otros actores que disponen de niveles altos de digitalización.

Por eso, existe un amplio nicho de negocio en el campo del desarrollo e implantación de sistemas de gestión empresarial del tipo ERP, ya que todas estas empresas, si quieren ser competitivas tendrán que adoptar este tipo de sistemas tan pronto como les sea posible.

Ya se ha comentado en capítulos anteriores acerca de la amplia variedad de soluciones que se pueden encontrar en la actualidad, también sobre las distintas modalidades de licencia y los entornos en los que se pueden utilizar, y sobre la conveniencia en algunos casos de realizar un desarrollo propio, lo que implica un trabajo previo de estudio, meditación y maduración para la toma de una decisión informada, pues esta determinará el éxito o el fracaso de la implantación del ERP en la empresa.

De hecho, suele ser tan complicado que no es extraño que las empresas opten por encargar a consultoras externas que analicen las necesidades de la empresa y emitan un informe final con las características y recomendaciones de implantación de un ERP, que permita a la gerencia tomar la decisión de adquirir una u otra aplicación entre las existentes en el mercado o desarrollarla.

5.2. *Sites de ayuda para elección de un ERP*

Es tal el volumen de la oferta tanto por el número de fabricantes desarrolladores, como por la variedad de productos que proponen cada uno de ellos y tantas las posibilidades de instalación y los tipos de licencia, que han aparecido empresas cuyo negocio consiste en la recopilación de información sobre las soluciones de software existentes en el mercado actual.

Han construido portales que contienen abundante información sobre las más importantes soluciones presentes en el mercado y proporcionan informes y comparativas que ayudan en la elección de la solución que más se adapta a un supuesto empresarial.

Tres de las más conocidas son:

- a) *Softdoit*: es una plataforma de una consultora *online* que permite comparar software de gestión para que los interesados puedan encontrar el más apropiado a sus intereses (<https://www.softwaredoit.es/index.html>).

Respondiendo a menos de una decena de preguntas, se obtiene un informe que incluye un ranking de las soluciones del mercado que más se adaptan a las necesidades proporcionadas en las respuestas.

- b) *TEC Technology Evaluation Center*: proporciona información, guías, informes, white-papers, casos de éxito... para encontrar la solución adecuada (<https://www3.technologyevaluation.com/es>). Dispone de una herramienta *itadvisor* con la que comparar software a la que se accede mediante una cuenta gratuita (<http://itadvisor.technologyevaluation.com>).

De nuevo, escogiendo entre una serie de opciones, se obtiene un listado de posibles soluciones.

- c) *Software Advice*: este es un portal más generalista en el que, además, se puede encontrar información comparada sobre herramientas de BI, sistemas de telefonía y comunicaciones para callcenters, seguridad perimetral, legal, GDPR... (<https://www.softwareadvice.com>).

Actividades propuestas



- 5.1.** Utilizando uno de los portales descritos, busca una solución ERP para una empresa española del sector servicios (educación superior) con los siguientes condicionantes:

- Más de 50 trabajadores.
- Facturación de 2,5 millones de euros.
- Necesita como mínimo módulos (funcionalidades) de Facturación, Ventas, CRM, Inventario, RRHH, Compras, Contabilidad.
- Dispone de un presupuesto para el proyecto de 30 000 €.

- 5.2.** Haz una pequeña descripción de los servicios que ofrece el portal [softwarereviews.com](https://www.softwarereviews.com) y compáralo con el resto de los portales mencionados en el apartado.

5.3. *Fases de la implantación de un ERP*

La implantación de un sistema ERP requiere de una serie de fases ordenadas, de seguimiento muy recomendable para asegurar la llegada a buen término del proyecto. Desde el punto de vista del equipo consultor encargado, es fundamental conocer el perfil de empresa con la que se va a

trabajar. No es lo mismo realizar una implantación para una multinacional que para una pyme, que normalmente cuenta con unos recursos más limitados. Además, la cultura empresarial, la política interna, los procesos y sus flujos de información son sustancialmente distintos en ambos casos.

La implantación de un ERP es un proyecto ambicioso que normalmente revolucionará el día a día del negocio, con lo que eso supone de traumático.

Para abordar una implantación de este tipo es imprescindible:

1. Un análisis previo que incluya un estudio de las necesidades detectadas, pero también una evaluación de los procesos empresariales utilizados en la actualidad.
2. La elección de la plataforma, propia o subcontratada, que se va a utilizar.
3. Una concreción del alcance funcional del desarrollo, donde se especifiquen los objetivos, habiendo escuchado a los departamentos que participan.
4. Un proyecto particularizado del desarrollo de software y parametrizaciones necesarias si se hubiera optado por el desarrollo, o una elección adecuada de los módulos que se van a instalar y las personalizaciones necesarias en el caso de optar por soluciones existentes en el mercado.
5. Un proceso de instalación y configuración del nuevo sistema.
6. La importación desde los sistemas anteriormente utilizados, de los datos históricos de la empresa, si los hubiera.
7. Un seguimiento y control estricto de la consecución de todos los objetivos que inicialmente se habían fijado, con especial atención a los elementos críticos para la rentabilidad del proyecto.
8. Una etapa de finalización del proyecto con la elaboración de baterías de pruebas de funcionamiento y rendimiento, la elaboración de documentación variada y la formación de usuarios.

5.3.1. Primera fase. Diseño de la solución

El *Análisis inicial* consiste en la descripción detallada de los procesos de negocio en los que intervienen los distintos departamentos, el estudio de la estructura de la información actualmente utilizada, el procedimiento general utilizado, los medios por los que fluye esta información entre las distintas áreas de la empresa y la información procesada que se obtiene de los distintos de informes que se manejan. Incluye la relación de necesidades detectadas y una evaluación de los procesos empresariales actuales.

Recoge de forma pormenorizada el funcionamiento de la empresa: identifica los procesos clave, las tareas que se repiten y por tanto pueden automatizarse y qué nuevas prestaciones son deseadas.

Como resultado de este análisis, es posible que se detecten procedimientos manifiestamente mejorables. Puede ocurrir, entonces, que la gestión de los procesos de negocio se adapte a las funcionalidades del ERP y no al revés, lo que se conoce como reingeniería de procesos (BPR). Esta debe ser una decisión meditada de los responsables, valorando si el procedimiento marcado por el ERP es más eficiente que el seguido hasta entonces por la empresa, y supone una mejora que influirá en la competitividad.

En la mayoría de las implementaciones se adopta un enfoque mixto. La empresa adaptará parcialmente la forma en la que trabaja al diseño de procesos de negocio implícito en el ERP, para así beneficiarse de las que normalmente se denominan “buenas prácticas”, pero también

se adaptará el ERP a las particularidades de la empresa ya sea con una adecuada selección del software y módulos a instalar, o mediante la parametrización y adaptación del código.

El *Alcance funcional* especificará los *requisitos* de la aplicación, los objetivos que se quieren conseguir, no solo incluyendo los procesos de negocio actuales, sino también aquellos nuevos procedimientos que se van a incorporar al nuevo sistema y que con el sistema actual no se realizan o son inefficientes, surgidos durante el análisis inicial.

Consiste en un diseño de la aplicación que enumere cómo se resolverán las necesidades de la empresa y se integrarán los datos históricos existentes en la nueva aplicación. Es decir, la adaptación de la aplicación a lo obtenido en el análisis inicial.



PARA SABER MÁS

Los requisitos son las características que un software debe tener para tener valor y proporcionar una utilidad al usuario.

En trabajos de desarrollo de software, los llamados *requisitos de software* pueden ser:

- *Funcionales*. Características que describen qué debe proporcionar el desarrollo.
- *Reglas de dominio*. Restricciones generales: accesos, políticas, estándares, compliance legal, como la adaptación a la LOPD, por ejemplo, que aplican a todos los sistemas de la organización.
- *No funcionales*. Son aquellos requisitos a nivel del sistema pero que en sí mismos no tienen valor de usuario. Son atributos de calidad que describen prestaciones: velocidad, precisión, uso de recursos, seguridad de la información, autenticación, fiabilidad, disponibilidad etc.

Para la descripción de los requisitos existen dos enfoques:

- a) *Dirigido al producto*. Descripción pormenorizada de los objetos, funciones, atributos que el sistema ha de implementar para cumplir los objetivos esperados.
- b) *Dirigido al usuario (al uso)*. Descripción de objetivos de los usuarios, previendo el uso que harán del sistema para conseguir alcanzarlos.

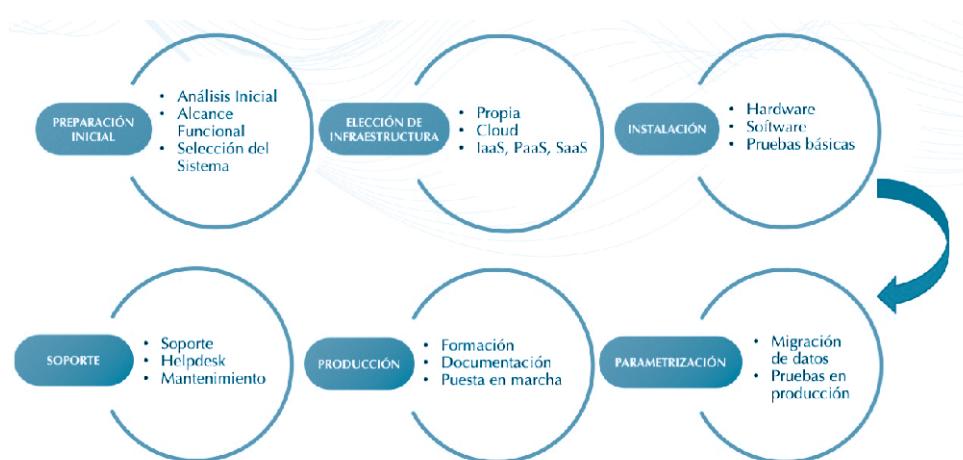


Figura 5.1
Fases de implantación de un ERP.

En este momento es cuando, con la información recabada en los dos primeros puntos, se puede tomar la decisión del camino que se va a seguir, es decir, la *Selección del sistema*: si optar por una solución existente en el mercado, identificando los módulos apropiados del ERP seleccionado, adaptándolos a la empresa e incluso modificando los procesos de negocio o, si por el contrario, se acomete un proyecto de desarrollo de software propio, que como ya se ha comentado en apartados anteriores, solo es aconsejable en el caso de procesos empresariales muy específicos.

En definitiva, la identificación de los procesos claves de la empresa, la estructura de los datos y el *workflow* determinarán la elección de una aplicación u otra en función de lo que ofrezca la definición de los objetivos perseguidos, los recursos necesarios y el presupuesto.

Estos tres puntos: *Análisis inicial*, *Alcance funcional* y *Selección del sistema*, son el resultado del trabajo inicial realizado por el equipo de consultoría.

Se realiza también una estimación de tiempo, coste y recursos externos y/o propios necesarios.

5.3.2. Segunda fase. Estudio de las necesidades de infraestructura

En este punto, y de nuevo con las diferencias de enfoque en función del camino elegido, es necesario revisar la infraestructura de IT de la que se dispone, para ver si se cumplen las necesidades del sistema que se va a implementar.

No cabe duda de que una apropiada elección en este sentido es imprescindible para el éxito del proyecto. Pero no es fácil, ni barato, disponer de una infraestructura que asegure, y mantenga a lo largo del tiempo, un soporte adecuado al ERP.

Ya se ha visto en apartados anteriores que, en la actualidad, cada vez más empresas optan por el *pago por uso* de la misma forma que en lo que se refiere a infraestructura cada vez más se opta por soluciones de renting frente a la adquisición de equipamiento con sus consiguientes amortizaciones.

Los modelos basados en IaaS, PaaS y SaaS evitan realizar inversiones grandes y proporcionan acceso de los usuarios desde cualquier lugar y en cualquier horario.

5.3.3. Tercera fase. Desarrollo de la Implantación (como proyecto)

En esta etapa, normalmente encargada al fabricante del ERP o a uno de sus distribuidores autorizados, o al equipo de desarrollo en su caso, se detallan y planifican todas las tareas que forman parte de la implantación: cuándo se hace, quién la realiza, a quién afecta.

Incluye la posible instalación de hardware y su configuración (*on-site, hosting, housing*, máquinas físicas, entorno virtual). También la instalación del software de sistema necesario (elección del sistema operativo, servicios, programas auxiliares).

El siguiente paso será el de la realización de las distintas tareas de configuración y adaptación necesarias y la ejecución de las baterías de *pruebas* iniciales, que después se ampliarán durante el periodo de coexistencia del sistema o sistemas antiguos y el nuevo.

Para la empresa, es fundamental asegurarse de la posibilidad de la *migración de datos* anteriores, es decir, el traspaso de datos desde el histórico del sistema anterior. De hecho, suele ser un requisito indispensable que se tiene en cuenta durante la elección de la solución. Sería poco

deseable tener que mantener durante años (los marcados por las obligaciones legales y fiscales) ambos sistemas en funcionamiento. Por eso la migración de datos es una parte fundamental.

5.3.4. Cuarta fase. Puesta en marcha y explotación

En la fase de puesta en marcha se realizan las *pruebas definitivas* de todos los módulos. Estas pruebas de funcionamiento se pueden realizar de dos formas distintas:

- Pruebas de funcionamiento en paralelo. Se trabaja en paralelo con los dos sistemas para evaluar posibles desajustes, y si los resultados coinciden se abandona el antiguo sistema. Esto implica tener que efectuar entradas de datos duplicadas y un gran coste en tiempo.
- Bloqueo del sistema antiguo y puesta en marcha del nuevo con el ERP recién implementado. Esta opción tiene el inconveniente de que, si no se ha probado suficientemente el sistema nuevo, puede fallar en la puesta en marcha con el consiguiente perjuicio en las operaciones de la empresa.

Previo a la puesta en producción se aborda la *formación*, no solo de los distintos grupos de usuarios, sino también de los administradores (o usuarios vip que ayudarán con su *expertise* informática y de gestión). Es otra labor fundamental en esta fase. Y muy relacionado con esto, la elaboración de la *documentación*, tanto en formato manual como mediante la programación de ayudas contextuales y sistemas asistentes.

La documentación generada es un aspecto importante en cualquier proyecto, y es precisamente, una de las que más reticencias provoca en los responsables de su desarrollo y que más debe preocuparse el cliente en conseguir.

La documentación técnica exhaustiva del proyecto, incluyendo tecnología, código, especificaciones y requerimientos, es fundamental para que la empresa como tal no quede cautiva de los técnicos que circunstancialmente hayan participado en el proyecto inicial, y que, a estas alturas, ya se debe tener claro que es un proceso vivo con final a largo plazo.



TOMA NOTA

Es responsabilidad de la gerencia asegurarse de disponer de toda la información técnica detallada que generen las distintas fases del proyecto para, en un futuro, poder abordar cualquier tipo de cambio, mejora o recodificación sin depender nada más que de quien se quiera asignar como responsable (interno o externo) de las nuevas fases.

Además, la documentación de usuario, no ya la elaboración de intensos manuales, que también son necesarios, sino una buena ayuda *online*, así como asistentes claros e intuitivos son determinantes para la aceptación de la nueva aplicación.

En el caso particular de instalación de un ERP *Open Source*, con toda probabilidad será necesario abordar la creación de nuevas vistas e informes, además de la *personalización* del entorno de trabajo de los distintos grupos de usuarios y, en muchos casos incluirá el desarrollo de nuevos componentes para cubrir determinadas funcionalidades requeridas.

Esta fase suele incluir un *soporte tutelado*.

Transcurrido un tiempo prudencial, y después de una revisión exhaustiva final en cuanto al cumplimiento de requisitos no funcionales (prestaciones de velocidad, seguridad, disponibilidad), funcionalidad, y aprendizaje del nuevo entorno, y realizados los *ajustes finales* surgidos en esta última fase, se puede dar por terminada la fase troncal de implantación.

Es el momento de la puesta en servicio y explotación del nuevo ERP.

En función del nivel de modificación del software ERP original en el caso de adaptación de soluciones existentes, o siempre, en el caso de desarrollos propios, hay que considerar la necesidad de un *mantenimiento y soporte* posterior que contemple la adaptación a nuevos entornos técnicos y al marco normativo e incluya la resolución de problemas futuros que se van a producir, tanto con motivo de adaptaciones al entorno como por circunstancias sobrevenidas de incompatibilidad con nuevas versiones de los sistemas, es decir, que las actualizaciones o mejoras pueden no ser compatibles con la personalización realizada en el código, lo que puede requerir recodificaciones con todo lo que ello supone en cuanto a coste, nuevas pruebas, estrés de los equipos, etc.

El soporte asegurará la resolución de incidencias habidas en el desarrollo normal de la actividad diaria.

5.4. Usabilidad

La usabilidad de un ERP es la medida de la facilidad, eficiencia, comodidad y satisfacción subjetiva de la experiencia que tiene un usuario cuando lo utiliza.

RECUERDA

- ✓ La usabilidad, tan importante en cualquier sistema, es más importante, si cabe, en los ERP debido a la complejidad, extensión y multifuncionalidad de estos.

La importancia de la usabilidad trasciende hasta el entorno de los ERP buscando disponer de una interfaz en la cual se pueda ver todo el flujo de forma clara, lo que facilitará la comprensión y asimilación de las operaciones por parte de los usuarios, operaciones que deben responder a un sistema automatizado para que los procesos internos fluyan por las distintas etapas y no necesiten la intervención del usuario.

El uso de un cliente web es la tendencia general. El trabajo con el ERP debe ser intuitivo y consistente para que se pueda manejar la aplicación fácilmente gracias a la estandarización de las distintas pantallas. Además, la adaptabilidad a los distintos dispositivos, como teléfonos móviles y tabletas, que cada vez son más utilizados por los usuarios, es una característica irrenunciable en el desarrollo de un ERP moderno.

Son características imprescindibles:

- La posibilidad de personalización de pantallas, menús e informes por parte del usuario o, al menos, del responsable del departamento.
- Existencia de unos sistemas de búsqueda parametrizables, adecuados y potentes.
- Posibilidad de procesos de alta de nuevos elementos sin tener que salir de la pantalla actual.
- Sistemas de avisos del estado de las actividades del usuario.
- Presencia de una ayuda contextual y procesos de creación y operación asistidos.
- La coherencia y uniformidad de las distintas pantallas.

5.5. Evolución tras la implementación

Tras la implementación de un ERP las empresas suelen experimentar una caída de su eficacia en general. Es paradójico que, después de abordar un proyecto con un alto coste en recursos humanos y económicos, aun con una gestión y seguimiento serios del proyecto, el resultado se traduce en que la empresa trabaja peor que antes: la calidad del servicio prestado y la satisfacción de los clientes disminuyen, los empleados no se sienten cómodos y su productividad se ve mermada.

En realidad, esta caída en el rendimiento, conocido como *performance dip*, es normal que se produzca, puesto que los cambios importantes como este requieren un tiempo de ajuste antes de demostrar su verdadero potencial.

En este punto, todavía hay que solucionar algunas cuestiones y esperar un tiempo prudencial a que los usuarios se hagan al nuevo entorno de trabajo. Es importante ser conscientes de este fenómeno, y advertir con antelación a todos los participantes, para así evitar esta reacción negativa. Uno de los mayores desafíos con el que se encuentra el equipo responsable del desarrollo del proyecto es el de vencer la natural negativa al cambio de los usuarios a un nuevo sistema que trastoca sus rutinas: es la barrera de entrada colocada por los propios integrantes de la organización.

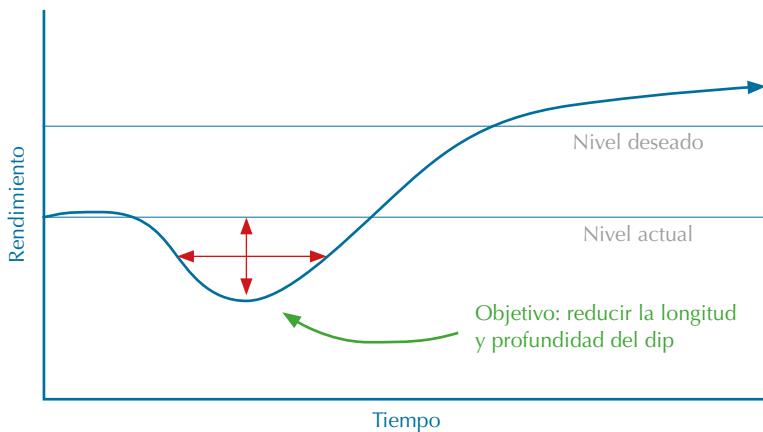


Figura 5.2
La curva de *Implementation dip*.

Cuando un usuario concluye que un ERP le impide realizar su trabajo de forma correcta, echa la culpa a que la funcionalidad del sistema es deficiente, o a que la implementación no fue la correcta, olvidando una tercera posibilidad: la de que se trata de una decisión consciente de la gerencia.

Desde luego, la causa de la frustración del usuario es una discrepancia con el diseño del proceso de negocio implementado (por ejemplo, no poder reservar stock sin pedidos en firme, no poder aceptar pedidos si no hay una propuesta previamente aprobada, tener que esperar la aprobación de otro departamento dentro del *workflow*, etc.). Con la mayoría de ERP actuales, un usuario solo puede aceptar un pedido de cliente, realizar una compra, ofrecer un descuento a un cliente, o pagar una factura, de la forma establecida por el sistema.

Estas limitaciones, si realmente son el resultado de la ausencia de funcionalidades apropiadas del ERP, pueden provocar la reformulación o readaptación del código o una nueva parametrización una vez que la dirección de la empresa la asuma.

Pero con frecuencia este tipo de circunstancias no son más que el reflejo de una decisión empresarial de la dirección, que en realidad quiere asegurarse de que esa sea la forma en la que se lleve el negocio aun en contra de la opinión del usuario.

El ERP se convierte así en el instrumento que garantiza que los empleados siguen las reglas del negocio marcadas por la gerencia, puesto que las limitaciones que parece imponer son simplemente decisiones de negocio.

En resumen, además de ser complicados y costosos, los proyectos de implementación de un ERP se alargan en el tiempo. Nuevos módulos, nuevas parametrizaciones, actualizaciones o mejoras los convierten en una sucesión continua de proyectos.

5.6. Errores y costes ocultos

Es proverbial la creencia generalizada de que la implantación de sistemas de información gerencial se convierte en muchas ocasiones en proyectos fracasados, si no traumáticos, o que experimentan grandes retrasos y sobrecostes.

A veces, esto está provocado por los conflictos surgidos entre proveedor y cliente, o por la poca implicación de los mandos intermedios, o por la falta de cualificación del personal encargado de la parte del cliente, o por una conjunción de varias de ellas.

Pero uno de los mayores errores cometidos históricamente en el desarrollo de proyectos relacionados con la implantación de un sistema de gestión empresarial es el de la no superación de la barrera de entrada colocada por los usuarios.

No “vender” bien el proyecto internamente puede ser un motivo de fracaso seguro. Si los estamentos de cualquier nivel no están implicados, los problemas esperados al abordar cualquier cambio en una organización se multiplicarán exponencialmente y harán peligrar la llegada a buen puerto del proyecto.

TOMA NOTA



Relación no exhaustiva de errores habituales que provocan el fracaso en la implantación de un ERP en la empresa:

- Contratación de partner no cualificado para el desarrollo o la implementación.
- Elección errónea del software.
- Implementación equivocada sin reingeniería de procesos de negocio.
- Enfoque del proyecto fundamentalmente técnico en lugar de empresarial.
- Expectativas irreales.
- Objetivos indefinidos y responsabilidades diluidas.
- Ausencia de liderazgo.
- Defectuosa gestión del cambio.
- Comunicación o ineficaz o ausente.
- Elección errónea de los *key users*.
- Baja implicación de los usuarios.
- Falta de motivación y resistencia al cambio de los usuarios.
- Dificultad de formación de los usuarios.

Todo proyecto que aspire al éxito debe evitar el incumplimiento de los plazos previstos, superar desproporcionadamente el presupuesto asignado y, por descontado, evitar presentar un funcionamiento no esperado.

Hay que considerar que parte de la organización (no solo la parte externa que ayude en el proceso) se verá involucrada y esto afectará a la productividad de sus integrantes.

La implementación de un ERP suele, con frecuencia, dar la impresión de no acabar nunca, de convertirse en un proyecto al que no se le ve el fin. Pero, dejando al margen los proyectos fallidos, la razón de ello suele ser que se implementa inicialmente un conjunto de módulos y, paulatinamente, se abordan ampliaciones que suponen la integración de nuevos módulos.

WWW

Recurso web

Estudio de la utilización de sistemas ERP y grado de satisfacción con los beneficios obtenidos (Ibermática):



Supuestos prácticos

1. Odoo publica un informe titulado *Implementation Methodology* en el que explica la importancia de planificar de forma correcta un proyecto y las mejores prácticas para abordar un proyecto de implantación de una solución ERP como Odoo. Lee el informe al que se puede acceder con este código QR y haz un resumen.
2. Aplica los conocimientos adquiridos en el módulo de “Entornos de desarrollo” acerca de las metodologías de desarrollo apropiadas en el caso de un proyecto de implantación de un ERP para una empresa de prestación de servicios profesionales y proyectos, llave en mano, del sector de las tecnologías de la información con recursos de desarrollo propios de la empresa.



Elabora una solución básica con los módulos que se consideren imprescindibles, ya sea considerando modelos de ciclos de vida en cascada, con retroalimentación, evolutivos (iterativo incremental o espiral).

Aplica metodologías *Agile* con elaboración de prototipados rápidos e iteraciones cortas con estrecha colaboración entre desarrolladores y responsables del negocio.

Resumen

- La utilización de un ERP es fundamental para la competitividad de la empresa.
- La elección del ERP adecuado es una tarea de enorme dificultad y trascendencia para el futuro de la empresa.
- Existe un nicho de negocio importante para empresas que se dedican a asesorar mediante comparativas y herramientas que generan informes acerca de las soluciones ERP más adecuadas a un caso concreto.
- La implantación de un ERP es un proyecto ambicioso que requiere del cumplimiento, cuanto más estricto, mejor, de una serie de patrones y buenas prácticas divididas en fases que incluyen:
 - Un análisis previo.
 - La elección de la plataforma.
 - Una concreción del alcance funcional del desarrollo.
 - Un proyecto particularizado del desarrollo de software o de la elección adecuada de la solución existente.
 - Un estudio de la infraestructura y un proceso de instalación y configuración del nuevo sistema.
 - La integración con sistemas existentes y la importación de datos históricos
 - Un seguimiento y control.
 - La realización de pruebas de funcionamiento y rendimiento.
 - La elaboración de documentación y la formación de usuarios.
 - Un mantenimiento posterior a lo largo del tiempo.
- El análisis inicial recoge de forma pormenorizada los procesos de negocio y evalúa su idoneidad.
- El alcance funcional especificará los objetivos que se quieren conseguir.
- La selección del sistema se traduce en decidir si optar por una solución existente en el mercado, o por realizar un desarrollo propio.
- El estudio de las necesidades de infraestructura dará como resultado la información acerca de las distintas opciones, sus costes, modalidades de contratación y amortizaciones.
- Un desarrollo a medida, con personal interno o con partner contratado, o la elección del producto comercial es el siguiente paso del proyecto.
- Un requisito indispensable es la migración de datos anteriores al nuevo sistema.
- Durante la fase final se elaboran las baterías de todo tipo de pruebas de funcionamiento y la integración con el resto de sistemas.
- La formación específica de los distintos grupos de usuarios, así como de personal más especializado es una fase crítica que repercute en el éxito de la implantación de un ERP.
- Es imprescindible la implementación de un sistema de mantenimiento y soporte posterior que asegure la resolución de incidencias habidas en el desarrollo normal de la actividad diaria y la adaptación a nuevos entornos técnicos y del marco normativo.
- La usabilidad en los ERP es muy importante, debido a la complejidad, extensión y multifuncionalidad de estos.

- Es normal que se produzca una caída en el rendimiento inicial de la solución ERP, conocido como *performance dip*.
- Una gran cantidad de posibles errores de distinto tipo jalonan el camino del desarrollo de implantación de un ERP.

ACTIVIDADES DE AUTOEVALUACIÓN

1. Señala cuál de las siguientes tareas no forma parte del proceso de instalación e implantación de un sistema ERP/CRM:
 a) Formación de usuarios.
 b) Migración de datos.
 c) Corrección del software.
 d) Configuración del sistema.
2. En la implantación de un ERP, el análisis previo incluye:
 a) La concreción del alcance.
 b) Un informe de necesidades detectadas.
 c) Las instrucciones de instalación.
 d) Una propuesta de la infraestructura.
3. La usabilidad de un ERP mide (señala todas las correctas):
 a) La complejidad.
 b) La multifuncionalidad.
 c) La facilidad.
 d) La satisfacción subjetiva.
4. ¿Qué expresión define mejor el concepto *performance dip*?:
 a) Caída de la eficacia general.
 b) Aumento de la productividad.
 c) Caída de la satisfacción general.
 d) Disminución inicial de clientes.
5. Señala todos los requisitos no funcionales:
 a) Velocidad.
 b) Restricciones de acceso.
 c) Disponibilidad.
 d) Reglas de dominio.

6. ¿Cuál de los siguientes no es un factor de éxito o fracaso de un ERP?:

- a) Implicación de la gerencia.
- b) Resistencia al cambio de los empleados.
- c) Presupuesto económico del proyecto.
- d) Falta de medios.

7. Respecto a los factores que se deben tener en cuenta a la hora de elegir un ERP, señala la respuesta correcta:

- a) Periodo de migración y adaptación sencillo.
- b) Informes y análisis proporcionados.
- c) Características de uso.
- d) Se deben tener en cuenta todos los mencionados.

8. En la fase de diseño se define:

- a) La operación de mantenimiento.
- b) El alcance funcional.
- c) La migración de datos.
- d) El proceso de instalación y parametrización.

9. Los enfoques para la definición de requisitos son:

- a) Dirigido al producto y dirigido al usuario.
- b) Dirigido al desarrollo y dirigido a una solución existente.
- c) Funcionales y no funcionales.
- d) Dirigido a objetivos y dirigido al desempeño.

10. Durante los trabajos de adaptación de un ERP a una empresa se aborda:

- a) La modificación y creación de informes.
- b) La programación de módulos específicos.
- c) El estudio de viabilidad del sistema.
- d) La eliminación de la redundancia de datos.

SOLUCIONES:

1. **a** **b** **c** **d**

2. **a** **b** **c** **d**

3. **a** **b** **c** **d**

4. **a** **b** **c** **d**

5. **a** **b** **c** **d**

6. **a** **b** **c** **d**

7. **a** **b** **c** **d**

8. **a** **b** **c** **d**

9. **a** **b** **c** **d**

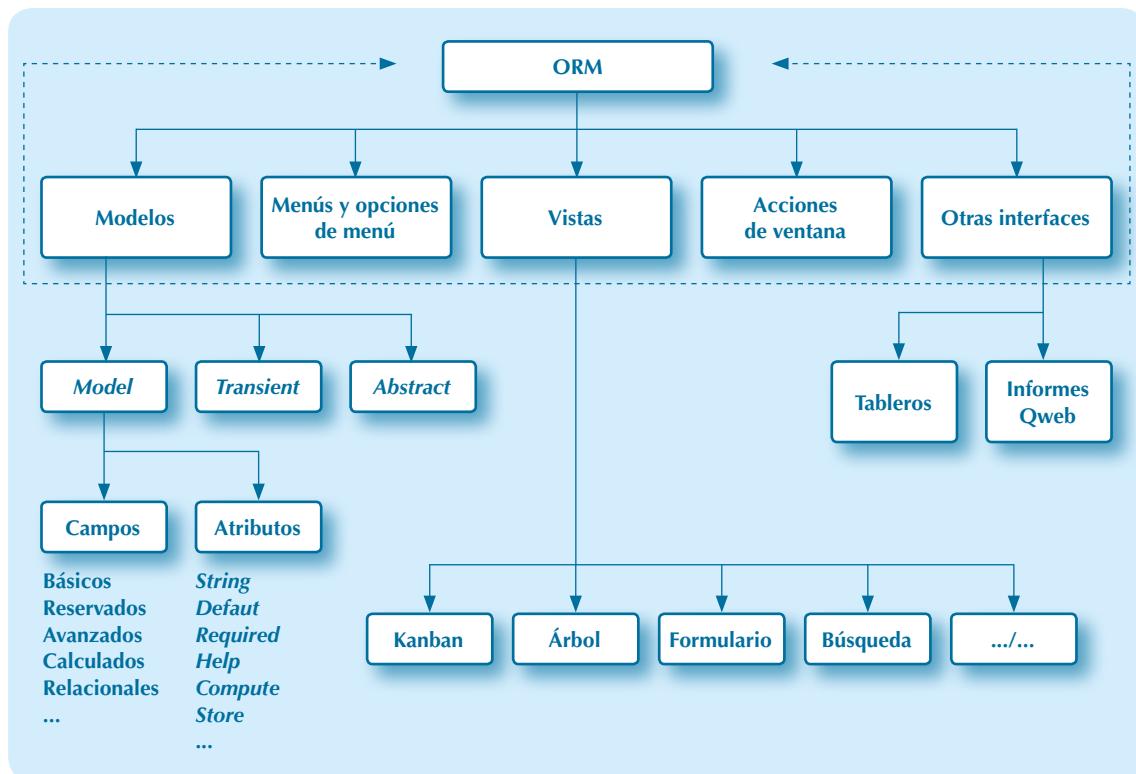
10. **a** **b** **c** **d**

Visión general de Odoo ERP

Objetivos

- ✓ Conocer los conceptos básicos sobre los que se desarrolla Odoo ERP.
- ✓ Conocer el significado de ORM.
- ✓ Entender el concepto de modelo.
- ✓ Describir los tipos de campos que componen el modelo.
- ✓ Conocer los principales atributos.
- ✓ Entender el funcionamiento de menús y opciones de menú.
- ✓ Estudiar el concepto de las acciones de ventana.
- ✓ Diferenciar las distintas interfaces de usuario.
- ✓ Entender el concepto de vista y los distintos tipos.
- ✓ Conocer el funcionamiento de los tableros.
- ✓ Elaborar informes.
- ✓ Entender la serialización de datos.
- ✓ Utilizar el concepto de herencia.
- ✓ Comprender y crear secuencias de documentos.
- ✓ Conocer el motor de plantillas Qweb.

Mapa conceptual



Glosario

Framework. Marco o entorno de trabajo. Básicamente, se trata de una plantilla conceptual o guía (modelo de programación) para el desarrollo de aplicaciones.

Herencia. Técnica de programación mediante la que se crean objetos a partir de otros preexistentes, que extienden la funcionalidad de estos.

Secuencia. Patrón de números correlativos.

Serialización. Proceso de codificación de un objeto (estructura de datos, función, método) de un programa en bytes para su transmisión en serie.

Stream. Flujo. Se dice de la transmisión de datos mediante un flujo continuo

XML. eXtensible Markup Language. Lenguaje de Marcas Extensible. Metalenguaje no predefinido y que da soporte a bbdd, que permite definir lenguajes de marca.

6.1. Introducción. Conceptos básicos

Odoo ERP está desarrollado en lenguaje Python y utiliza la base de datos PostgreSQL. Además, usa XML y Qweb, motor de plantillas basado en XML que permite generar páginas HTML.

Existen cuatro conceptos básicos para entender su estructura: modelos, vistas, menús y acciones. De forma básica se pueden definir de la siguiente manera:

- Un modelo es un objeto que contiene su propia descripción y sus relaciones con otros objetos.
- Una vista es lo que la aplicación muestra en la pantalla mediante el navegador.
- Una opción de menú es cada una de las posibles elecciones que se pueden hacer a partir de la ubicación del usuario en la vista.
- Una acción (de ventana o menú) es lo que se ejecuta al pulsar una opción de menú o un botón.

Tanto las vistas como los menús y las acciones se construyen en archivos XML.

La *herencia* es la propiedad mediante la que un objeto deriva de otro extendiendo su funcionalidad, es decir, en un entorno de programación orientada a objetos, permite que una clase adquiera las características (atributos y métodos) de otra clase.

Una de las características más importantes de Odoo es la capacidad para agregar objetos sin modificar directamente los objetos de base, mediante un potente mecanismo de herencia, que funciona añadiendo capas con las modificaciones por encima de dichos objetos. Los mecanismos de herencia son aplicables tanto a los modelos como a las vistas y, en general, a la lógica de negocio.

El procedimiento estándar para adaptar los módulos de Odoo a las necesidades de una empresa en particular consiste en crear nuevos objetos agregando las modificaciones a uno existente, en lugar de modificar directamente este.

6.2. ORM (*Object Relational Mapping*)

La arquitectura de Odoo se estructura en tres niveles: datos, lógica y presentación.

- a) La capa de *datos* es el nivel más bajo. Odoo se basa en la base de datos PostgreSQL mediante la que gestiona el almacenamiento y la persistencia de los datos. En lo que se refiere a los archivos binarios (archivos de documentos, imágenes, etc.), normalmente los almacena en un directorio denominado *filestore*.
- b) La capa *lógica* que reside en el servidor de Odoo es la responsable de las interacciones con la capa de datos (peticiones y recepción de datos). En ella se encuentra el motor de asignación relacional de objetos (ORM) que se describirá en próximos párrafos. Realiza los cálculos y validaciones y devuelve los datos al cliente. Odoo utiliza Python.

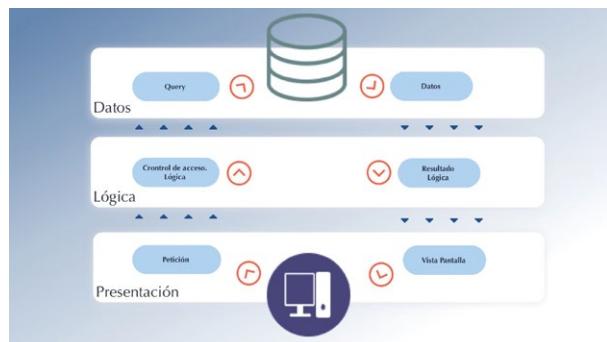


Figura 6.1
Arquitectura
en capas de Odoo.

- c) La capa de *presentación* es la que muestra los datos y permite la interacción del usuario mediante un cliente que interactúa con la API de ORM a través de llamadas a procedimientos remotos (RPC). Odoo utiliza XML, JavaScript y Qweb.

ORM o Motor de asignación de objetos es un *framework* que proporciona funciones para enlazar (mapear) la estructura de una base de datos relacional con la estructura de entidades de la aplicación para que los datos de la aplicación queden vinculados a la base de datos de forma automática.

Este mapeo automático mediante el ORM simplifica enormemente el trabajo del programador, ya que se pueden tratar las consultas CRUD (*Create, Read, Update, Delete*) como si fueran objetos gestionables mediante lenguaje de programación. Permite olvidar la complicación de las sentencias SQL evitando tener que escribirlas, lo que se traduce en una reducción de código de acceso a la BBDD. Sin embargo, puede generar código basura que puede impactar en el rendimiento de la aplicación.

Lo ventaja principal es que la modificación del modelo (por ejemplo, añadir un campo) no implica la revisión de todas las sentencias INSERT, SELECT y UPDATE para añadir la modificación en ellas.

Utilizando ORM, al añadir una propiedad a una clase, este se encargará de modificar las sentencias. En algunos ORM también hay que modificar archivos XML de las vistas.

En este sentido, Odoo cambia CRUD por métodos Postgres en models.Model:

CUADRO 6.1

Cuadro comparativo CRUD-Métodos

CRUD	models.Model (postgres métodos)
Create	create
Read	search, browse
Update	wtitle
Delete	unlink

Gracias a ORM el programador se abstrae de la base de datos y se centra en el desarrollo. Además, en caso necesario, puede cambiar el motor de base de datos sin mucha dificultad.



TOMA NOTA

Para generar un presupuesto en Odoo, simplemente se instancia un objeto presupuesto estableciendo su número de orden mediante el constructor.

La líneas de detalle se obtienen de la propiedad líneas del objeto presupuesto. Será un registro de la base de datos que mapea. Cada dato de un producto estará en la propiedad del modelo correspondiente y será un campo de la base de datos. De esta manera no se necesitan consultas, relaciones, claves foráneas...

6.3. Modelos

En Odoo, cualquier dato es accesible a través de *modelos*. Los modelos constituyen las tablas de la BBDD. Cada *modelo* es un objeto que contiene información descriptiva en campos, información acerca de relaciones con otros objetos, así como de lógica de negocio.

Estos objetos se declaran mediante modelos Python, que se pueden consultar en *Ajustes/Técnico/Estructura de la base de datos/Modelos*



Figura 6.2
Listado de modelos básicos de Odoo.

Los modelos se pueden editar para modificarlos o se pueden crear desde cero.

La pantalla que describe un *modelo* incluye la descripción de todos los campos, los módulos en los que se usa, así como los permisos de acceso y las vistas asociadas, entre otras cosas.

Nombre de campo	Etiqueta de campo	Tipo de campo	Requerido	Sólo lectura	Indexado	Tipo
__last_update	Última modificación en	Fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
active	Activo	booleano	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
active_lang_count	Active Lang Count	entero	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
activity_date_deadline	Siguiente plazo de actividad	fecha	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
activity_exception_decoration	Decoración de Actividad de Excepción	Selección	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
activity_exception_icon	Icono	Carácter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
activity_ids	Actividades	one2many	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
activity_state	Estado de la actividad	Selección	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
activity_summary	Datos resumen de la actividad actualizada	Carácter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base

Figura 6.3
Campos del modelo Contacto (*res.partner*).

Son clases Python que se corresponden con tablas Postgres. Se escriben en Python y todos extienden la clase *models.Model*. Es el conjunto de *campos* y atributos que definen una entidad, es decir, determina de qué manera se almacenan, organizan y manipulan los datos.

Se nombran, indicando con un prefijo la aplicación a la que pertenecen, en singular aunque hay excepciones heredadas de las primeras versiones del módulo raíz que no cumplen esta convención. Por ejemplo, el objeto *res.users* contiene los datos correspondientes a usuarios.

Existen tres tipos de modelos:

- a) *Modelo (Model)*. Modelo estándar más común, almacenado en la BBDD.
- b) *Transitorios (TransientModel)*. Basados en la clase *models.TransientModel*. Son utilizados para almacenar datos de forma temporal que se eliminan periódicamente sin intervención del usuario.
- c) *Abstractos (AbstractModel)*. Basados en la clase *models.AbstractModel* y utilizados para definir clases abstractas que pueden ser heredadas por múltiples modelos.

En el menú *Técnico/Estructura de base de datos/Modelos* se encuentra la lista de todos los modelos disponibles en la base de datos.

Los modelos contienen campos con datos y atributos. Se pueden considerar los siguientes tipos de campos principales:

- *Campos básicos*. Contienen datos numéricos o de tipo texto como *Integer*, *Boolean*, *Char*, *Float*.
 - *Campos reservados*. Los crea Odoo automáticamente para su uso por el sistema, como *id*, *create_date*, *create_uid*, *write_date*.
 - *Campos avanzados*. Odoo requiere un campo *name* en todos los modelos. Campos como *Binary*, *Html*, *Image*, *Monetary*, *Selection*, *Date*.
 - *Campos calculados*. Son los que toman su valor mediante una función, no de la BBDD. Se trata de un campo declarado de forma estándar pero que tiene entre sus argumentos uno llamado *compute* que referencia la función de cálculo definida en otra parte del código.
 - *Campos relacionales*. Son los que definen las relaciones entre modelos vinculando los datos que contienen cada uno de ellos. En Odoo existen tres tipos:
 - *Many2one*. Relaciones “de muchos a uno”: relación unidireccional que selecciona un registro de otra tabla. Ejemplo: varios empleados pertenecen a un departamento.
 - *Many2many*. Relaciones “de muchos a muchos”: relación bidireccional. Ejemplo: los clientes pueden comprar distintos producto y cada producto puede ser adquirido por varios clientes.
 - *One2many*. Relación “de uno a muchos”: relación en un sentido que selecciona uno o varios registros de otra tabla. Ejemplo: un cliente ha efectuado varios pedidos.
- Un campo *One2many* debe tener un *Many2one* relacionado con él.

Los campos admiten *atributos*. Los más comunes son:

- *string*. Etiqueta visible.
- *default*. Valor por defecto estático o calculado.
- *required*. (True) Obligatorio.
- *help*. ToolTip de ayuda.
- *readonly*. Solo lectura.
- *index*. Si el campo se indexa en la BBDD.
- *compute*. Transforma en campo calculado.
- *store*. (True) Almacena el resultado de campos calculados.

 PARA SABER MÁS

En cuanto a la relación de clases principales de Odoo, no existe una relación oficial de modelos existentes. La mejor forma de conocerlos es utilizando pgAdmin. Una lista de los modelos más utilizados es:

Modelo	Tabla postgres	Descripción
res.partner	res_partner	Usuarios, contactos, clientes y proveedores
res.groups	res_groups	Grupos de usuarios
res.company	res_company	Compañías
ir.module.module	ir_module_module	Módulos disponibles y su estado de instalación
ir.ui.view	ir_ui_view	Tipos de vistas
ir.ui.menu	ir_ui_menu	Elementos de menú
ir.actions.act_window	ir_act_window_view	Acciones de ventana
ir.model	ir_model	Modelos usados
ir.model.fields	ir_model_fields	Campos funcionales y sus respectivas tablas.
ir.model.data	ir_model_data	Tabla relacional identificador único-nombre
ir.model.access	ir_model_access	Permisos de usuarios
ir.sequence	ir_sequence	Campos que utilizan secuencias
ir.config_parameter	ir_config_parameter	Parámetros del sistema



Actividad propuesta 6.1

Modifica el modelo producto (product.product) añadiendo un campo “procedencia” que contenga uno de los continentes.

6.4. Interfaces de usuario: menú y opciones de menú

Un *menú* es un conjunto de opciones que se muestran en cualquier interfaz de Odoo, en zonas específicas como la parte superior o el costado izquierdo. Las *opciones de menú* son cada uno de los submenús de un menú.

Las *opciones de menú* que aparecen en las pantallas de las aplicaciones de Odoo están internamente asignadas a grupos y usuarios. Lógicamente, las más básicas son accesibles por la genera-

lidad de los usuarios, pero, al igual que las *vistas*, los *menús* y sus opciones pueden ser asignados y, por tanto, solo visibles por ellos, a un usuario o grupo.

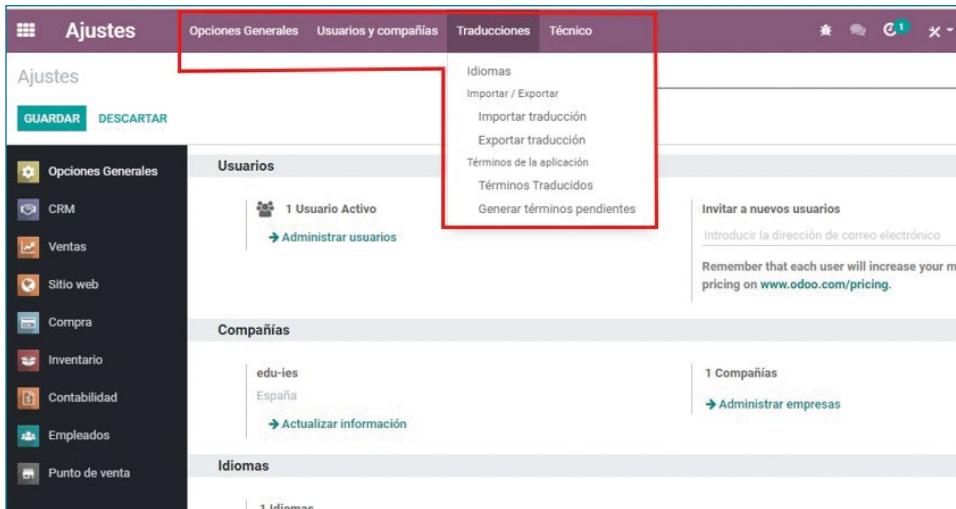


Figura 6.4
Ejemplo de menú.

Se almacenan en el modelo *ir.ui.menu*

En *Ajustes/Técnico/Interfaz de usuario/Elementos de menú* se obtiene la relación de opciones de menús disponibles, cuyos nombres responden a una jerarquía en árbol que los ubica en la Aplicación, menú y submenú u opción donde se pueden encontrar.

Elementos de menú	
Crear	Importar
Importar	
<input type="checkbox"/> Inventario/Información general	
<input type="checkbox"/> Inventario/Configuración/Ajustes	
<input type="checkbox"/> Facturación / Contabilidad/Configuración/Contabilidad analítica/Cuentas analíticas	
<input type="checkbox"/> Facturación / Contabilidad/Configuración/Ajustes	
<input type="checkbox"/> Ventas/Configuración/Ajustes	
<input type="checkbox"/> CRM/Configuración/Ajustes	
<input type="checkbox"/> CRM/Configuración/Flujo/Etapas	
<input type="checkbox"/> Punto de venta/Configuración/Ajustes	
<input type="checkbox"/> Compra/Pedidos/Solicitudes de presupuesto	
<input type="checkbox"/> Compra/Configuración/Ajustes	
<input type="checkbox"/> Ajustes/Usuarios y compañías	
<input type="checkbox"/> Ajustes/Técnico/Correo electrónico	
<input type="checkbox"/> Ajustes/Técnico/Interfaz de usuario/Elementos de menú	
<input type="checkbox"/> Ajustes/Traducciones/Idiomas	
<input type="checkbox"/> Conversaciones	
<input type="checkbox"/> Ajustes/Técnico/Teléfono / SMS/SMS	
<input type="checkbox"/> Inventario/Configuración/Gestión de almacenes	
<input type="checkbox"/> Inventario/Datos principales/Productos	
<input type="checkbox"/> Inventario/Configuración/Gestión de almacenes/Almacenes	
<input type="checkbox"/> Ajustes/Técnico/Perfiles/Términos de Trabajo	

Figura 6.5
Elementos de menú existentes.

Se pueden crear menús nuevos (se aconseja hacerlo mediante la opción de *Duplicación* hasta tener suficientes conocimientos) o, entrando en cada uno de ellos, se pueden modificar y/o asignarlos a unos usuarios en concreto.

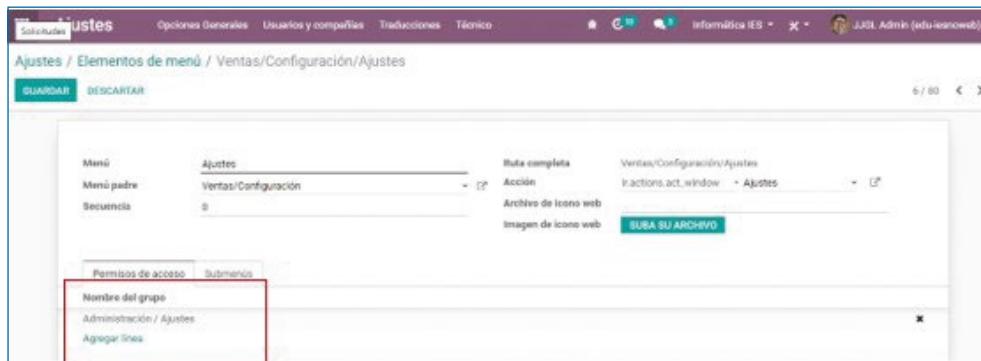


Figura 6.6
Asignación de elementos de menú.

Los *elementos de menú*:

- Si no están asignados a un grupo o usuario, son visibles por todos los usuarios.
- Si la opción *Menú padre* está vacía, aparecen en la pantalla de aplicaciones/módulos (a partir de la versión 14) y si contiene el nombre de la aplicación, en el menú superior.

Están relacionados con las *vistas* (véase el apartado siguiente 6.5). Por ejemplo, con la opción de *Editar* en el elemento de menú *Ventas/Productos/Productos* se obtiene la ventana de la figura 6.7:

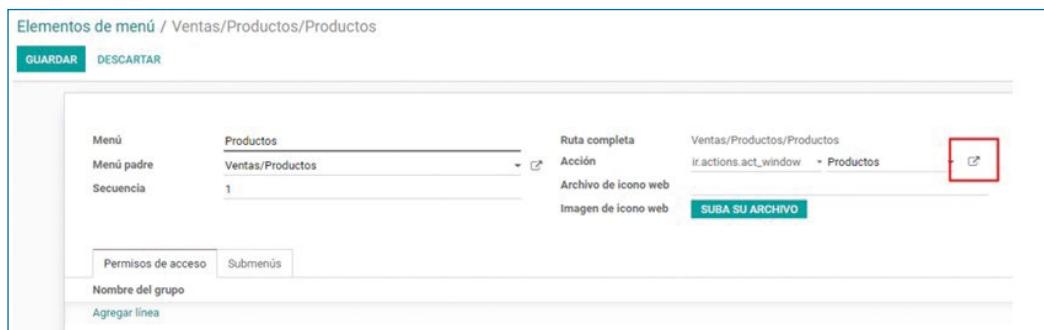


Figura 6.7
Detalle de Acción de ventana.

Y accediendo al enlace externo de la opción *Acción* de esta ventana, se obtienen las vistas relacionadas con este elemento de menú. Es decir, se obtiene la información sobre qué tipos de vistas se van a poder obtener a través de él.

En la figura 6.8 se observa que las vistas relacionadas en este caso son Kanban, Árbol, Formulario y Actividad.

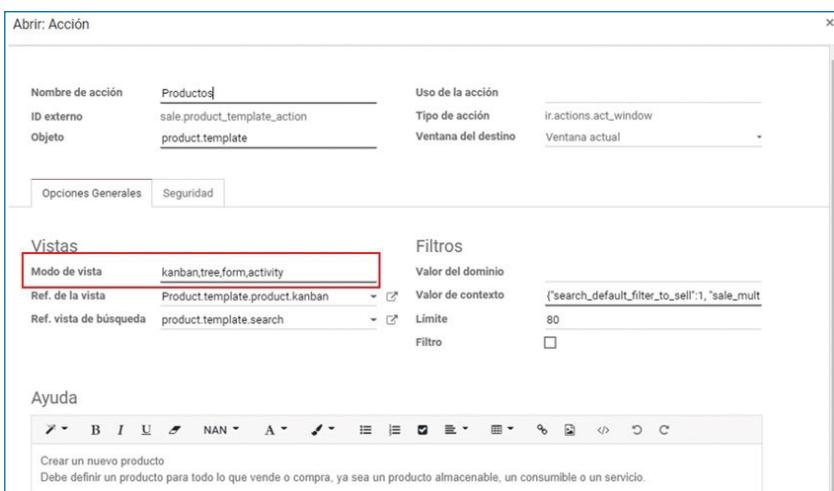


Figura 6.8
Tipos de vistas asociadas a una Acción de ventana.

Actividad propuesta 6.2



Duplica una opción de menú para que salga en otro elemento de menú distinto al suyo. Asígnala a un grupo concreto y comprueba las consecuencias de ello.

6.5. Interfaces de usuario: vistas

Una *vista* es un diseño XML que determina cómo verá el usuario los registros de un modelo. Existen tres *vistas* básicas: Kanban, Árbol y Formulario, aunque también existen *vistas* de búsqueda y *vistas* calendario, de Gantt, de Gráfico, de Actividad, etc.

Las *vistas* responden a una estructura básica común como la de la figura 6.9:

```
<record id = "MODEL_view_TYPE" model = "ir.ui.view" >
<field name = "name" > NAME </field>
<field name = "model" > MODEL </field>
<field name = "arch" type = "xml" >
<VIEW_TYPE>
<VIEW_SPECIFICATIONS />
</VIEW_TYPE>
</field>
</record>
```

Figura 6.9
Código básico de una vista.

El único campo obligatorio es el “name” que identifica a la *vista* por su nombre. Las instrucciones XML que se utilicen conformarán el aspecto de la *vista*.

Todas las pantallas de Odoo están construidas como *vista*. Una *vista* actúa sobre un modelo y especifica qué campos se mostrarán y con qué diseño gráfico. Una *vista* es el resultado de una consulta a una o más tablas de la BBDD.

En *Ajustes/Técnico/Interfaz de usuario/Vistas* se obtiene la relación de *vistas* disponibles, cuyos nombres responden a una jerarquía en árbol.

El nombre de las *vistas* responde al formato *modelo.nombre_descriptivo.extension*, donde *extension* puede ser tree, kanban, search... Como en casos anteriores, pulsando sobre una de ellas se accede al detalle de su composición y configuración.

Se muestra entonces información sobre nombre, tipo, modelo sobre el que actúa, si se trata de una *vista* base o hereda de otra *vista*..., así como la estructura XML y los permisos de acceso (para qué usuarios o grupos estará visible).

Nombre de la vista	Tipo de vista	Modelo	ID externo	Vista heredada
res.config.settings.view.form.inherit.base.setup	Formulario	res.config.settings	base_setup.res_config_settings_view_form	res.config.settings.view.form
Upload Bank Statements	Formulario	account.bank.statement.import	account_bank_statement_import.account_bank_statemen...	
account.analytic.line.form	Formulario	account.analytic.line	analytic.view_account_analytic_line_form	
account.bank.statement.form	Formulario	account.bank.statement	account.view_bank_statement_form	
account.journal.form	Formulario	account.journal	account.view_account_journal_form	
account.journal.group.form	Formulario	account.journal.group	account.view_account_journal_group_form	
account.journal.group.tree	Árbol	account.journal.group	account.view_account_journal_group_tree	
account.journal.kanban	Kanban	account.journal	account.view_account_journal_kanban	
account.journal.search	Búsqueda	account.journal	account.view_account_journal_search	
account.journal.tree	Árbol	account.journal	account.view_account_journal_tree	
account.move.line.tree_grouped	Árbol	account.move.line	account.view_move_line_tree_grouped	
calendar.event.form	Formulario	calendar.event	calendar.view_calendar_event_form	
crm.lead.kanban.lead	Kanban	crm.lead	crm.crm_case_kanban_view_leads	
crm.stage.form	Formulario	crm.stage	crm.crm_stage_form	

Figura 6.10
Relación de vistas.

En la opción de *Herramientas de desarrollo* se pueden editar las *vistas* para acceder al código xml que las construye.

Nombre de la vista	account.journal.search	
Tipo de vista	Búsqueda	
Modelo	account.journal	
Secuencia	1	
Activo	✓	
Estructura	Permisos de acceso	Vistas heredadas
<pre><?xml version="1.0"?> <search string="Search Account Journal"> <field name="name" string="Journal" /> <filter name="dashboard" string="Favorito" /> <separator/> <filter name="sales" string="Sales" /> <filter name="purchases" string="Purchases" /> <filter name="liquidity" string="Liquidity" /> <filter name="miscellaneous" string="Miscellaneous" /> <separator/> <filter name="inactive" string="Archived" /> </search></pre>		

Figura 6.11
Edición del código de una vista.

6.5.1. Herencias de vistas

Mediante la *herencia de vistas* se pueden obtener nuevas *vistas* trabajando sobre *vistas* estándar de Odoo y modificándolas.

Para crear vistas heredadas se utiliza lenguaje xpath que permite elaborar expresiones para acceder y aplicarlas a un documento XML. Mediante él se puede indicar las modificaciones que se van a incluir en la vista base de la que hereda utilizando las marcas:

```
<xpath>
...
</xpath>
```

Por ejemplo, en la pantalla de la *vista product.template.common.form* se puede observar en su estructura la *vista* de donde hereda (en este caso la vista base) y las vistas heredadas en la pestaña llamada así:

ID	Secuencia	Nombre de la vista	ID externo
1.138	16	product.template.procurement	stock.product_template_form_view_procurement_button
1.488	16	product.template.sale.order.button	sale.product_template_form_view_sale_order_button
1.585	16	product.template.purchase.button.inherit	purchase.view_product_template_purchase_buttons_from

Figura 6.12
Vistas heredadas
de la vista
product.template.product.form.

Y accediendo al detalle de una de las *vistas heredadas*, por ejemplo, de la vista *product.template.procurement*, se puede observar:

Nombre de la vista	product.template.procurement
Tipo de vista	Formulario
Modelo	product.template
Secuencia	16
Activo	■
Campo hijo	
Vista heredada	product.template.product.form
Modo de herencia de la vista	Vista de extensión
Datos del modelo	product.template.procurement
ID externo	stock.product_template_form_view_procurement_button

```
<?xml version="1.0"?>
<data>
    <header position="inside">
        <button string="Update Quantity" type="object" groups="stock.group_stock_manager" name="action_update_quantity_<br/>replenish" type="action" name="444" context="{'default_product_tmpl_id': id}" groups="stock.gro<br/>upner">
    </header>
    <div name="button_box" position="inside">
        <button type="object" name="action_open_qty" attrs="{'invisible': [('type', '!=', 'product')]} class="oe_st<br/>a<br/>at<br/>on<br/>button<br/>oe_stat_info">
            <div class="o_field_widget o_stat_info">
                <span class="o_stat_value" widget="statinfo">
                    <field name="qty_available" widget="statinfo" nolabel="1" class="mr4"/>
                </span>
            </div>
        </button>
        <button type="object" name="action_view_orderpoints" attrs="{'invisible': [('type', '!=', 'product'), ('nbr_r<br/>ules', '!=', 0)]}">
            <field name="nbr_reordering_rules" widget="statinfo"/>
        </button>
        <button string="Lot/Serial Numbers" type="object" name="action_open_product_lot" attrs="{'invisible': [(<br/>'tracki<br/>ng', '!=', 0)]}">
            <button string="Routes" type="action" name="445" attrs="{'invisible': [('type', 'not in', ['product', 'consu'])]}>
                <button string="Putaway Rules" type="object" name="action_view_related_putaway_rules" class="oe_stat_button" ic<br/>on="oe<br/>stat<br/>button">
            </button>
        </button>
    </div>
    <!-- change attrs of fields added in view_template_property_form<br/>to restrict the display for templates ...-->
    <xpath expr="/group[@name='group_lots_and_weight'] position='attributes'">
        <attribute name="attrs">{'invisible': [('type', 'not in', ['product', 'consu'])]}</attribute>
    </xpath>
    <xpath expr="/group[@name='group_lots_and_weight'] position='inside'">
        <field name="responsible_id"/>
    </xpath>
</data>
```

Figura 6.13
Tipos de herencia
y código xpath.



Actividad propuesta 6.3

Averigua las diferencias de *vista base* y *vista de extensión* en el modo de herencia de *vista*. Crea una *vista heredada* de la *vista árbol de productos*, quitando la imagen y añadiendo el precio de coste.

6.5.2. Prioridad de las vistas

Como de cada modelo se pueden crear distintos tipos de *vista*, es necesario controlar cuál de ellas se mostrará por defecto al activar un elemento de menú o un botón.

Una vez en la *vista* por defecto, esta se podrá cambiar utilizando los botones de *tipo de vista* que aparecen en la parte superior derecha de la *vista*.

Las *vistas* disponen de un campo denominado *Secuencia*. Cuando hay más de una *vista* asociada a un modelo se muestra por defecto la que tiene un valor menor en este campo, salvo que la *Acción de ventana* asociada al elemento de menú sí especifique la *vista* que debe presentar, pues la *Acción de ventana* tiene preferencia sobre la *Secuencia*.

6.5.3. Vistas complejas

Se consideran así las *vistas* que disponen de más de una solapa, es decir, *vistas* que se componen de varias páginas. En este caso, el código contenido entre marcas <page> </page> dibujará cada una de las pestañas.

```

</group>
<field name="lang" invisible="True"/>
<field name="user_id" invisible="True"/>
</sheet>
</form>
</field>
</page>
<page name="sales_purchases" string="Sales & Purchase">
<group name="container_row_2">
<group string="Sales" name="sale" priority="1">
<field name="user_id"/>
</group>
<group string="Purchase" name="purchase" priority="2">
</group>
<group name="misc" string="Misc">
<field name="ref" string="Reference"/>
<field name="company_id" groups="base.group_multi_company" options="{'no_create': True}" at="1"
<field name="industry_id" attrs="{'invisible': [('is_company', '=', False)]}" options="{'no_create': True}" at="2"/>
</group>
</page>
<page name="internal_notes" string="Internal Notes">
<field name="comment" placeholder="Internal note..."/>
</page>
</notebook>

```

Figura 6.14
Parte de código de una pestaña de una vista compleja.

Actividad propuesta 6.4



Cambia el orden de las columnas que muestra una vista de lista cualquiera. Modifícalo mediante la inclusión de una columna con el campo nuevo añadido al modelo en la actividad propuesta 6.1.

6.6. Interfaces de usuario

6.6.1. Acciones de ventana

Las *Acciones* son los métodos desde los que se van a abrir las *vistas*.

Una *Acción de ventana* contiene las instrucciones para construir la interfaz que se mostrará al pulsar un menú o botón. También se puede hablar de *Acción de menú*, pues relaciona una *Opción de menú* con la *vista* que se abre al presionarla. No siempre una *Opción de menú* lleva aparejada una *Acción*.

Son archivos XML que contienen las instrucciones para construir la nueva *vista* cuando un usuario selecciona un *elemento de menú* o un botón.

Las *acciones de ventana* se almacenan en el modelo *ir.actions.act_window* y también disponen de un procedimiento de herencia.

Aparecen relacionadas con los *elementos de menú* y los botones de las *vistas*.

Los botones y *menús* tienen asociada una acción: lo que debe ocurrir al activarlos. Es un dato necesario que forma parte del proceso de creación del *menú* o botón que indica el *modelo* sobre el que se va a actuar y la *vista* que va a construir. Los tipos de *acciones* más importantes son:

- Window*: abre una vista en una nueva ventana.
- Report*: imprime un informe.
- Wizard*: ejecuta un asistente para realizar un determinado trabajo o proceso.

Así, para crear una *acción de ventana* que abra una *vista* se debe introducir al menos los siguientes datos:

- Nombre de la acción.
- Tipo de la acción (en este caso, *ir.actions.acc_window*).
- Nombre de la vista.
- Tipo (árbol, formulario...).

6.6.2. Tableros

Es un tipo especial de *vista* Odoo que contiene otras *vistas* encajadas para proporcionar en una única pantalla información variada.

Es un cuadro de mandos que se puede diseñar con XML, pero que tiene su propio módulo llamado *Tableros*, desde el que se pueden añadir las *vistas* de cualquier tipo que se escojan en una única.

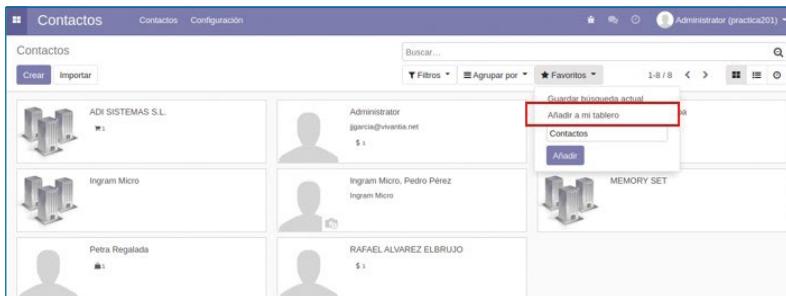


Figura 6.15
Creación de Tablero 1.

Permiten hacer el seguimiento de la marcha de la empresa o de un proceso empresarial específico de una manera visual. Se trata de generar un informe parametrizable a modo de resumen gráfico de procesos empresariales.

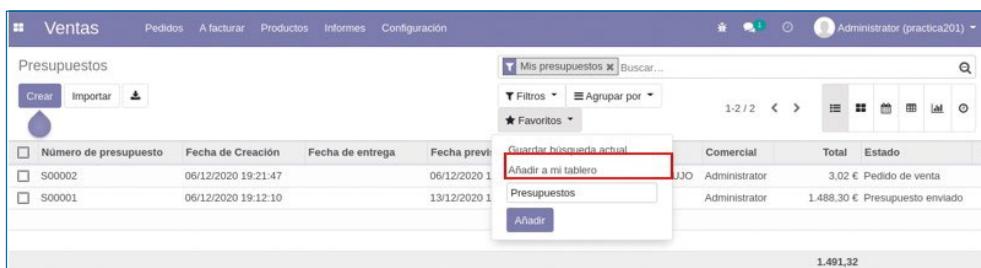


Figura 6.16
Creación de Tablero 2.

Añadiendo *vistas* a *Mi tablero* se puede obtener una visión en una única pantalla de varias *vistas*:

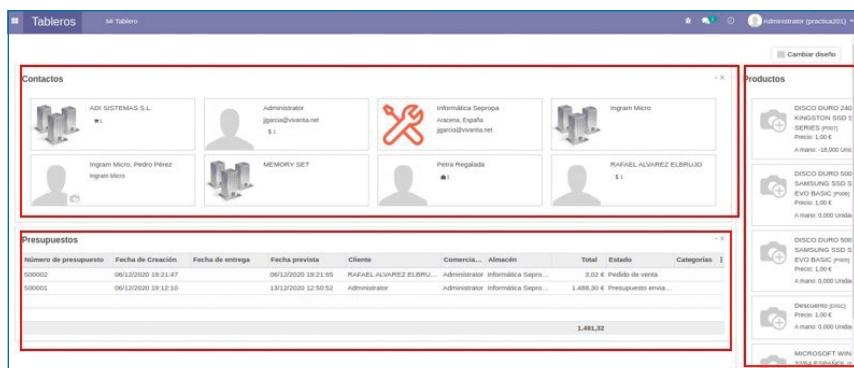


Figura 6.17
Resultado creación de Tablero.



Actividad propuesta 6.5

Añade a *Mi tablero* tres vistas de tres aplicaciones diferentes.

6.6.3. Informes. Qweb

La mayoría de las empresas necesitan realizar modificaciones en los documentos e *informes* estándar que cualquier ERP proporciona.

Un informe en Odoo es un tipo especial de vista que muestra informaciones fijas (no modificables como las vistas generales). Cada Aplicación proporciona un conjunto variado de informes que pueden ser vistos en pantalla o convertidos en ficheros .pdf.

Esta es la razón por la que, durante la instalación de Odoo, se instala también la herramienta wkhtmltopdf.

Cada informe tiene asociado el nombre de la plantilla XML con su diseño.

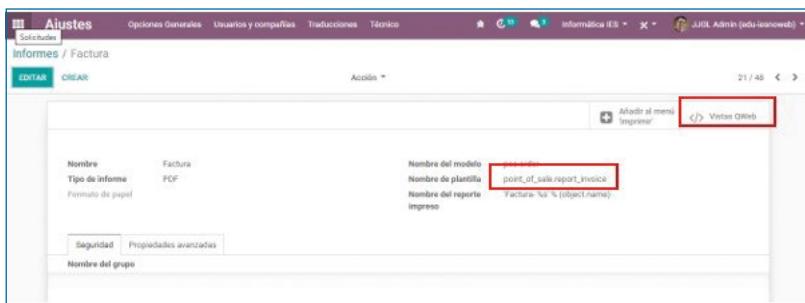


Figura 6.18
Plantilla y acceso
a Qweb.

En Odoo, es posible realizar algunos cambios sin modificar los propios documentos. Por ejemplo, es muy fácil cambiar el encabezado y pie de página que aparecen en las distintas Aplicaciones. Pero, además, Odoo dispone de un lenguaje de plantillas llamado QWeb que permite integrar datos en los informes, haciendo posible tanto modificar los informes por defecto que incluye Odoo como crear nuevos informes.

Qweb es el motor de plantillas, basado en XML, que, aunque originalmente fue introducido para mejorar el aspecto de vistas Kanban, en la actualidad lo usa Odoo para la generación de informes. La vista de tipo Qweb que los contiene por defecto es *web.html_container*.

Los informes disponen de encabezado y pie de página.

Las últimas versiones de Odoo construyen los informes como las vistas del sitio web y permiten crear páginas web dinámicas vinculadas al ERP o no

Los nombres internos de todos los informes empiezan por *report_*

Nombre	Nombre del modelo	Tipo de acción...	Nombre de plantilla	Tipo de informe	Guardar como Prefijo del Adjunto
Audiencia de Libros	account.common.journal.report_ir.actions.report	ir.actions.report	account.report_journal	PDF	
Código de Barras del Producto (PDF)	product.template	ir.actions.report	product.report_producttemplatebarcode	PDF	
Código de Barras del Producto (PDF)	product.product	ir.actions.report	product.report_productbarcode	PDF	
Código de barras de ubicación	stock.location	ir.actions.report	stock.report_location_barcode	PDF	
Código de barras del paquete (PDF)	stock.picking	ir.actions.report	stock.label_transfer_template_view_pdf	PDF	
Código de barras del paquete (PDF)	stock.quant.package	ir.actions.report	stock.report_package_barcode_email	PDF	
Código de barras del paquete (ZPL)	stock.quant.package	ir.actions.report	stock.label_package_template_view	Texto	
Código de barras del paquete (ZPL)	stock.picking	ir.actions.report	stock.label_transfer_template_view_zpl	Texto	
Código de barras del paquete con contenido	stock.quant.package	ir.actions.report	stock.report_package_barcode	PDF	
Código de barras del producto (ZPL)	product.template	ir.actions.report	stock.label_barcode_product_template_view	Texto	
Código de barras del producto (ZPL)	product.product	ir.actions.report	stock.label_barcode_product_product_view	Texto	
Embalaje del Producto (PDF)	product.packaging	ir.actions.report	product.report_packagingbarcode	PDF	
Etiqueta del Producto (PDF)	product.product	ir.actions.report	product.report_productlabel	PDF	
Etiqueta del Producto (PDF)	product.template	ir.actions.report	product.report_producttemplatelabel	PDF	
Etiqueta del producto (ZPL)	product.product	ir.actions.report	stock.label_product_product_view	Texto	
Etiqueta del producto (ZPL)	product.template	ir.actions.report	stock.label_product_template_view	Texto	
Extracto	account.bank.statement	ir.actions.report	account.report_statement	PDF	
Factura PRO-FORMA	sale.order	ir.actions.report	sale.report_saleorder_pro_forma	PDF	
Facturas	account.move	ir.actions.report	account.report_invoice_with_payments	PDF	(object.state == 'posted') and ((object.name or 'INV') replace(' ','_')) + '_original_vendor_bill.pdf'
Facturas originales	account.move	ir.actions.report	account.report_original_vendor_bill	PDF	(object.state == 'posted') and ((object.name or 'INV') replace(' ','_')) + '_original_vendor_bill.pdf'
Facturas sin pago	account.move	ir.actions.report	account.report_invoice	PDF	(object.state == 'posted') and ((object.name or 'INV') replace(' ','_')) + '_original_vendor_bill.pdf'
Guía Técnica	ir.module.module	ir.actions.report	basis.report_imodulereference	PDF	

Figura 6.19
Relación
de Informes.

Pulsando en la opción Vistas QWeb se obtiene la relación de vistas que utiliza, y cada una de ellas, en su solapa Estructura, contiene la plantilla Qweb XML.

Las plantillas de QWeb son fragmentos HTML reutilizables que se pueden insertar en otras plantillas. Se pueden diseñar bloques con los que componer vistas más complejas.

Con el modo desarrollador activo, en *Ajustes/Configuración/Técnico/Informes* se obtiene una relación de los informes estándar.

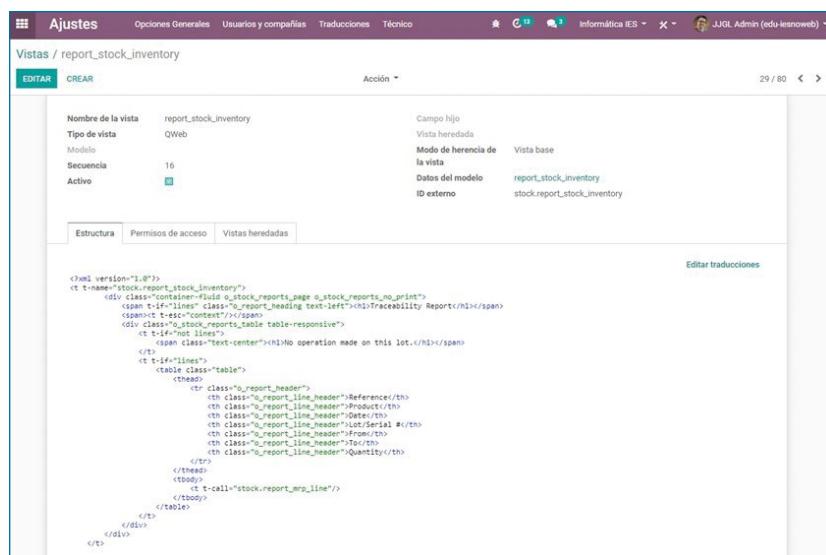


Figura 6.20 Estructura Qweb del informe.

En particular, si se hace una búsqueda de *vistas* de tipo Qweb cuyo nombre empieza por `report`, seleccionando la vista `report_stock_inventory`, se muestra en detalle la estructura Qweb:

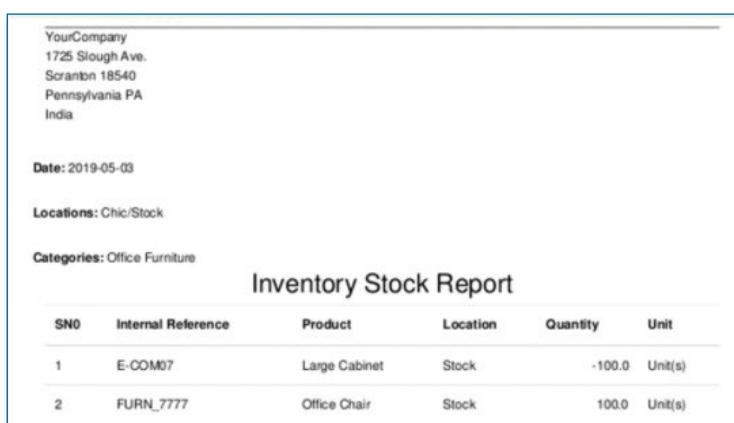


Figura 6.21
Informe de Inventario.

En el siguiente capítulo se profundizará en la configuración y creación de informes y en el motor de plantillas Qweb.

6.7. Serialización e identificadores externos

Los objetos se pueden serializar, es decir, pueden ser serializables. Se llama *serialización* al proceso de convertir un objeto en una secuencia de bytes para almacenarlo o transmitirlo a la memoria, a una base de datos o a un archivo.

Las distintas interfaces de usuario vistas hasta ahora se encuentran almacenadas en registros de las tablas internas de Odoo que utiliza la serialización para exportar e importar datos. Cada registro de una tabla de Odoo tiene un identificador único llamado “*id*” asignado secuencial y automáticamente en el momento de su creación. La etiqueta está compuesta por el nombre del módulo y el nombre de identificador separado por puntos.

Para poder hacer referencia al registro relacionado con un *id* que aún no está creado, se utilizan los *identificadores externos*, que asignan identificadores con nombre a los registros, mediante una tabla que contiene y relaciona los *id* externos con nombre, con los *id* numéricos de la BBDD. Se trata del modelo *ir.model.data*.

En la opción *Ajustes/Técnico/Secuencias e identificadores/Identificadores externos*, entre otras existe una opción de *crear*:

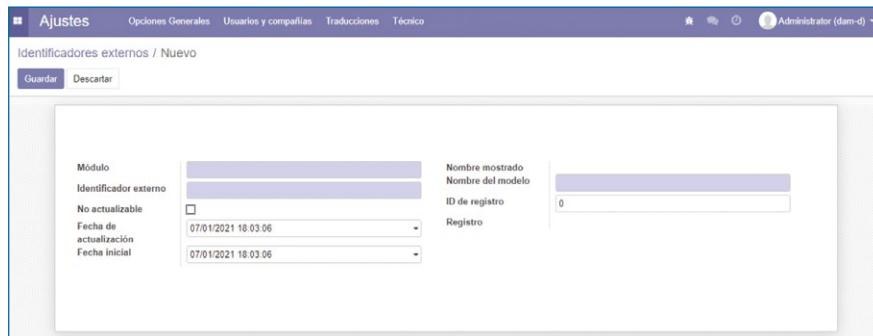


Figura 6.21
Pantalla de creación de identificadores externos.

Donde:

- Los campos *Módulo* e *Identificador externo* concatenados separados por un punto deben coincidir con el nombre de la vista.
- *Nombre del modelo* se obtiene de la url (*ir.ui.view*).
- El *Id de registro* se obtiene de la url.

Añadiendo estos datos, se actualizará el campo *id externo* de la vista utilizada para el informe.

Los identificadores externos también son fundamentales en los procesos de importación de datos pues evitan las duplicidades, realizando un proceso de actualización en vez de creación, en el caso de importaciones repetidas.

6.8. Secuencias o contadores

Una *secuencia* en Odoo no es más que un contador secuencial automático.

Los campos de secuencia permiten numerar cualquier entidad. Las *secuencias* en Odoo generan automáticamente un número consecutivo.

Las *secuencias* se componen de un prefijo, un elemento de relleno, un contador o número consecutivo y un sufijo.

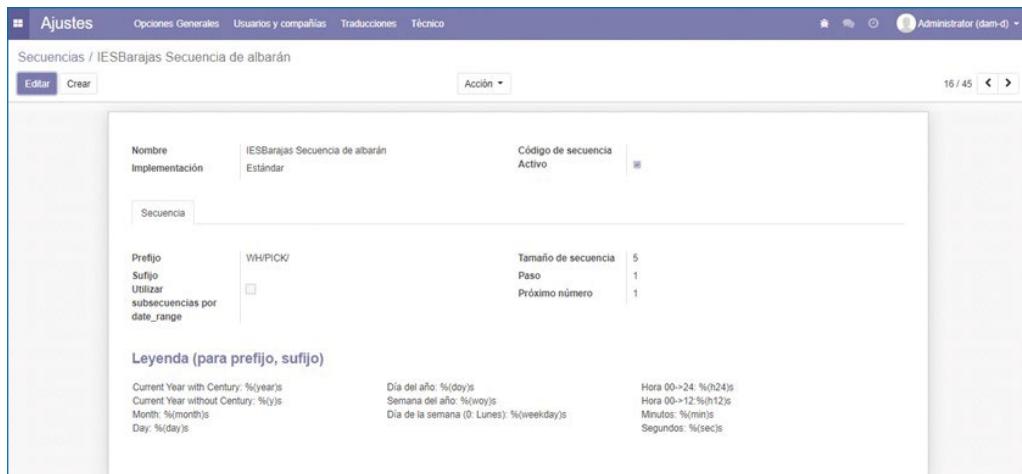


Figura 6.22
Formato de una secuencia.

Odoo crea las *secuencias* estándar para casi todos los formularios de Odoo: para presupuestos, pedidos, albaranes, asientos contables, facturas de venta, etc., que se pueden modificar para adaptarlas a las necesidades o preferencias de la empresa.

El modelo *ir.sequence* tiene asociadas las siguientes *vistas* que se muestra en la figura 6.23:

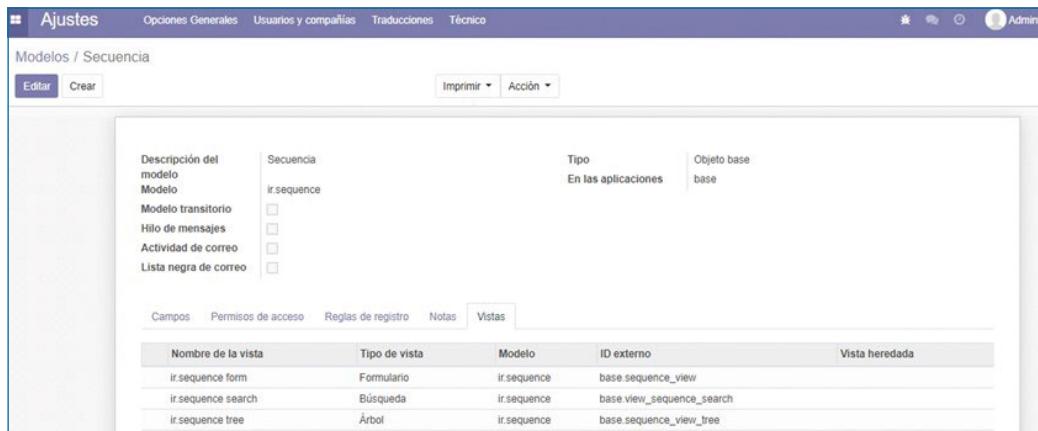


Figura 6.23
Vistas asociadas.

La *vista formulario* asociada tiene la estructura que se muestra en la figura 6.24:

Abrir: Vistas

Nombre de la vista	ir.sequence.form	Campo hijo	
Tipo de vista	Formulario	Vista heredada	
Modelo	ir.sequence	Modo de herencia de la vista	Vista base
Secuencia	16	Datos del modelo	ir.sequence.form base.sequence_view
Activo	<input checked="" type="checkbox"/>	ID externo	

Estructura Permisos de acceso Vistas heredadas

Editar traducciones

```
<?xml version="1.0"?>
<form string="Sequences">
    <sheet>
        <group>
            <group>
                <field name="name"/>
                <field name="implementation"/>
            </group>
            <group>
                <field name="code"/>
                <field name="active"/>
                <field name="company_id" groups="base.group_multi_company"/>
            </group>
        </sheet>
        <notebook>
            <page string="Sequence">
                <group>
                    <group>
                        <field name="prefix"/>
                        <field name="suffix"/>
                        <field name="use_date_range"/>
                    </group>
                    <group>
                        <field name="padding"/>
                        <field name="number_increment"/>
                        <field name="number_next_actual" string="Next Number" attrs="{'invisibility': [('use_date_range', '=', True)]}"/>
                    </group>
                </group>
                <field name="date_range_ids" attrs="{'invisibility': [('use_date_range', '=', False)]}"/>
            </page>
        </notebook>
    </form>
```

Figura 6.24

Vista formulario asociada.

Una *secuencia* puede estar asociada al nombre del registro o formar parte de su información interna.

Por ejemplo:

```
<field name="prefix">JJ</field>
<field name="padding">4</field>
```

generará un contador que empezará con JJ0001 (un prefijo JJ que aparecerá en todos los nombres y un número secuencial de 4 dígitos).



TOMA NOTA

Secuencia (*sequence*) es uno de los atributos principales de los elementos de menú y vistas y se utilizan para establecer el orden en el que se presentarán las vistas del mismo nivel. Tiene preferencia la de menor secuencia (salvo que esté especificada una acción de ventana).

6.9. Documentos y plantillas

Odoo dispone de una serie de plantillas de documentos comerciales y de correo electrónico que son editables y parametrizables según las preferencias de la empresa.

Se accede a ellas desde el menú *Ajustes/Técnico/Plantillas*.

The screenshot shows two windows of the Odoo 'Ajustes / Plantillas' module. The top window is a list of templates with columns for Nombre, Aplica a, Asunto, Dende, Para (correo electrónico), A (empresas), and Fichero del informe. One template, 'Invoice: Send by email', is highlighted with a red box and a red arrow pointing to the second window. The second window is a detailed view of this template, titled 'Invoice: Send by email'. It shows the template code: 'Asunto \${object.company_id.name} Factura (Ref \${object.name or 'n/a'})'. Below the code, there is explanatory text: 'Estimada % if object.partner_id.parent_id \${object.partner_id.name} \${object.parent_id.parent_id.name}, % else \${object.partner_id.name}, % endif'. It also includes a note: 'Se renderiza dentro de la factura \${object.name} % else factura \${object.name} % if object.invoice_origin (con referencia: \${object.invoice_origin}) % endif por un importe de \${format_amount(object.amount_total, object.currency_id)} \$if object.invoice_payment_state == "paid". Esta factura ya se encuentra pagada. % else. Consulte los detalles de pago y modo de pago en el documento. % endif'. At the bottom, it says 'En caso de requerir cualquierclarificación, por favor no dude en contactarnos.'

Figura 6.25
Edición
de una plantilla.

Es necesario indicar el objeto del que se va a extraer la información para incrustar en el texto mediante el uso de campos personalizados (variables referenciadas a alguna propiedad del objeto). En la figura anterior `$(object.company_id.name)` por ejemplo.

No hay que olvidar que una vez configurada hay que asociarle la *acción de servidor* que indique cómo y cuándo se utilizará la plantilla:

The screenshot shows the 'Ajustes / Acciones de servidor / Nuevo' interface. It displays a form for creating a new action. The 'Nombre de acción' field contains the text 'Por ejemplo, Actualizar la cantidad de pedido'. The 'Modelo' dropdown is set to 'Por ejemplo, Actualizar la cantidad de pedido'. The 'Acción a realizar' dropdown is set to 'Enviar correo electrónico'. Below these fields are 'Seguridad' and 'Nombre del grupo' buttons, and an 'Agregar línea' button. The overall interface is clean and modern, typical of Odoo's design.

Figura 6.26
Acción asociada a plantilla.

Supuestos prácticos

1. Desde la interfaz de Odoo de la máquina del supuesto práctico 3.2 se pretende añadir un campo al formulario de contactos: el vehículo que conduce. Utiliza la herencia para modificar las vistas tree, form y kanban y que aparezca este campo.
2. Desde la interfaz de Odoo de la máquina del ejercicio 3.2, crea un nuevo modelo llamado *pelicula* y agrega cinco campos: *titulo*, *director*, *protagonista_masculino*, *protagonista_femenino* y *año_de_estreno*.
 - a) Diseña la vista formulario de tal forma que aparezca el campo *titulo* en la primera línea con tipografía de mayor tamaño y el resto de los campos en dos columnas.
 - b) Crea la vista de tipo *tree* incluyendo todos los campos, con el *año_de_estreno* en la segunda columna.
 - c) Aplica permisos para que estas vistas estén disponibles solo para el grupo Administradora de la Aplicación *Punto de Venta* (TPV o PoS).

Resumen

- Odoo ERP está desarrollado en Python y utiliza PostgreSQL.
- Con XML y Qweb genera distintas interfaces de usuario e informes.
- Los cuatro conceptos básicos, objetos o entidades de la estructura de Odoo son: Modelo, Vista, Menú y Acción.
- La herencia es la propiedad mediante la que un objeto deriva de otro y extiende su funcionalidad.
- ORM realiza un mapeo entre las entidades y la base de datos.
- Con ORM el programador se puede abstraer de la base de datos y evitar la complicación de sus sentencias.
- Modelos son clases Python que se corresponden con tablas PostgreSQL.
- Existen tres tipos: Estándar, Transitorios y Abstractos.
- Los principales tipos de campos de los modelos son: básicos, reservados, avanzados, calculados, relacionales.
- Los campos relacionales pueden ser Many2one, Many2many y One2many.
- Algunos de los atributos más usados para definir campos son: string, default, required, help, readonly, compute, store.
- Las interfaces de usuario que utiliza Odoo son los menús y las vistas con sus acciones de ventana, los tableros, los informes.
- Las opciones de menú que aparecen en cada vista se almacenan en el modelo *ir.ui.menu*.
- Si no están asignados a un grupo en concreto, son visibles por todos los usuarios.
- Todas las pantallas están construidas como una vista.

- Los principales tipos de vista son: Formulario, Lista, Kanban y Búsqueda.
- También existen otros tipos: Qweb, Gantt, Gráfico, Calendario, Pivote, Cohorte, Tablero.
- Las Acciones son archivos XML que contienen instrucciones que se ejecutan al activar un botón o elemento de menú.
- Los tableros (*dashboard*) son vistas especiales que contienen otras vistas encajadas.
- Un informe es una vista, no modificable una vez creada, que proporciona datos de las tablas obtenidos mediante aplicación de uno o varios criterios.
- Qweb es un motor de plantillas utilizado también en la generación de informes.
- Una secuencia es un contador secuencial. Obbedecen a un formato prefijado y se pueden modificar.
- Odoo dispone de una gran variedad de plantillas que permiten la personalización de campos.

ACTIVIDADES DE AUTOEVALUACIÓN

1. Señala cuál no es un objeto en Odoo:

- a) res.partner
- b) res.empresa
- c) res.partner.address
- d) account.invoice

2. En el MVC de Odoo las vistas se implementan con:

- a) Python.
- b) XML.
- c) Qweb.
- d) HTML.

3. Una secuencia en Odoo es:

- a) Una sucesión de sentencias QWeb.
- b) Un sistema de prioridades de vistas.
- c) Un código de identificación de vistas.
- d) La estructura de herencias que tiene una vista.

4. Son tipos de campos de un modelo de Odoo:

- a) Reservado.
- b) Compute.
- c) String.
- d) Relacionales.

5. Los objetos sobre los que articula Odoo son:

- a) Modelos, Vistas, Menús y Elementos de menú.
- b) Vista Árbol, Vista Formulario y Vista Kanban.
- c) Modelo, Vistas, Menús y Acciones.
- d) Modelos, Vistas, Herencia, Qweb.

6. Todos los modelos de Odoo (señala todas las correctas):

- a) Se escriben en Python.
- b) Son transitorios.
- c) Extienden de la clase models.Model.
- d) Sus campos básicos no admiten atributos.

7. La tabla postgres de usuarios, contactos, clientes y proveedores es:

- a) res.partner
- b) res_partner
- c) res.users
- d) res_users

8. Para crear vistas heredadas, en Odoo se utiliza:

- a) HTML.
- b) JavaScript.
- c) Python.
- d) xPath.

9. Para describir las vistas en Odoo se utiliza:

- a) Un lenguaje de consulta de datos como SQL.
- b) Un lenguaje de consulta de datos como XML.
- c) Un lenguaje de descripción de datos como XML.
- d) Ninguna es cierta.

10. Los mecanismos de herencia en Odoo son aplicables a (señala todas las correctas):

- a) Modelos, Vistas y, en general, a la lógica de negocio.
- b) Modelos y Vistas.
- c) Modelos.
- d) Vistas.

SOLUCIONES:

1. **a** **b** **c** **d**

2. **a** **b** **c** **d**

3. **a** **b** **c** **d**

4. **a** **b** **c** **d**

5. **a** **b** **c** **d**

6. **a** **b** **c** **d**

7. **a** **b** **c** **d**

8. **a** **b** **c** **d**

9. **a** **b** **c** **d**

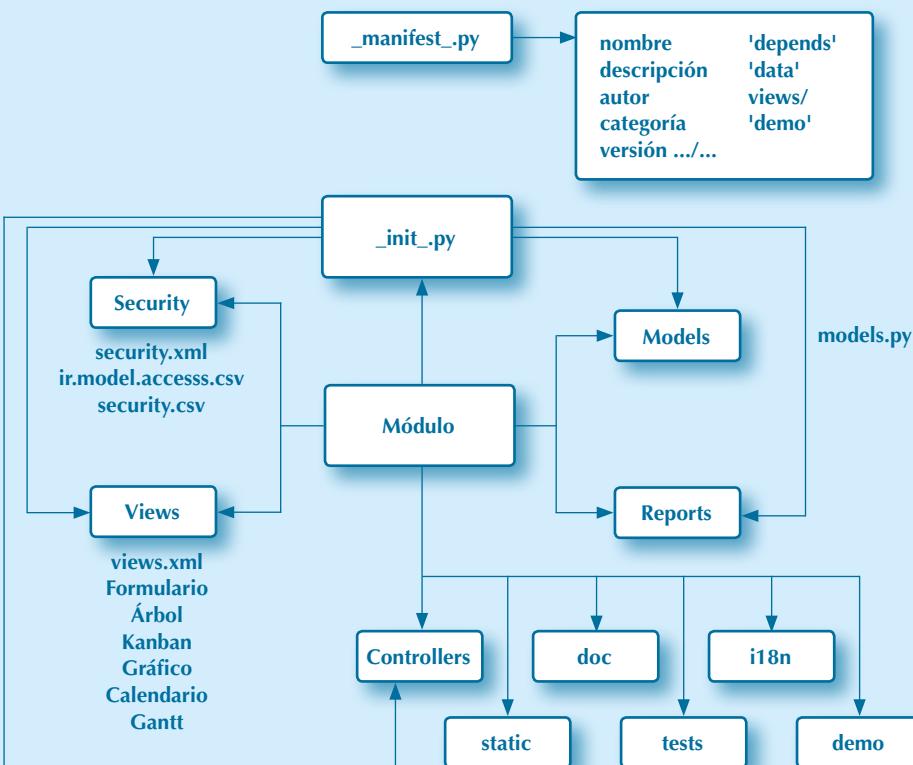
10. **a** **b** **c** **d**

Desarrollo de componentes

Objetivos

- ✓ Realizar la programación de módulos.
- ✓ Conocer la estructura que tiene un Módulo o Aplicación de Odoo.
- ✓ Saber utilizar el comando *scaffold*.
- ✓ Aprender a utilizar un editor de código en remoto.
- ✓ Crear los modelos de una aplicación.
- ✓ Diseñar los distintos tipos de vista.
- ✓ Distinguir el código xml que define cada tipo de vista.
- ✓ Conocer la relación de las vistas con los menús y acciones.
- ✓ Aprender a codificar básicamente en Python.
- ✓ Saber crear filtros calculados para las vistas de búsqueda.
- ✓ Aprender a elaborar informes.
- ✓ Aplicar restricciones de seguridad a las vistas.

Mapa conceptual



Glosario

Atom. Variante de Chromium diseñada para ser editor de textos de código abierto especializado para la codificación.

PyCharm. IDE para Python con versión Community, de código abierto que admite también JavaScript, SQL, HTML/CSS, AngularJS, Node.js y otros, y que permite depurar aplicaciones en hosts remotos.

SublimeText. Procesador de textos para crear código, con licencia comercial, heredero de vim.

Visual Studio Code. Es un editor de código optimizado para crear y depurar aplicaciones creado por Microsoft.

7.1 Introducción

En este capítulo se trabajará con ficheros *.py* (Python), *.xml* y *.csv* directamente. Para ello, se recomienda la utilización de un procesador de textos adecuado considerando que con él se modificará código, como SublimeText, Visual Studio Code o Atom o, incluso de un ide como Pycharm.

El entorno utilizado en este texto es el de un servidor Odoo instalado en una máquina virtual con sistema operativo Ubuntu server. Las conexiones entre la máquina virtual Ubuntu Server donde corre Odoo y el host (una máquina Windows 10) se han realizado mediante navegador y con PuTTY para conexiones ssh. Para explicar los procedimientos y obtener las imágenes, se ha utilizado Atom con un plugin de acceso remoto, que ha permitido trabajar directamente en el sistema de ficheros del servidor.

Dependiendo de las versiones de los distintos paquetes de software que se utilicen puede haber pequeñas modificaciones en los procedimientos.

7.2 Estructura básica de un módulo de Odoo. El comando scaffold

Se ha visto en capítulos anteriores que es posible realizar desarrollos y modificaciones utilizando la parte gráfica del sistema ERP. Se verá en este tema el desarrollo de módulos en Odoo desde cero.

Cada aplicación y cada módulo de Odoo se almacena en una carpeta propia, dentro del directorio *addons*.

El nombre técnico de la aplicación o módulo se corresponde con el nombre de la carpeta que lo alberga en el directorio *addons* (o en cualquier otro que se haya creado específicamente para guardar de forma separada módulos propios y que se haya declarado añadiendo su ruta absoluta en el fichero de configuración de Odoo).

En líneas generales un módulo de Odoo se compone de, al menos, dos archivos principales y una serie mínima de carpetas que obedecen a una estructura común:



Figura 7.1
Carpetas de addons.

CUADRO 7.1
Estructura de un módulo Odoo

<i>Ficheros</i>	<i>__init__.py</i>	Archivo principal del módulo y de carga del código Python.
	<i>__manifest__.py</i>	Archivo de manifiesto del módulo con metadatos y declaración de ficheros XML que deben ser cargados.
<i>Directorios básicos</i>	models/	Contiene archivos Python (.py) para definición de las entidades (modelos, clases y métodos). Buenas prácticas aconsejan un archivo por modelo.

[.../...]

CUADRO 7.1 (CONT.)

Directarios básicos	<code>views/</code>	Contiene los archivos .xml que generan interfaces de usuario: vistas, acciones, elementos de menú, etc.
	<code>security/</code>	Incluye archivos .xml y csv donde se especifican permisos de grupos y reglas de seguridad (listas de control de acceso). Suele contener los archivos <i>ir.model.access.csv</i> y <i>security.xml</i>
	<code>report/</code>	Guarda modelos de informes (.xml) y archivos .py y .xml relacionados.
	<code>controllers/</code>	Contiene archivos de código para controladores y rutas http del sitio web.
	<code>demo</code>	Se usa para proporcionar datos de prueba precargados en ficheros .xml (son de carga opcional).
	<code>static</code>	Para documentación para el sitio web en directorios img/, css/, js/ lib/ etc. Este directorio contiene principalmente archivos JavaScript, hojas de estilos e imágenes. No es necesario mencionarlo en el manifiesto del módulo, pero son referenciados en la plantilla web.
<i>Otros directrios</i>	<code>i18n/</code>	Contiene archivos de localización y traducción.
	<code>data/</code>	Archivos css y xml que cargan datos al sistema (si tiene contenido, se carga obligatoriamente).
	<code>wizard/</code>	Contiene vistas de tipo wizard. Reagrupa modelos transitorios
	<code>tests/</code>	Para incluir código de tests para pruebas de funcionamiento.

a) Archivo `__init__.py`

Archivo principal del módulo con instrucciones de importación: es el primer archivo que se lee en la ejecución e indica qué otros archivos Python se deben importar y ejecutar pues contiene las referencias a otros archivos .py y a los directorios donde se encuentran. Cualquier directorio que contenga archivos Python debe incluir un archivo de este tipo, aunque esté vacío.

b) Archivo `__manifest__.py`

Se trata de otro archivo que contiene los datos básicos del módulo, como la descripción, información acerca del autor, las dependencias, así como los datos relacionados con la instalación, la seguridad y la existencia o no de datos demo. Es un diccionario Python clave-valor con metadatos del módulo.

El orden en que se declaran los archivos .xml es importante. Si, por ejemplo, existieran dos archivos .xml, uno para las *vistas* y otro para *menús* y *acciones de ventana*, es necesario que primero se declare el archivo de vistas y luego el archivo donde estén las *actions*, dado que en ellas se hace referencia al *id* de las vistas.

Los módulos cuyas carpetas se encuentran en la ruta de acceso de *addons* y que son incluidos en el momento de creación de una nueva base de datos se pueden instalar directamente, y también desde la línea de comandos.

Al agregar nuevos archivos a un módulo, no hay que olvidar declararlos ya sea en el archivo *__manifest__.py* (archivos de datos) o en el archivo *__init__.py* (archivos de código) para evitar que sean omitidos y no se carguen.



PARA SABER MÁS

La nomenclatura habitualmente seguida en Odoo está basada en un nombre jerárquico separado en ramas por cada modelo.

En un módulo llamado *mimodulo* sus dos modelos principales, *modelo1* y *modelo2*, se crearán en los ficheros *mimodulo.modelo1.py* y *mimodulo.modelo2.py*. A las vistas, menús y action se las suele denominar añadiendo *view* con su tipo, *action* o menú como sufijo. Así, se llamarán:

```
mimodulo.modelo1_view_form
mimodulo.modelo1_view_tree
mimodulo.modelo1_view_kanban
mimodulo.modelo1_action_window
mimodulo.menu_root
mimodulo.modelo1_menu con padre mimodulo.menu_root
```



Consulta la página de directrices de Odoo para obtener información detallada.

7.3. Generación de un módulo

Mediante la creación manual de ficheros y directorios con la estructura someramente expuesta en el apartado 7.2 se puede generar un nuevo módulo totalmente operativo. Pero Odoo proporciona un mecanismo para crear esta estructura básica de forma automática, incluyendo en algunos de los archivos líneas ejemplo del contenido que debe tener, lo que ayuda considerablemente al programador.

7.3.1. Creación de un módulo con el comando *scaffold*

La generación de la estructura de un módulo básico se puede hacer de forma automática mediante el comando *scaffold* que, a modo de andamio, genera los archivos y carpetas mínimas necesarias para que el módulo sea reconocido como parte propia por Odoo.

Su sintaxis es:

```
$ ./odoo-bin scaffold "nombre_del_modulo" "carpeta_para_addons"/
```

Por ejemplo, para crear el módulo llamado *jjg-modulo* en la misma carpeta donde están el resto de los módulos, la sintaxis con el usuario *odoo*, sería:

```
$ ./odoo-bin scaffold jjg-modulo addons/
```

que genera el directorio *jjg-modulo* con el contenido básico especificado, como se puede ver en la figura siguiente:

```
odoo@srv10x: ~/odoo/addons/jjg-modulo
-rw-rw-r-- 1 odoo odoo 2113 Dec 7 09:19 README.md
-rw-rw-r-- 1 odoo odoo 1734 Dec 7 09:19 SECURITY.md
drwxrwxr-x 337 odoo odoo 12288 Dec 27 10:05 addons
drwxrwxr-x 3 odoo odoo 4096 Dec 7 09:20 debian
drwxrwxr-x 10 odoo odoo 4096 Dec 7 09:20 doc
drwxrwxr-x 12 odoo odoo 4096 Dec 7 10:19 odoo
-rwxrwxr-x 1 odoo odoo 180 Dec 7 09:20 odoo-bin
-rw-r--r-- 1 odoo odoo 177 Dec 20 17:47 odoo.conf
-rw-rw-r-- 1 odoo odoo 1358 Dec 7 09:20 requirements.txt
drwxrwxr-x 4 odoo odoo 4096 Dec 7 09:20 setup
-rw-rw-r-- 1 odoo odoo 876 Dec 7 09:20 setup.cfg
-rw-rw-r-- 1 odoo odoo 1698 Dec 7 09:20 setup.py
odoo@srv10x:~/odoo$ ./odoo-bin scaffold jjg-modulo addons/
odoo@srv10x:~/odoo$ cd addons/jjg-modulo/
odoo@srv10x:~/odoo/addons/jjg-modulo$ ls -l
total 20
-rw-rw-r-- 1 odoo odoo 71 Dec 27 10:08 __init__.py
-rw-rw-r-- 1 odoo odoo 916 Dec 27 10:08 __manifest__.py
drwxrwxr-x 2 odoo odoo 4096 Dec 27 10:08 controllers
drwxrwxr-x 2 odoo odoo 4096 Dec 27 10:08 demo
drwxrwxr-x 2 odoo odoo 4096 Dec 27 10:08 models
drwxrwxr-x 2 odoo odoo 4096 Dec 27 10:08 security
drwxrwxr-x 2 odoo odoo 4096 Dec 27 10:08 views
odoo@srv10x:~/odoo/addons/jjg-modulo$
```

Figura 7.2
Resultado del comando *scaffold*.

Como ya se ha comentado, en cada uno de los ficheros que se generan se incluye, mediante líneas comentadas, un ejemplo del contenido que debe tener el fichero.

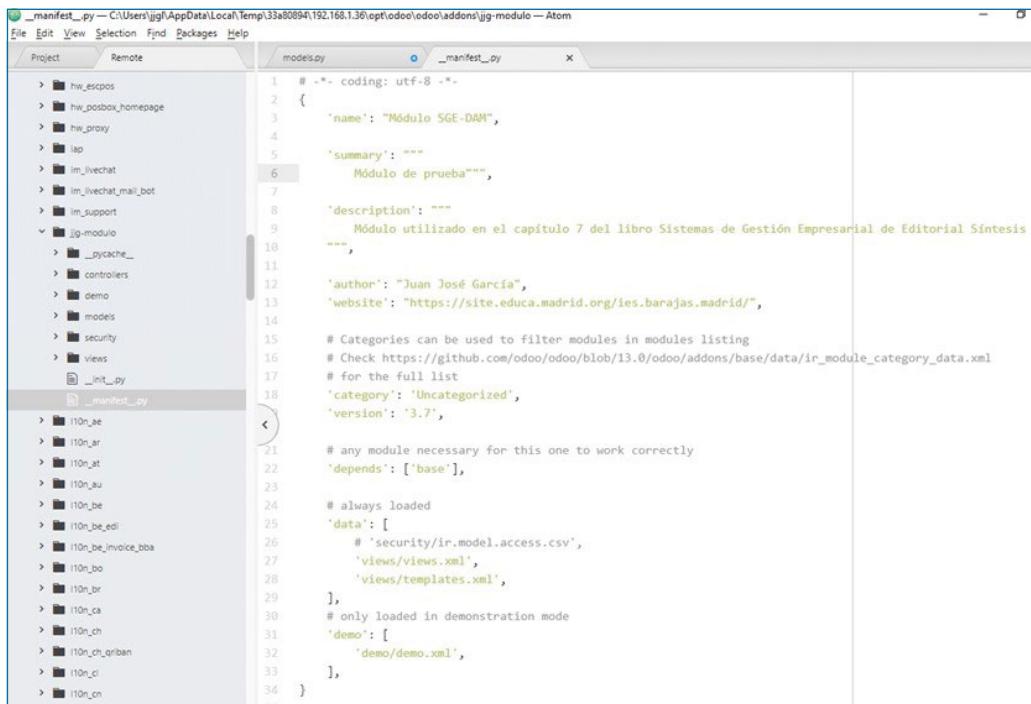
En *__init__.py* se incluyen las carpetas en las que debe buscar Python los archivos *.py* que se van a ejecutar.

```
__init__.py — C:\Users\jjg\AppData\Local\Temp\33a80894\192.168.1.36\opt\odoo\odoo\addons\jjg-modulo — Atom
File Edit View Selection Find Packages Help
Project Remote __manifest__.py × __init__.py ×
> hw_escpos
> hw_posbox_homepage
> hw_proxy
> iap
> im_livechat
> im_livechat_mail_bot
> im_support
> jjg-modulo
> __pycache__
> controllers
> demo
> models
> security
> views
__init__.py
__manifest__.py
```

```
1 # -*- coding: utf-8 -*-
2
3 from . import controllers
4 from . import models
```

Figura 7.3
Contenido de *__init__.py*.

En *__manifest__.py* se pueden incluir los datos básicos del módulo, que aparecerán en la caja Kanban del mismo.



```

manifest__py -- C:\Users\jjg\AppData\Local\Temp\33a80894\192.168.1.36\opt\odoo\odoo\addons\jjg-modulo -- Atom
File Edit View Selection Find Packages Help
Project Remote models.py __manifest__.py
1  # -*- coding: utf-8 -*-
2 {
3     'name': "Módulo SGE-DAM",
4
5     'summary': """
6         Módulo de prueba""",
7
8     'description': """
9         Módulo utilizado en el capítulo 7 del libro Sistemas de Gestión Empresarial de Editorial Síntesis
10    """,
11
12     'author': "Juan José García",
13     'website': "https://site.educa.madrid.org/ies.barajas.madrid/",
14
15     # Categories can be used to filter modules in modules listing
16     # Check https://github.com/odoo/odoo/blob/13.0/odoo/addons/base/data/ir_module_category_data.xml
17     # for the full list
18     'category': 'Uncategorized',
19     'version': '3.7',
20
21     # any module necessary for this one to work correctly
22     'depends': ['base'],
23
24     # always loaded
25     'data': [
26         # 'security/ir.model.access.csv',
27         'views/views.xml',
28         'views/templates.xml',
29     ],
30     # only loaded in demonstration mode
31     'demo': [
32         'demo/demo.xml',
33     ],
34 }

```

Figura 7.4
Contenido de __manifest__.py ya modificado.



Figura 7.5
El Módulo
creado aparece
en la lista de
Aplicaciones.

Con este simple procedimiento, si se actualiza la lista de Aplicaciones de la instancia de Odoo utilizada, se obtiene la caja Kanban del módulo creado, lógicamente, sin ninguna funcionalidad aún, pero con los datos introducidos en el __manifest__.py

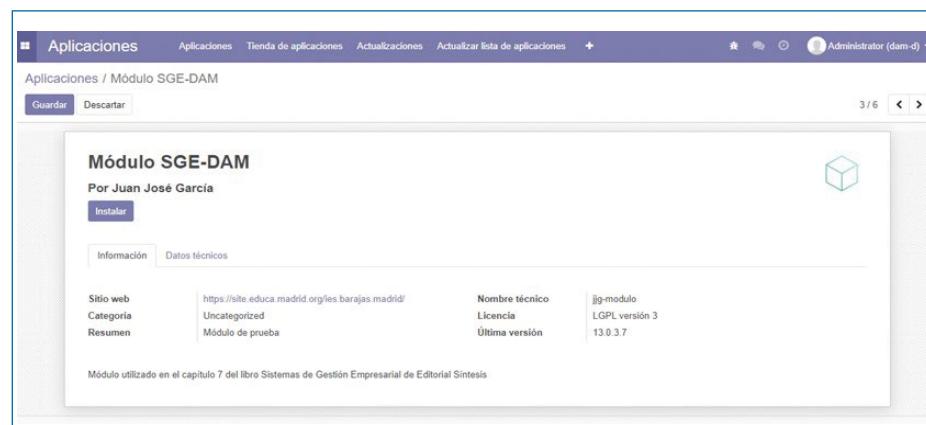


Figura 7.6
Detalle de la
caja Kanban del
módulo creado.



PARA SABER MÁS

En entorno Windows, la creación de nuevos módulos mediante el comando scaffold debe seguir el siguiente procedimiento:

1. Abrir cmd como administrador
2. Ejecutar:
C:\Windows\system32> "C:\Archivos de programa\Odoo XXXX\python\python.exe" "C:\Archivos de programa\Odoo XXXX\server\odoo-bin" scaffold Mimodulo "C:\Archivos de programa\Odoo XXXX\server\odoo\addons"

donde XXXX hace referencia a la versión de Odoo instalada

7.3.2. Creación del modelo

Para definir el modelo o modelos que usará el módulo, se utiliza el fichero *models.py* (puede haber varios) de la carpeta *models*.

Cada objeto Odoo se corresponde con una clase Python. Siempre extienden de la clase *Model* y es buena práctica nombrarlas incluyendo el nombre del módulo.

Los campos que como mínimo debe contener son:

- *_name*: las buenas prácticas recomiendan que siga la sintaxis: *mimodulo.modelo1*
- *name*: de tipo *char* con atributo *string* con lo que se quiere instanciar. Siempre debe existir este campo *name* que se utilizará en las vistas de búsqueda y para establecer relaciones entre objetos.

Además, se enumerarán con sus respectivos tipos y atributos el resto de campos que vayan configurando el modelo en creación.

```

models.py
1  # -*- coding: utf-8 -*-
2
3  # from odoo import models, fields, api
4
5
6  class modulo(models.Model):
7      _name = 'modulo.modulo'
8      _description = 'modulo.modulo'
9
10     name = fields.Char()
11     value = fields.Integer()
12     value2 = fields.Float(compute="_value_pc", store=True)
13     description = fields.Text()
14
15     @api.depends('value')
16     def _value_pc(self):
17         for record in self:
18             record.value2 = float(record.value) / 100
19

```

Figura 7.7
Estado inicial de
models.py

En el caso de la existencia de varios modelos, es fundamental establecer claramente las relaciones entre ellos, utilizando los campos *fields.Many2one*, *fields.Many2many* o *fields.One2many* (en este caso especificando con qué campo del otro modelo).

Respecto a los campos calculados, se construyen mediante un *método* dentro del objeto, que se encarga de hacer un cálculo con un atributo *compute*, que es la función privada que se usará para efectuarlo. La forma de definirlo es mediante:

```
@api.depends('campo1', 'campo2') Anotación (se ejecuta cuando cambien campo1 o campo2).
def _campocalculado(self): Se itera por si al método se le llama con varios
    for r in self: registros.
        r.campocalculado=r.campo1"operación"r.campo2
```

7.3.3. Diseño de las vistas básicas

Se describen en el archivo *views.xml* de la carpeta *views*. En él se definen tanto las vistas como las opciones de menú y las acciones de ventana relacionadas.

Las vistas son objetos de tipo *record* de la tabla *ir.ui.view*. Su identificador (necesario) sigue la nomenclatura *nombredelmodulo.nombredelmodelo_tipo* donde tipo puede ser form, tree, kanban... (es importante que solo se utilice un único punto en el nombre).

Las líneas que debe incluir son:

```
<record model="ir.ui.view" id="nombredelmodulo.nombredelmodelo_tipo">
```

- name: *nombredelmodulo.nombredelmodelo.tipo* (su nombre en el atributo id cambiando el guion bajo por punto).
- model: objeto sobre el que aplica.
- arch: tipo de archivo.

Por ejemplo:

```
<record model="ir.ui.view" id="nombredelmodulo.nombredelobjeto_form">
    <field name="name">nombredelmodulo.nombredelobjeto.form</field>
    <field name="model">nombredelmodulo.nombredelobjeto</field>
    <field name="arch" type="xml">
```

A continuación, siguen una estructura basada en lo siguiente, según el tipo de vista:

- *Árbol*

```
<tree>
    <field name="name"/>
    <field name="value"/>
    ...
</tree>
```

- *Formulario*

```
<form>
    <group colspan="x" col="y">
        <field name="name"/>
        <field name="value"/>
        ...
    </group>
</form>
```

Con el uso de *group* y *colspan* se consigue un aspecto similar al utilizado habitualmente por Odoo en sus vistas de tipo formulario.

- *Búsqueda*

Por defecto, existe una vista de búsqueda por el campo *name*. Se pueden crear procedimientos de búsqueda por otros campos del modelo mediante la inclusión de código entre etiquetas

```
<search>
    <field name="name" string="" />
    <field name="value" string="" />
    ...
</search>
```

Además, se puede añadir un *filtro*, concepto mediante el que se aplica una determinada condición sobre un campo.

Los filtros se construyen definiendo un *dominio* (*domain*) que es la tripleta de la *condición* formada por *campo*, *operador* y *valor*.

Los operadores utilizados en ficheros .xml obedecen la siguiente sintaxis:

CUADRO 7.2
Operadores para filtros

Operador	En ficheros .py	En ficheros .xml
igual	=	=
distinto	!=	<>;
mayor	>	>;
mayor o igual	>=	>=
menor	<	<;
menor o igual	<=	<=

7.3.4. Acciones de ventana

El siguiente paso es especificar las acciones de ventana, que responden a la siguiente estructura:

```
<record model="ir.actions.act_window" id="nombredelmodulo.action_window">
    <field name="name">nombredelmodulo window</field>
    <field name="res_model">nombredelmodulo.nombredelmodelo</field>
```

```
<field name="view_mode">tree,form</field>
</record>
```

7.3.5. Opciones de menú

Por último, los elementos de menú, que responden a una jerarquía de tipo árbol, y que especifican las acciones de ventana mediante el atributo "action":

```
<menuitem name="nombredelmodulo"
id="nombredelmodulo.menu_root"/>

<menuitem name="nombremenu 1"
id="nombredelmodulo.nombremenu_1"
parent="nombredelmodulo.menu_root"
action="nombredelmodulo.action_window "/>

<menuitem name="nombremenu 2"
id="nombredelmodulo.nombremenu_2"
parent="nombredelmodulo.menu_root"
action="nombredelmodulo.action_window "/>
```

7.4. Generación de informes

Los informes en Odoo se elaboran en ficheros xml utilizando Qweb. Para crear nuevos informes, es una buena práctica guardarlos en un directorio específico al que se llama normalmente *reports*.

Una vez creados, se deben declarar en apartado 'data' del fichero *__manifest__.py* y añadir las dependencias necesarias en el mismo fichero, en el apartado 'depends'.

La estructura de un informe se compone de dos partes:

- <record/>, donde se crea un registro de tipo ir.action.report:
`<record id="nombre_informe" model="ir.actions.report">`
y se definen las características básicas generales del informe
`<record id="nombre" model="ir.actions.report">`
model= modelo sobre el que actúa
report_name= modulo.template id de la parte <template/>
report_file= modulo.template id de la parte <template/>
report_type="qweb-tipo" donde tipo puede ser html o pdf
- <template/>, que especifica los campos que incluye y las plantillas utilizadas para generar el informe.

La vista de tipo Qweb que la contiene por defecto es *web.html_container*.

Además, si se utiliza *web.internal_layout* se generará un pdf con encabezado y pie de página de mínimos (con paginación, fecha de impresión y nombre de la compañía), mientras que con *web.external_layout* se obtiene un encabezado y pie de página con la información de contacto completa y el logotipo disponible de la compañía. Como ambos layout son en sí mismos vistas, se puede utilizar la herencia usando el *inherit_id* correspondiente.

<template id= que debe coincidir con report_name y report_file de <record/> salvo que no lleva el nombre del módulo>
 <t t-call contenedor (por ejemplo web.html_container)>
 <t t-foreach iteración por modelo>
 <t t-call llamada a formato por ejemplo web.internal_layout >

Entre etiquetas <div> ... </div> los campos del modelo que se quieran obtener en el informe. Además, es imprescindible ajustar todo el contenido en un elemento <page>. (<div class="page"> ... </div>)

Se pueden crear en uno o varios ficheros reports.xml e incluso se pueden separar ambas partes <record/> y <template/> en ficheros distintos.

Una vez creado, el informe se obtendrá a través de la opción *Imprimir* de la ventana del formulario (vista formulario del modelo objeto del informe). En la url de dicha ventana se puede encontrar el modelo sobre el que está construido.

Es necesario comprobar que alguno de los dos parámetros de configuración *web.base.url* o *report.url* son accesibles desde la instancia de Odoo, pues si no, el informe tardará excesivo tiempo en generarse.

En versiones anteriores de Odoo, se utilizaba una etiqueta <report/> para registrar un informe. Pero a partir de la versión 14, está en desuso y es necesario crear un registro de *ir.actions.report* con la nueva etiqueta <record/>.

La etiqueta <record/> todavía se admite en Odoo 14 para asegurar la compatibilidad con informes realizados en versiones anteriores, pero su uso generará una advertencia que será incluida en el registro (*odoo.log*).



PARA SABER MÁS

En Qweb, cada directiva de plantilla se denomina con un prefijo t- y como marcador de posición se utiliza <t>.

Campo record

Es el nodo principal. Tiene la siguiente estructura:

```
<odoor>
  <record id="nombre_interno_informe" model="ir.actions.report">
    <field name="name">Nombre que aparece en "Imprimir"</field>
    <field name="model">modelo sobre el que actúa</field>
    <field name="report_type">qweb-pdf o qweb-html</field>
    <field name="report_name">nombremodulo.nombredelplantilla</field>
    <field name="report_file">nombremodulo.nombredelplantilla</field>
    <field name="binding_model_id" ref="model_modelosobreelqueactua"/>
    <field name="binding_type">report</field>
  </record>
</odoor>
```

Campo template

Indica el esquema de la plantilla. Un informe puede tener varias plantillas.

```
<odoo>
    <template id=" nombredeplantilla ">
        <t t-call="web.html_container">           (contenedor)
            <t t-foreach="docs" t-as="doc">         (iteración por modelo)
                <t t-call="web.external_layout">
                    <div class="header"></div>
                    <div class="page"></div>           |   (se repiten en todas
                                                las páginas)
                    <div class="footer"></div>           |
                </t>
            </t>
            <t/>
        </template>
    <odoo/>
```

Algunas directivas de Qweb

t-set	Declara una variable
t-field	Para mostrar un campo
t-if	Directiva condicional que permite evaluar una expresión
t-foreach	Directiva de iteración.
t-call	Directiva para llamar a otras plantillas de informe.

Una descripción completa de Qweb se encuentra en QWeb-Odoo documentation.

7.5. Permisos del módulo

Se trabaja creando un fichero *security.xml* y modificando el *ir.model.access.csv* de la carpeta *security*.

Hay que añadir en el fichero xml una línea (*record*) por cada uno de los roles posibles (grupos), especificando en el fichero csv los permisos que va a tener.

```
<odoo>
    <data>
        <record id="nombredelmodulo.categoriadeusuario model="res.groups">
            <field name="name"> Módulo / Administradora</field>
            <field name="name"> Módulo / Usuario</field>
            ...
        </record>
    </data>
</odoo>
```

El fichero security.csv tiene la cabecera:

id, name, model_id:id, group_id:id, perm_read, perm_write, perm_create, perm_unlink

Donde:

- *id*: id de la regla
- *name*: nombre para la regla
- *model_id: id*: *model_tabla_sobre_la_que_aplica*
- *group_id: id*: grupo al que aplica

A continuación, cuatro valores 1 y 0 (los campos perm_read, perm_write, perm_create, perm_unlink se ponen a 1 o 0 según se concede o no ese permiso en particular).

Se debe añadir una línea por cada rol añadido en el *security.xml* creado y por cada modelo.

Después, hay que incluir ambos en el *__manifest__.py* (al principio del apartado ‘data’, primero el *security.xml* y después el *ir.model.access.csv*).

Y posteriormente, añadir en el fichero (o ficheros) *views.xml* el atributo *groups=""* especificando qué grupo tendrá asignada cada vista y, por tanto, quién la verá y quién no.

www

Recurso web

Cómo construir un módulo (página oficial de Odoo).



7.6. Creación paso a paso de un módulo

Se pretende crear un módulo llamado *lapeliculera* que permita gestionar una particular videoteca en la que se guardará cierta información sobre dos objetos: películas y el género cinematográfico en el que se encuadra.

Sobre el primer objeto, se almacenarán los siguientes datos: título, director, industria (con los posibles valores Hollywood, Europea, Bollywood y Otras), si es color o blanco y negro, la duración en minutos y una sinopsis. Por otra parte, el género puede tomar los siguientes valores: Acción, Comedia, Musical, Drama, Aventura, Western, Terror o Suspense.

Por tanto, se trabajará con dos modelos. Los modelos constituyen la base del módulo y se corresponden con tablas de una base de datos PostgreSQL. Es conocido ya que todos los modelos en Odoo extienden de la clase *models.Model* y están definidas en uno o varios archivos *models.py* guardados en el directorio *models*.

Cada uno de los modelos es una clase Python que hay que definir en *models.py*

Al abrir por primera vez el fichero *models.py* aparece una serie de líneas comentadas que dan una idea de lo que se puede incluir de forma elemental:

```

1 # -*- coding: utf-8 -*-
2
3 from odoo import models, fields, api
4
5
6 # class jjg-modulo(models.Model):
7 #     _name = 'jjg-modulo.jjg-modulo'
8 #     _description = 'jjg-modulo.jjg-modulo'
9
10 #     name = fields.Char()
11 #     value = fields.Integer()
12 #     value2 = fields.Float(compute="_value_pc", store=True)
13 #     description = fields.Text()
14 #
15 #     @api.depends('value')
16 #     def _value_pc(self):
17 #         for record in self:
18 #             record.value2 = float(record.value) / 100

```

Figura 7.8
Estado inicial
de models.py.

TEN EN CUENTA

- ✓ Es importante distinguir entre: `_name` que es el código interno que se le va a asignar en la aplicación general, y `name` que será usado en relaciones entre modelos y en búsquedas.

De acuerdo con el objetivo perseguido, unos posibles modelos podrían quedar de la siguiente manera:

```

1 # -*- coding: utf-8 -*-
2
3 from odoo import models, fields, api
4
5 class lapeliculera_pelicula(models.Model):
6     _name = 'lapeliculera.pelicula'
7     _description = 'Película'
8     name = fields.Char(string="Título", required=True, help="Introduce el título de la película")
9     director = fields.Char(string="Director", required=True, help="Introduce el director")
10    color = fields.Boolean(string="Color")
11    duracion = fields.Integer(string="Duración en minutos")
12    industria = fields.Selection([('0', 'Hollywood'), ('1', 'Europea'), ('2', 'Bollywood'), ('3', 'Otras')], default="1", string="Industria")
13    fecha = fields.Date(string="Fecha de estreno en España")
14    sinopsis = fields.Text(string="Sinopsis")
15    # Una Película debe pertenecer a un Género. No se indica campo por lo que por defecto cogerá name
16    genero = fields.Many2one("lapeliculera.genero", string="Género", required=True)
17
18 class lapeliculera_genero(models.Model):
19     _name = 'lapeliculera.genero'
20     _description = 'Género cinematográfico'
21     name = fields.Char(string="Género", required=True, help="Introduce el género cinematográfico")
22     comentario = fields.Text(string="Comentarios")
23     # Cada Género puede contener varias películas. En el caso One2many hay que especificar el campo de relación
24     pelicula = fields.One2many("lapeliculera.pelicula", "genero", string="Películas")
25     #
26     #     @api.depends('value')
27     #     def _value_pc(self):
28     #         for record in self:
29     #             record.value2 = float(record.value) / 100

```

Figura 7.9
Vista de models.py.

En este ejemplo se han añadido diversos campos de tipos variados que se han completado con algunos atributos. Se recomienda profundizar posteriormente tanto añadiendo campos de otros tipos como atributos.

Respecto a los campos relacionales entre modelos, se ha considerado que una película pertenece a un género y que un género puede contener varias películas. Cuando la relación es Many2one no hace falta especificar el campo de relación pues la realiza, por defecto, con `name`.



PARA SABER MÁS

El atributo `ondelete`, utilizado en las relaciones Many2one, define lo que pasa cuando el registro relacionado es eliminado. De forma predeterminada está fijado como `set null`, lo que significa que al ser eliminado el registro relacionado se fija a un valor vacío.

Los valores posibles son: `set null`, `restrict` y `cascade`.

Después de reiniciar el servicio de Odoo (imprescindible siempre que se modifica un fichero de Python) y de instalar el módulo, se pueden encontrar los dos modelos creados en la relación de modelos del ERP que se obtiene en *Ajustes/Modelos*:

Figura 7.10
Modelos creados.

El siguiente paso de la creación del módulo *lapeliculera* será la confección de Vistas asociadas a ambos modelos. También en este caso, al editar el archivo *views.xml* de la carpeta *views* se obtiene código ejemplo que sirve de base al programador.

Como se ha visto en el capítulo 6, hay que relacionar los modelos con las vistas y añadir los elementos de menú y las acciones de ventana. Es en este fichero (y otros ficheros como este que eventualmente puedan añadirse) donde se relacionan.

Toda vista es un objeto de tipo *record* basado en *ir.ui.view* que se define únicamente mediante un *id* que incluye el nombre de módulo, el del modelo y su tipo. Por ejemplo, *lapeliculera.pelicula_tree* será una vista de tipo árbol (también llamado lista) del modelo *pelicula* del módulo *lapeliculera*.

Un código básico para la vista lista del modelo *genero* puede ser el de la figura 7.11:

```

1 <odoo>
2   <data>
3     <record model="ir.ui.view" id="lapeliculera.genero_tree">
4       <field name="name">lapeliculera.genero.tree</field>
5       <field name="model">lapeliculera.genero</field>
6       <field name="arch" type="xml">
7         <tree>
8           <field name="name"/>
9           <field name="comentario"/>
10      </tree>
11    </field>
12  </record>
13

```

Figura 7.11
Código básico vista tree.

Y para la vista formulario, el de la figura 7.12 (en este caso se ha añadido código para que, a continuación de la vista del modelo *genero*, en la parte inferior de la vista, aparezca una lista con las películas incluidas en este género).

Figura 7.12
Código básico vista formulario.

```

14 <record model="ir.ui.view" id="lapeliculera.genero_form">
15   <field name="name">lapeliculera.genero.form</field>
16   <field name="model">lapeliculera.genero</field>
17   <field name="arch" type="xml">
18     <form>
19       <group colspan="2" col="2">
20         <field name="name"/>
21         <field name="comentario"/>
22       </group>
23     </form>
24   </field>
25 </record>
```

Repitiendo el proceso para las vistas de tipo árbol y formulario del objeto *pelicula*, se tiene la vista 7.13:

Figura 7.13
Vistas del modelo película.

```

35 <record model="ir.ui.view" id="lapeliculera.pelicula_tree">
36   <field name="name">lapeliculera.pelicula.tree</field>
37   <field name="model">lapeliculera.pelicula</field>
38   <field name="arch" type="xml">
39     <tree>
40       <field name="name"/>
41       <field name="director"/>
42       <field name="duration"/>
43     </tree>
44   </field>
45 </record>
46
47 <record model="ir.ui.view" id="lapeliculera.pelicula_form">
48   <field name="name">lapeliculera.pelicula.form</field>
49   <field name="model">lapeliculera.pelicula</field>
50   <field name="arch" type="xml">
51     <form>
52       <group colspan="2" col="2">
53         <field name="name"/>
54         <field name="director"/>
55         <field name="duration"/>
56         <field name="industria"/>
57         <field name="fecha"/>
58         <field name="genero"/>
59       </group>
60     </form>
61   </field>
62 </record>
63
```

Lo siguiente será, tal y como recuerdan las líneas comentadas del archivo que se está editando, las *acciones de ventana* de cada modelo:

Figura 7.14
Acciones de ventana.

```

4 <record model="ir.actions.act_window" id="lapeliculera.genero_action_window">
5   <field name="name">lapeliculera.genero.action_window</field>
6   <field name="res_model">lapeliculera.genero</field>
7   <field name="view_mode">tree,form</field>
8 </record>
9
10 <record model="ir.actions.act_window" id="lapeliculera.pelicula_action_window">
11   <field name="name">lapeliculera.pelicula.action_window</field>
12   <field name="res_model">lapeliculera.pelicula</field>
13   <field name="view_mode">tree,form</field>
14 </record>
15
```

Y, por último, los *elementos de menú* que se observan en la figura 7.15:

```

66     <menuitem name="LaPeliculara" id="lapeliculera.menu_root"/>
67
68     <menuitem name="Género" id="lapeliculera.genero_menu" parent="lapeliculera.menu_root" action="lapeliculera.genero_action_window"/>
69     <menuitem name="Película" id="lapeliculera.pelicula_menu" parent="lapeliculera.menu_root" action="lapeliculera.pelicula_action_window"/>
70
71     </data>
72 </odoo>
73

```

Figura 7.15

Elementos de menú.

Si ahora se instala el módulo *lapeliculera* se obtienen las pantallas de las figuras 7.16 y 7.17.

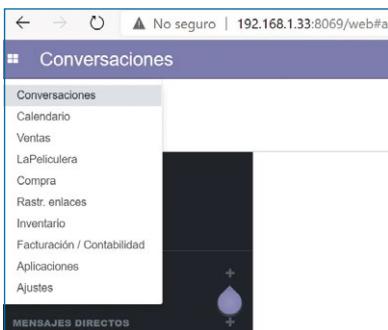


Figura 7.16

Módulo instalado.

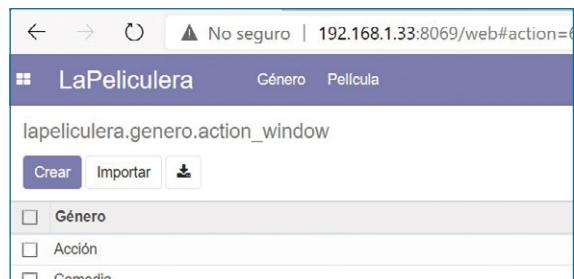


Figura 7.17

Vista general del módulo.

Y las vistas de las figuras 7.18, 7.19 y 7.20:

Figura 7.18

Vista formulario del modelo *película*.

LaPeliculara		Película	Género	System (dom-0)	
<i>lapeliculera.pelicula.action_window</i> / Nuevo					
<input type="button" value="Guardar"/>	<input type="button" value="Descartar"/>				
Título					
Director					
Duración en minutos	0				
Industria	Europa				
Fecha de estreno en España					
Género					

Figura 7.19

Vista árbol del modelo *película*.



Figura 7.20
Vista formulario del modelo *genero*
(con vista árbol de películas incluidas integrada).

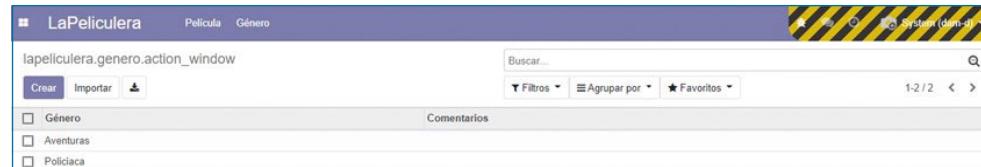


Figura 7.21
Vista árbol del
modelo *genero*.

El procedimiento para hacer informes se ha visto en el apartado 7.4. Un ejemplo de un informe de la aplicación que se está desarrollando se puede ver en la figura 7.22. Para que se pueda obtener hay que incluir la ruta del fichero xml del informe en el fichero *__manifest__.py* tal y como muestra la figura 7.23 y así se añadirá automáticamente bajo el botón *Imprimir* una opción para obtener el informe denominado “Informe sobre películas” (figura 7.24).

```
informe_pelicula.xml
<?xml version="1.0" encoding="utf-8"?>
<odoo>
    <report
        id="lapeliculera.informe_pelicula"
        string="Informe sobre películas"
        model="lapeliculera.pelicula"
        report_type="qweb-pdf"
        name="lapeliculera.informe_pelicula_view"
        file="lapeliculera.informe_pelicula_view"/>
    <template id="informe_pelicula_view">
        <t t-call="web.basic_layout">
            <t t-foreach="docs" t-as="pelicula">
                <div class="page">
                    <h1 t-field="pelicula.name"/>
                    <div>
                        <strong>Director: </strong>
                        <span t-field="pelicula.director"/>
                    </div>
                    <div>
                        <strong>Duración: </strong>
                        <span t-field="pelicula.duracion"/>
                    </div>
                    <div>
                        <strong>Género: </strong>
                        <span t-field="pelicula.genero"/>
                    </div>
                    <div>
                        <strong>Sinopsis: </strong>
                        <span t-field="pelicula.sinopsis"/>
                    </div>
                </div>
            </t>
        </template>
    </odoo>
```

```
21
22     # any module necessary for this one to work correctly
23     'depends': ['base'],
24
25     # always loaded
26     'data': [
27         #'security/ir.model.access.csv',
28         'views/views.xml',
29         'views/templates.xml',
30         'reports/informe_pelicula.xml'
31     ],
32     # only loaded in demonstration mode
33     'demo': [
34         'demo/demo.xml',
35     ]
```

Figura 7.23
Inclusión en *__manifest__.py*.

Figura 7.22
Ejemplo de
Informe.



Figura 7.24
Informe añadido a botón Imprimir.

Actividad propuesta 7.1



¿Qué ocurre al añadir `<t t-call="report.external_layout"> </t>`? ¿Dónde debería añadirse?

Hasta ahora, se ha estado trabajando con el superusuario para no tener incidencias relacionadas con los permisos. Si se hace la prueba de cerrar esta sesión y entrar con otro usuario, incluso si es el usuario *Administrator*, se comprobará que no aparece el módulo *lapeliculera*. Esto es porque aún no se han asignado los permisos del módulo a los grupos de usuarios que van a utilizarlo.

Para ello, como ya se ha comentado en el capítulo 6, una vez creados los distintos grupos de usuarios del módulo (por ejemplo, becario, usuario, administrador, etc.) hay que asignar las distintas vistas a los grupos adecuados (o dejarlas disponibles para todos los grupos) y añadir los usuarios del ERP a los grupos correspondientes.

En el directorio *security* es necesario crear el archivo *security.xml* con un registro (`<record>`) por cada grupo creado, con el siguiente formato:

```

__manifest__.py      x  ir.model.access.csv      x  security.xml      x
1   <?xml version="1.0" encoding="utf-8"?>
2   <ooodo>
3       <record id="lapeliculera_jefe" model="res.groups">
4           <field name="name">Lapelículera / Jefe</field>
5       </record>
6       <record id="lapeliculera_becario" model="res.groups">
7           <field name="name">Lapelículera / Becario</field>
8       </record>
9   </ooodo>
10

```

Figura 7.25
Archivo *security.xml*.

Y, una vez hecho esto, se añaden líneas con permisos específicos por cada grupo y para cada modelo en el *ir.model.access.csv*

```

__manifest__.py          ir.model.access.csv          security.xml
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2 lapeliculera.genero_jefe,Genero Jefe,model_lapeliculera_genero,lapeliculera_jefe,1,1,1,1
3 lapeliculera.pelicula_jefe,Pelicula Jefe,model_lapeliculera_pelicula,lapeliculera_jefe,1,1,1,1
4 lapeliculera.genero_becario,Genero Becario,model_lapeliculera_genero,lapeliculera_becario,1,0,0,0
5 lapeliculera.pelicula_becario,Pelicula Becario,model_lapeliculera_pelicula,lapeliculera_becario,1,0,0,0
6

```

Figura 7.26
Archivo `ir.model.access.csv` con permisos.

El formato lo indica la cabecera. Como ya se ha visto, los cuatro dígitos se corresponden con los permisos de lectura, escritura, creación y eliminación.

En las figuras 7.26 y 7.27 se observa el resultado de crear los grupos de usuario llamados *Lapeliculera/Jefe* y *Lapeliculera/Becario*, el primero con permisos para leer, escribir, crear y borrar, y el segundo solo con permiso de lectura.

Ambos ficheros se añaden al `__manifest__.py`. El orden es importante para que el código incluido en los distintos ficheros haga referencia a objetos previamente definidos.

```

24
25     # always loaded
26     'data': [
27         'security/security.xml',
28         'security/ir.model.access.csv',
29         'views/views.xml',
30         'views/templates.xml',
31         'reports/informe_pelicula.xml'
32     ],

```

Figura 7.27
Inclusión en `__manifest__.py`.

De esta manera, al iniciar sesión con un usuario distinto al superusuario, será mostrado el módulo *lapeliculera*.

Y en los *Ajustes de usuario* de cualquier usuario aparece la posibilidad de asignarlo a uno de los grupos creados:

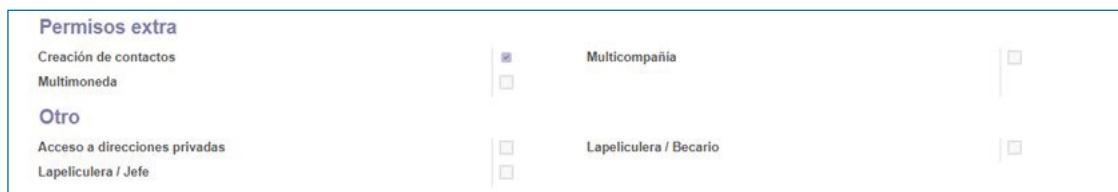


Figura 7.28
Opciones que aparecen en pantalla de permisos de usuario.

En el fichero `views.xml` se pueden asignar los elementos de menú creados a un grupo si lo que se persigue es que sea accesible únicamente por ese grupo. O no se asigna a ninguno, y así accederán todos los usuarios e interactuarán en función de los permisos que se les haya

asignado en el *ir.model.access.csv*. Para que Administrator acceda habrá que añadirlo al grupo correspondiente.

Como broche final, para incluir un icono que identifique visualmente el módulo creado, icono que se incluirá en la caja Kanban que aparecerá en *Aplicaciones*, hay que subir un fichero *png* con el nombre *icon.png* al directorio */static/description* que hay que crear en el directorio principal del módulo.

Actividades propuestas



- 7.2.** Añade un icono para la vista Kanban del módulo *lapeliculera*.
- 7.3.** Añade datos iniciales (no los datos demo). Al menos 10 películas de 5 géneros distintos.
- 7.4.** Incluye las vistas de búsquedas correspondientes a *genero*.
- 7.5.** Establece un filtro llamado “películas cortas”, en el que debes definir un *dominio* para obtener la relación de películas de duración menor a 90 minutos.

Supuestos prácticos

1. Haz una nueva instalación de una máquina con Ubuntu Server y openssh-server para poder trabajar en remoto más cómodamente, por ejemplo, con PuTTY.
 - a) Instala Odoo de forma manual obteniendo el código fuente de los repositorios de Odoo Community y/o OCA.
 - b) Instala en el host desde el que hay conexión a la máquina virtual, el procesador *Atom* y un package para conexiones sftp, por ejemplo, *ftp-remote-edit*, que permitirá trabajar directamente con los ficheros del servidor.
2. En este supuesto se va a practicar la utilización del comando *scaffold* y para ello se va a diseñar una aplicación para Odoo que permita gestionar una cadena de Escuelas de Vela según el siguiente modelo de datos:
 - Escuelas con su denominación, logotipo y datos completos de contacto.
 - Cursos, con su título, duración en días y horas y precio.
 - Monitores, con su nombre y datos habituales para contacto, un código de identificación único y escuela con la que colabora (pueden ser varias).
 - Alumnos, con su nombre y datos habituales de contacto, número de matrícula (distinta secuencia para cada escuela).
 - a) Se deberán establecer las relaciones adecuadas entre las distintas tablas.
 - b) El módulo debe proporcionar vistas para todos los modelos (al menos de tipo formulario y árbol-lista) en un menú con elementos organizados adecuadamente.

- c) Crea modelos de hoja de cálculo o de fichero csv prediseñados para la importación de datos.
 - d) Diseña al menos dos informes distintos.
3. Utilizando el comando scaffold, crea un módulo llamado FCTIES que gestione (de manera limitada) la información sobre los alumnos de un centro de FP y las empresas colaboradoras en las que están haciendo o han hecho las prácticas de FCT.
- a) Crea dos objetos: alumno y empresa.

El modelo *alumno* tendrá, al menos, los siguientes campos:

- Nombre (obligatorio).
- Apellidos (obligatorio).
- Fecha de nacimiento (obligatorio).
- Curso académico (formato 20/21)*.
- Correo electrónico (opcional).
- Teléfono (opcional).
- Ciclo formativo (puede ser: DAM, DAW y ASIR, obligatorio).
- Periodo de práctica (abril o septiembre, obligatorio).
- Nota media (obligatorio).
- Nota media en formato texto (campo calculado como se indica más adelante).

El modelo *empresa* tendrá, al menos, los siguientes campos:

- Nombre (obligatorio).
- Persona de contacto (obligatorio).
- Teléfono de contacto (obligatorio).
- Correo electrónico (obligatorio).
- Dirección (obligatorio).

Además, deberás incluir los campos de relación necesarios para establecer la correspondencia entre ambos modelos.

- b) Incluye un ícono para el módulo.
- c) Crea las vistas listado y formulario de ambas entidades. La lista de alumnos que han hecho las prácticas de FCT en cada empresa debe aparecer incluida en la vista formulario del modelo empresa con nombre, apellidos, curso académico y periodo de realización de las FCT.
- d) Dentro de la vista lista (tree) incluye una vista de búsqueda que permita buscar a los alumnos por apellidos y por empresa en la que ha realizado las prácticas. También se debe poder realizar búsquedas de alumnos por curso académico, convocatoria y empresa.
- e) Diseña un filtro para obtener una relación de alumnos con una nota media mayor de 8 puntos. El campo (calculado) con la nota media en formato de texto debes programarlo de tal forma que si la nota está comprendida entre:

- [5,7]: la nota será “Aprobado”
 - [7,9]: la nota será “Notable”.
 - [9,10]: la nota será “Sobresaliente”.
- f) Crea los menús FCT (root) y submenús Alumnos y Empresas.
- g) Crea dos grupos de usuarios: Coordinador y Profesor. El primero dispondrá de control total sobre las dos tablas del módulo. El segundo grupo, tendrá los permisos de lectura y escritura sobre Alumnos y solo de lectura sobre Empresas.
- h) Diseña un informe que permita imprimir los datos de una empresa y con ella todos los alumnos que han hecho prácticas en la misma agrupados por curso académico. De cada alumno se visualizarán nombre, apellidos, teléfono, periodo de práctica y ciclo formativo.

Nota: La BBDD debe ser nueva y no tener instalado ningún módulo más que los básicos que por defecto instala Odoo. Proporciona la aplicación (BBDD) lista para su instalación en otro equipo. Para ello se podrá elegir cualquiera de las opciones conocidas hasta ahora para hacer un *backup* que se pueda restaurar en otra localización (recuerda proporcionar un usuario y su contraseña).

Resumen

- Además de la modificación mediante la edición gráfica de los objetos de Odoo, se puede escribir el código (Python, xml, etc.) que conforma toda la estructura de cualquier módulo en Odoo.
- Procesadores especializados o un IDE permiten, realizar el trabajo directamente sobre el servidor de Odoo evitando la descarga y subida de ficheros.
- El comando *scaffold* crea una estructura básica de un módulo de Odoo.
- Los ficheros incluidos en las carpetas creadas por el comando *scaffold* contienen líneas comentadas que ayudan al programador a escribir su código.
- Los ficheros *__init__.py* y *__manifest__.py* del directorio raíz son los archivos fundamentales del módulo.
- El archivo *__manifest__.py* contiene toda la información descriptiva del módulo , sus dependencias y en él se declaran los archivos .xml que lo integran.
- El archivo *__init__.py* es el primer archivo que se ejecuta, y contiene información y localización acerca de otros archivos Python que se deben ejecutar.
- Las vistas se escriben en lenguaje .xml y utilizando Qweb y se refieren siempre a un modelo concreto.
- Existen los conceptos de *herencia de vistas* y vistas que contienen otras vistas.
- En el fichero (o ficheros) *views.xml* de la carpeta *views* se definen las vistas, los elementos de menú y las acciones de ventana asociadas.
- Los informes se desarrollan en un fichero *xml*, utilizando Qweb y se suelen incluir en el directorio *reports*, que se crea al efecto. Deben declararse en *__init__.py*

- Para que las vistas sean accesibles solo por los grupos apropiados, se utiliza el fichero *security.xml* de la carpeta *security*, añadiendo una línea por cada grupo.
- En un fichero csv se especifican los permisos que va a tener cada grupo.
- El fichero *security.csv* tiene la cabecera: *id, name, model_id:id, group_id:id, perm_read, perm_write, perm_create, perm_unlink* y los campos *perm_read, perm_write, perm_create, perm_unlink* se ponen a 1 o 0 según se conceda o no el permiso.

ACTIVIDADES DE AUTOEVALUACIÓN

1. Los archivos de la carpeta raíz de un módulo Odoo son:

- a) *__init__.py* y *models.py*
- b) *models.py* y *manifest.py*
- c) *__init__.py* y *__manifest__.py*
- d) *__models__.py* y *__manifest__.py*

2. En Odoo, las vistas son objetos de:

- a) *ir.ui.view*
- b) *ui.ir.view*
- c) *ir.ui.menu*
- d) *ir_ui_view*

3. Los modos de herencia de vista son:

- a) Directa y jerárquica.
- b) Base y de extensión.
- c) Formulario y Kanban.
- d) Formulario, árbol y Kanban.

4. Los informes en Odoo (señala todas las correctas):

- a) Se suelen guardar en el directorio *reports*.
- b) Se desarrollan utilizando Qweb.
- c) Son editables.
- d) Se compone de una única sección *<report>*.

5. ¿Qué fichero incluye una descripción corta y una larga del módulo?:

- a) *__init__.py*
- b) *__models__.py*
- c) *__manifest__.py*
- d) *__security__.csv*

7. ¿En qué archivo se declaran las dependencias y los xml que integran el módulo?:

- a) __security__.csv
- b) __models__.py
- c) __init__.py
- d) __manifest__.py

8. ¿Qué es id,name,model_id:id,group_id:id,perm_read...?:

- a) Parte del contenido que debe incluir el fichero security.xml.
- b) Parte de la cabecera del fichero security.csv.
- c) Alguno de los campos del modelo res.security.
- d) Campos de modelos transitorios.

9. En Odoo, los campos calculados:

- a) Se construyen mediante un método.
- b) Se construyen mediante una función.
- c) Hay que refrescarlos en las vistas.
- d) Ninguna de las soluciones es cierta.

10. Señala los tipos de vista existentes en Odoo:

- a) search.
- b) qweb.
- c) list.
- d) extension.

SOLUCIONES:

1. **a** **b** **c** **d**

2. **a** **b** **c** **d**

3. **a** **b** **c** **d**

4. **a** **b** **c** **d**

5. **a** **b** **c** **d**

6. **a** **b** **c** **d**

7. **a** **b** **c** **d**

8. **a** **b** **c** **d**

9. **a** **b** **c** **d**

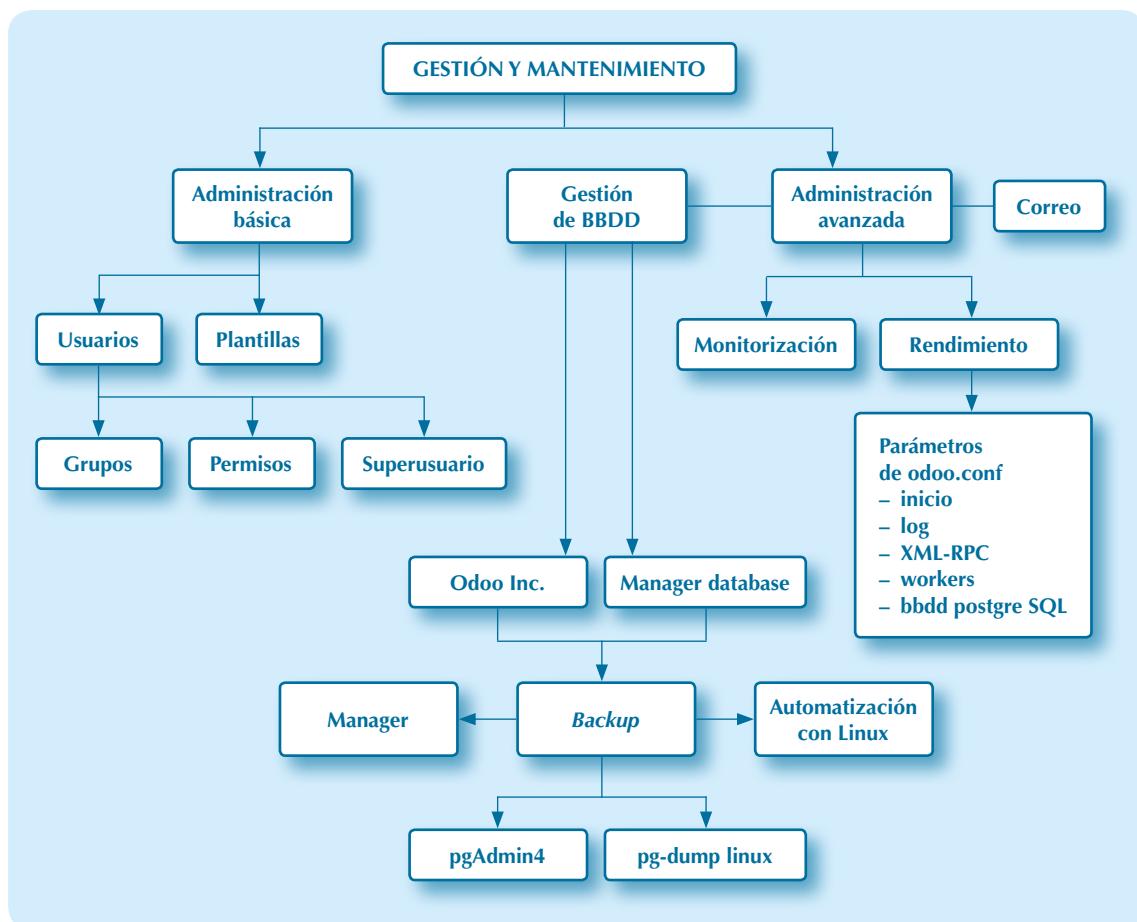
10. **a** **b** **c** **d**

Gestión y mantenimiento IT de un ERP

Objetivos

- ✓ Asimilar la necesidad de administración de una instancia de Odoo.
- ✓ Discernir entre labores de administración básica y avanzada.
- ✓ Aprender a gestionar usuarios, grupos y permisos.
- ✓ Saber utilizar y modificar plantillas de documentos que incluye Odoo.
- ✓ Realizar un mantenimiento seguro de las BBDD.
- ✓ Conocer las distintas opciones para automatizar *backups*.
- ✓ Aprender a configurar el correo electrónico corporativo dentro de Odoo.
- ✓ Entender el concepto de evaluación del rendimiento y los actores que contribuyen.
- ✓ Saber monitorizar procesos intervinientes.
- ✓ Realizar la configuración adecuada mediante la personalización del fichero de configuración de Odoo.

Mapa conceptual



Glosario

Campos de combinación. Etiqueta en una plantilla que se corresponde con un campo en una base de datos y que en el momento de ejecución se sustituye por el valor contenido en el campo.

CIDR (*Classless Inter-Domain Routing*). Nomenclatura estándar para la asignación de direcciones de red formada por una dirección y un sufijo entre 0 y 32 (en IPv6 entre 0 y 128) que indica el número de posiciones que identifican la dirección de red, separados por una "/".

GIL (*Global Interpreter Lock*). Bloqueo de intérprete global de Python. Característica de Python que provoca que un programa multiproceso nunca se ejecute en paralelo, pues permite que un solo hilo tome el control del intérprete, es decir, que solo un hilo puede estar en ejecución a la vez. También conocido por la comunidad como "*an infamous feature of Python*".

md5 Message-Digest Algorithm 5. Algoritmo de resumen del mensaje 5. Algoritmo criptográfico de 128 bits.

Modelo multihilo. Modelo de programación que ejecuta varias tareas (hilos) de un mismo proceso de forma concurrente (multitarea especializada).

Nginx. Proxy inverso de código abierto recomendado para Odoo.

Proxy inverso. Servicio que intermedia entre la solicitud de un cliente y los servidores que le responden. Proporciona balance de carga, encriptación del tráfico, firewall de aplicación, contenido estático, caché (caching), etc.

Werkzeug. Biblioteca de utilidades WSGI (*Web Server Gateway Interface*) para Python, utilizada por Odoo. WSGI intermedia entre un servidor web como Apache y el script Python.

Worker. Unidades de multiprocesamiento que utilizará Odoo en función de los núcleos de procesador disponibles (hardware).

8.1. Administración básica en Odoo

Tanto si se utiliza una versión SaaS como si se dispone de un instalación *On-Premise* de Odoo, es responsabilidad del administrador velar por el buen funcionamiento de la aplicación, así como realizar las labores básicas de administración.

Se entenderá por labores básicas las relacionadas con la gestión de los usuarios y permisos y la administración de plantillas de documentos, conviniendo que la administración avanzada incluirá la gestión de servicios relacionados con la web y el correo, la administración de las BBDD y la organización de las copias de seguridad.

8.1.1. Gestión de usuarios. Usuarios, grupos y permisos

Un *usuario* en Odoo es cualquiera que accede a la aplicación, ya sea empleado, contacto, cliente o proveedor. Debido a esto, la utilización de unos *permisos* de acceso perfectamente estructurados se hace imprescindible para restringir la información a la que no debe acceder en función de la identidad.

Un *Grupo* es un conjunto de usuarios con un perfil común que les dota de una serie de privilegios de acceso. Los grupos permiten definir reglas de acceso a una aplicación, conocer los permisos heredados, así como los menús y vistas a los que los usuarios pertenecientes van a tener acceso.

El *Control de acceso*, que es el primer nivel de seguridad, se refiere a un *modelo* y está asociado normalmente a los grupos de usuarios. Como cualquier usuario puede ser incluido en varios

grupos, este tipo de permisos de acceso son acumulativos. Si no están referidos a un grupo, se aplican a todos los usuarios sin excepción.

En *Ajustes/Usuarios* y *Compañías/Usuarios* se puede obtener la información sobre los permisos asociados a un usuario concreto para cada *Módulo* instalado:

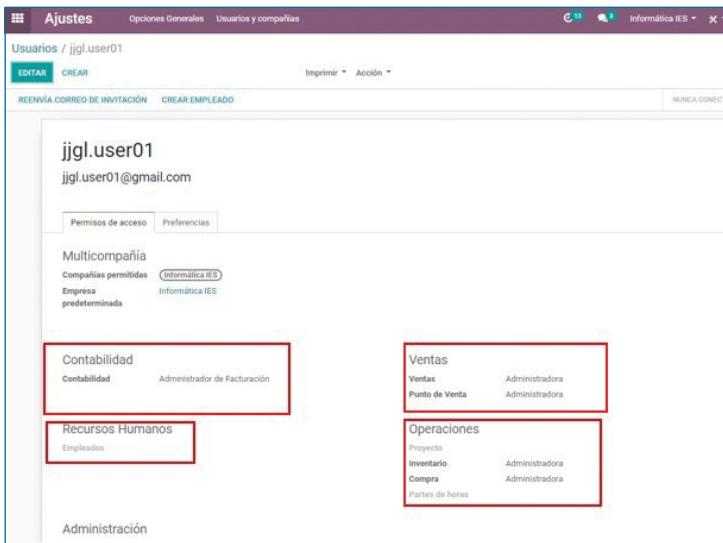


Figura 8.1
Permisos asignados
al usuario jjgl.user01.

Este usuario *jjgl.user01* tiene asignado, para cada módulo de la captura de la figura 8.1, los Controles de acceso definidos en los siguientes grupos de permisos:

- Contabilidad: administrador de facturación.
- Recursos humanos: empleado.
- Ventas: administradora.
- Punto de venta: administradora.
- Inventario: administradora.
- Compra: administradora.



TOMA NOTA

Con la opción Editar se puede cambiar entre los distintos niveles.

Para ver en detalle los tipos de permisos, hay que acceder a *Ajustes/Usuarios y Compañías/Grupos* para obtener un listado de todos los grupos existentes. Escogiendo el grupo, en la pestaña de *Permisos de acceso* se relacionan todos los permisos para cada objeto de forma detallada. El control de acceso está en el modelo *ir.model.access*.

Los posibles permisos son: crear (*perm_create*), leer (*perm_read*), escribir (*perm_write*) y borrar (*perm_unlink*).

Así, que el usuario de la figura 8.1 tenga el perfil *Administradora* para el módulo *Ventas* se traduce en el siguiente detalle de permisos de Acceso:

The screenshot shows a table with columns: Nombre, Objeto, Permiso para leer, Permiso de escritura, Acceso para crear, and Permiso para eliminar. The row for 'res.partner.crm.manager' is highlighted with a red border. The 'Permiso para leer' column contains checked boxes for most rows, while the 'res.partner.crm.manager' row has an unchecked box.

Nombre	Objeto	Permiso para leer	Permiso de escritura	Acceso para crear	Permiso para eliminar
stock.picking.sales	Albarán	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
account_move.manager	Asientos contables	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
calendar.event.manager	Calendario de eventos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
res.partner.crm.manager	Contacto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
res.partner.sale.manager	Contacto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
res.partner.group.sale.manager	Contacto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
crm.team.manager	Equipo de ventas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
crm.lead.manager	Iniciativa/Oportunidad	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
sale.order.manager	Pedido de venta	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.template.salemanager	Plantilla de producto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.product.stock.manager	Plantilla de producto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 8.2
Permisos de un grupo concreto.

Se puede observar que para el modelo *res.partner.crm.manager* (objeto contacto) el usuario *jjgl.user01* del ejemplo tiene permiso solo para leer. No tiene permiso para escribir, crear ni borrar.

Los permisos de un grupo pueden ser heredados de otros grupos (véase la pestaña *Heredado*), por lo que es buena práctica trabajar con grupos de usuarios existentes para después añadir los usuarios creados a sus grupos de pertenencia. Además, si se quieren crear nuevos grupos, es aconsejable, hasta disponer de un nivel avanzado de desarrollo en Odoo, utilizar como base uno de los grupos ya creados (utilizar la herencia) y modificar los permisos de acceso en función de las necesidades. Hay que tener en cuenta que la modificación de permisos de acceso puede tener un impacto importante sobre el ERP.



Los permisos se aplican a objetos y se refieren, entre otros, a modelos de la base de datos y a plantillas o vistas de formulario. Por ejemplo, en el grupo Administradora del módulo de Inventario, para *product* (producto) existen varias entradas y dos de ellas son: *product.product.stock_manager* y *product.template.stock_manager*.

La nomenclatura seguida es:

Modelo de producto	<i>product.product</i>
Plantilla de producto	<i>product.template</i>
Módulo	Inventario (stock)
Grupo	Administradora (manager)

Nombre	Objeto	Permiso para leer...	Permiso de escritura	Acceso para crear	Permiso para eliminar
stock.picking.manager	Albarán	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
res.partner.group.stock.manager	Contacto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
stock.scrap.manager	Desechar	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.template.stock.manager	Plantilla de producto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.product.stock.manager	Producto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
stock.warehouse.manager	Almacén	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.attribute.manager	Atributo de producto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 8.3
Nomenclatura estándar.

Los permisos asignados para product.product se refieren a los que permiten o no interactuar con la tabla de productos, mientras que los permisos especificados en product.template se aplican a la plantilla y permitirán modificar los botones y acciones de ventana que aparecen en el formulario.

Otras pestañas de este formulario contienen los Menús y Vistas a los que tiene acceso el Grupo al que se ha asignado el usuario.

En un segundo nivel se encuentran las Reglas, que son condiciones que se aplican sobre los registros de un Modelo y que sirven para afinar los permisos de acceso:

Nombre	Objeto	Dominio	Aplicar para lectura	Aplicar para escritura	Aplicar para creación	Aplicar para eliminación
SMS Template: sale mana...	SMS Templates	[(model_id.model, 'in', ('crm.lead', 'res.partner'))]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 8.4
Reglas de registro de un grupo.

Las reglas de registro son una parte importante de la seguridad en Odoo. Controlan a qué registros puede acceder un usuario, por ejemplo, para conseguir que los usuarios de ventas puedan acceder a sus propios presupuestos y pedidos de ventas y no a los de otros usuarios.

Las reglas se llaman *Globales* si no se aplican a un grupo concreto y solo pueden restringir el acceso o ser inoperantes.

Nombre	Objeto	Grupos	Dominio	Aplicar para lectura	Aplicar para escritura	Aplicar para creación	Aplicar para eliminación
ir.ui.view_custom_rule	Vista personalizada		[(user_id,'>',user_id)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Defaults: alter personal...	Valores por defecto	(Tipos de Usuario / Us...	[(user_id,'>',user_id)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Defaults: alter all defaults	Valores por defecto	(Administración / Ajustes)	[(1,'>',1)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
user rule	Usuarios		[(share,'>',False), (company_ids,'in', company_ids)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Location multi-company	Ubicaciones de inventa...		[(company_id,'>',False), (company_id,'in', company_ids)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Stock Operation Type ...	Tipo de albarán		[(company_id,'in', company_ids)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
multi-company currenc...	Tasa monetaria		[(company_id,'>',False), (company_id,'in', company_ids)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 8.5
Relación de reglas de registro.

Recurso web

Reglas de registro (página oficial de Odoo).

www

A) Añadir usuarios

Después del proceso de instalación, al crear la base de datos, el formulario solicita, además del nombre y otros detalles, una dirección de *e-mail* y una contraseña. Con ellas se crea el usuario *Administrator*. En un primer momento será este el único usuario con acceso.

La opción de *Administrar usuarios* de la pantalla de *Ajustes/Opciones generales* permite añadir nuevos usuarios individuales especificando el tipo (interno, del portal o invitado).

En el formulario de la pestaña *Permisos de acceso* se pueden asignar grupos de permisos para cada una de las aplicaciones actualmente instaladas.

Finalizado este proceso, una vez creado el usuario, un *e-mail* será enviado automáticamente a la cuenta de correo introducida en el campo correspondiente al usuario para invitarle a acceder al ERP.

El formato del cuerpo del mensaje está prefijado en una plantilla que puede modificarse (figura 8.7).

The screenshot shows the 'Nuevo' (New) user creation form. It includes fields for Nombre (Name), Correo electrónico (Email), Permisos de acceso (Access rights) set to 'Profesional' (Professional), Multicompañía (Multi-company) set to 'Informatika IES', Empresa predominante (Primary company) set to 'Informatika IES', and Tipo de Usuario (User type) set to 'Usuario interno' (Internal user). Below these, there are sections for Contabilidad (Accounting), Recursos Humanos (Human Resources), and Operaciones (Operations), each with dropdown menus for selecting specific roles like Ventas (Sales), Administrador (Administrator), etc. At the bottom right, there is a 'Siguiente' (Next) button.

Figura 8.6
Plantilla de creación de usuarios.

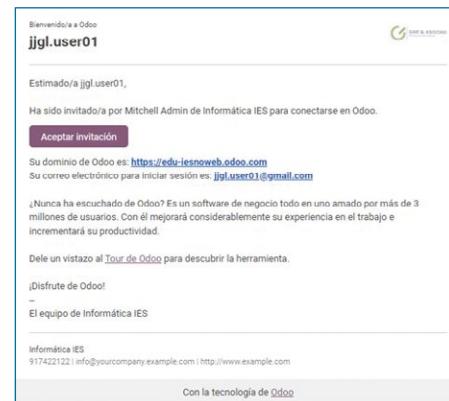


Figura 8.7
Formato del correo de invitación recibido por un nuevo usuario.

Cuando el receptor acepte la invitación, en el enlace proporcionado en el *e-mail*, Odoo permitirá al nuevo usuario escoger una contraseña para acceder de forma segura a la aplicación y le asignará los permisos especificados durante el proceso de creación.

Si para la creación de usuarios se opta por la opción más simple de *Invitar a nuevos usuarios*, un *e-mail* será enviado automáticamente pero, en este caso, el administrador no habrá tenido

ocasión de modificar los grupos de pertenencia y por tanto será añadido como usuario con unos permisos por defecto.

B) Superusuario

En las últimas versiones, el usuario *Administrator* ya no es el superusuario. En Odoo existe un superusuario (*System* u *OdooBot*, dependiendo de los módulos instalados) al que no se le aplica ningún permiso ni restricción de acceso. Se accede desde el ícono de *Herramientas de Desarrollo/Conviértete en superusuario* y para dejar de serlo es imprescindible cerrar sesión.

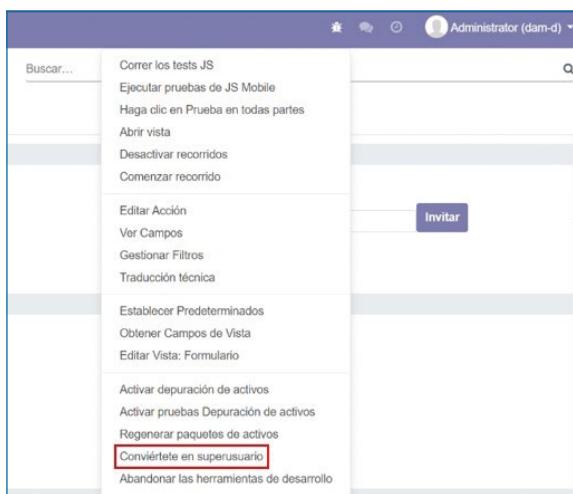


Figura 8.8
Opciones de Herramientas de desarrollo.

Así, la administración de una instancia de Odoo se comparte entre el usuario *Administrator* inicial que se ocupa de la parte funcional y el usuario *System/OdooBot* para la parte técnica.

El estado de superusuario es reconocible por el fondo amarillo-negro de rayas diagonales que aparece en la esquina superior derecha.



Figura 8.9
Entorno de superusuario.

8.1.2. Plantillas de documentos

Todos los módulos de Odoo disponen de plantillas de documentos prediseñadas.

Hay plantillas de presupuesto, orden de compra, albarán, factura y muchas más, todas editables y configurables.

En las opciones de configuración o en los asistentes de inicio de cada uno de los módulos se puede acceder a ellas.

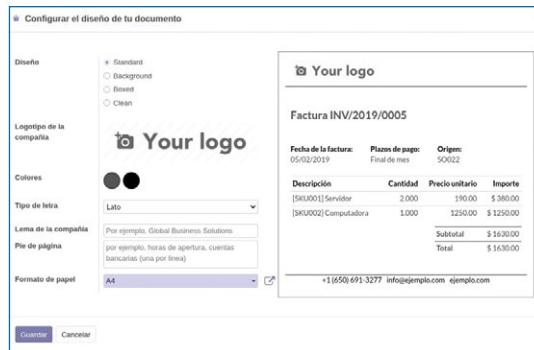


Figura 8.10
Formulario para edición de plantilla de Factura.

Ajustes							Opciones Generales	Usuarios y compañías	Traducciones	Técnico	Administrador (dam-d)	
Plantillas												
Crear	Importar		<input type="text"/> Buscar...									
<input type="checkbox"/> Nombre	Aplica a	Asunto	Desde	Para (correos electrónicos)	A (empresas)	Fichero del informe						
<input type="checkbox"/> Auth Signup: Odoo Accou...	Usuarios	Bienvenido a \${object.com...	\${object.company_id.nam...	\${object.email_formatted} ...								
<input type="checkbox"/> Auth Signup: Odoo Conne...	Usuarios	\$({object.create_uid.name})	* \${object.company_id.nam...	\${object.email}@safe								
<input type="checkbox"/> Auth Signup: Reset Passw...	Usuarios	Restablecer contraseña	* \${object.company_id.nam...	\${object.email_formatted} ...								
<input type="checkbox"/> Auth Signup: Unregistered...	Usuarios	Recordatorio para usuario...	* \${object.company_id.part...	\${object.email_formatted} ...								
<input type="checkbox"/> Calendar: Date updated	Información del Calendari...	\$({object.event_id.name}) F...	\${object.event_id.user_id} ...	\${("if object.partner_id.em...}	\${object.partner_id.id if obj...							
<input type="checkbox"/> Calendar: Meeting invitation	Información del Calendari...	Invitación a \${object.event...	\${object.event_id.user_id} ...	\${("if object.partner_id.em...}	\${object.partner_id.id if obj...							
<input type="checkbox"/> Calendar: Reminder	Información del Calendari...	\$({object.event_id.name}) - ...	\${object.event_id.user_id} ...	\${("if object.partner_id.em...}	\${object.partner_id.id if obj...							
<input type="checkbox"/> Delivery: Send by Email	Albarán	\$({object.company_id.nam...			\${object.partner_id.id})	\${(object.name or ")replac...						
<input type="checkbox"/> Digest: Default main templ...	Resumen	\$({% if % (cx.get('user')) ...										
<input type="checkbox"/> Invoice: Send by email	Asientos contables	\$({object.company_id.nam...	\${(object.invoice_user_id} ...		\${object.partner_id.id})	Factura. \${(object.name or ...						
<input type="checkbox"/> Partner Mass Mail	Contacto					\${(object.id)}						
<input type="checkbox"/> Payment Receipt: Send by ...	Pagos	\$({object.company_id.nam...				\${object.partner_id.id})	\${(object.name or ") replac...					
<input type="checkbox"/> Portal: new user	Configuración usuario portal	Su cuenta de Odoo en \${o...										

Figura 8.11
Relación de plantillas.

Odoo también dispone de varias plantillas de cuerpos de mail, accesibles desde Ajustes/Técnico/Plantillas. Permiten utilizar campos de combinación utilizando la solapa *Generador dinámico de expresiones con marcadores de posición*.



Figura 8.12
Campos combinados.



En la figura 8.13 se pueden ver las dos pantallas con la secuencia de acceso al e-mail enviado a un cliente para la confirmación de la recepción de un pedido.

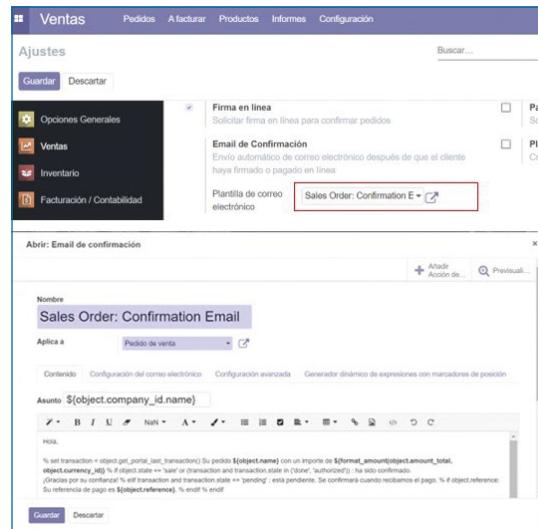


Figura 8.13
Secuencia de acceso a una plantilla.

8.2. Administración avanzada en Odoo

Las tareas relacionadas con el mantenimiento de las bases de datos y su disponibilidad, la gestión del correo corporativo y su imbricación en el ERP y la optimización para un mayor rendimiento del sistema son las más críticas de todas las ocupaciones responsabilidad del administrador de la instalación en producción de un ERP.

En este libro se ha convenido considerarlas parte de la Administración avanzada. De un buen desempeño en el día a día de los técnicos depende el buen funcionamiento de una solución tan importante para la marcha de la empresa.

8.2.1. Gestión de las bases de datos

En la página del perfil del administrador, que se encuentra en www.odoo.com aparece en el apartado *My Databases* una relación de las bases de datos instaladas en la modalidad *SaaS*.

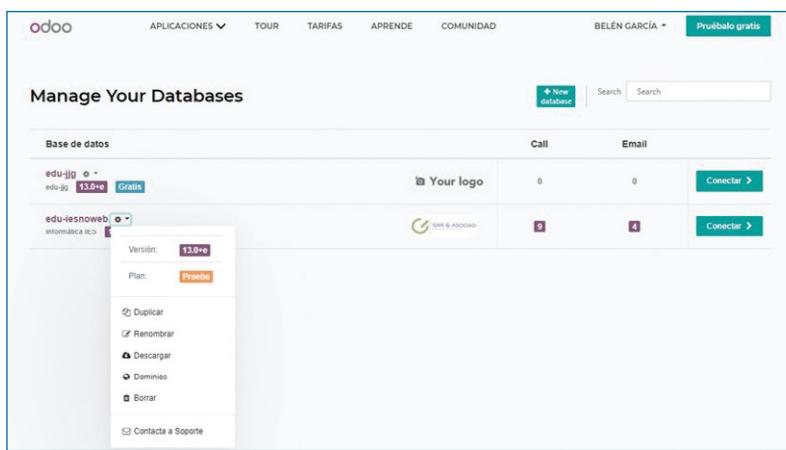


Figura 8.14
Bases de Datos
en Odoo SaaS.

Además de que el administrador se puede conectar desde aquí, como camino alternativo a https://nombre_bbdd.odoo.com, existe un menú de configuración básica para la gestión de las bases de datos creadas que permite Duplicar, Renombrar, Descargar, Asociar dominios y Borrar.

Además, Odoo ofrece servicios avanzados para este tipo de instalaciones de pago.

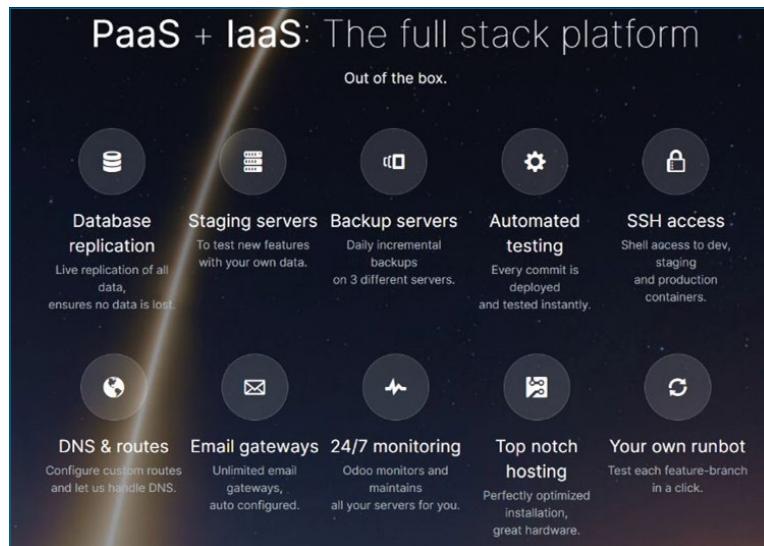


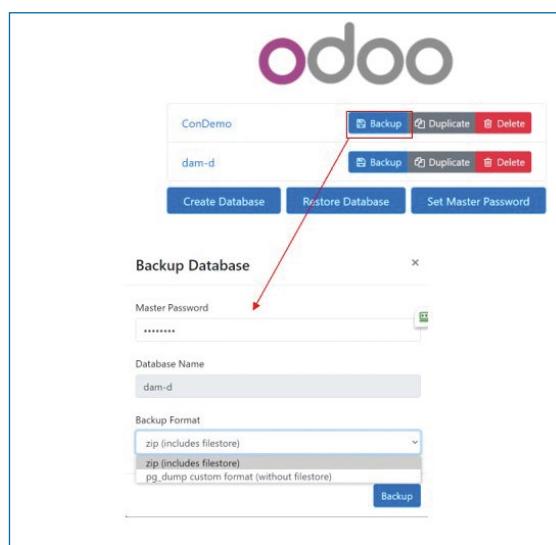
Figura 8.15
Página web de Odoo S.A. con oferta de servicios.

Sin embargo, en el caso de instalaciones *On-Premise*, dependiendo del entorno (del sistema operativo) existen varias opciones para gestionar el sistema de *backup* de las bases de datos del ERP.

A) Backup desde la opción *Manage Database* de la instancia de Odoo

El propio Odoo proporciona en la opción de gestión de las bases de datos, un sistema de copia de seguridad manual al que se accede desde el botón *backup* (figura 8.16) que permite obtener un *backup* en formato zip o mediante una utilización personalizada del comando de Linux *pg_dump*, comando que utiliza también, como se verá en siguientes apartados, la herramienta *pgAdmin4*.

Figura 8.16
Acceso al *backup* desde el Database Manager.



En el primer caso (zip), además de los archivos binarios, se obtiene el fichero *dump.sql* y el manifest.json que proporcionan información acerca de la copia realizada y mantiene la estructura json de los ficheros para la restauración, por lo que puede ser la solución más adecuada.

En el segundo caso se obtiene un fichero .dump que no incluye los ficheros binarios.

B) Backup utilizando pgAdminIV

La herramienta gráfica de gestión de PostgreSQL, pgAdmin4, también brinda la posibilidad de hacer copias de seguridad.

```

{
  "odoo_dump": "1",
  "db_name": "dam-d",
  "version": "13.0",
  "version_info": [
    13,
    0,
    0,
    "final",
    0,
    ""
  ],
  "major_version": "13.0",
  "pg_version": "10.0",
  "modules": {
    "l10n_es": "13.0.4.0",
    "http_routing": "13.0.1.0",
    "sale": "13.0.1.1",
    "stock": "13.0.1.1",
    "calendar_sms": "13.0.1.0",
    "fetchmail": "13.0.1.0",
    "base": "13.0.1.3",
    "sale_stock": "13.0.1.0",
    "account": "13.0.1.1",
    "resource": "13.0.1.1",
    "base_setup": "13.0.1.0",
    "odoo_retail": "13.0.1.0"
  }
}

```

Figura 8.17
Contenido de manifest.json.

RECUERDA

Por defecto PostgreSQL no permite conexiones remotas. Para acceder en remoto a las bases de datos PostgreSQL hay que configurar el entorno adecuadamente. Para ello hay que modificar los ficheros pg_hba.conf y postgresql.conf.

En el primero añadiendo una línea en el apartado correspondiente a las conexiones IPv4

```
host all all "red_CIDR" md5
```

#	TYPE	DATABASE	USER	ADDRESS	METHOD
# "local" is for Unix domain socket connections only					
local	all	all			peer
# IPv4 local connections:					
host	all	all		127.0.0.1/32	md5
host	all	all		192.168.1.0/24	md5
# IPv6 local connections:					
host	all	all		::1/128	md5

Figura 8.18
Fichero
pg_hba.conf.

Y en el segundo, por ejemplo, con listen_addresses='*' u otras opciones más restrictivas. Este ejemplo permite el acceso desde cualquier equipo de la red local 192.168.1.0/24. No olvidar:

```
$ sudo systemctl restart postgresql
```

```

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -
listen_addresses = '*'                      # what IP address(es) to listen on;
                                              # comma-separated list of addresses;
                                              # defaults to 'localhost'; use '*' for all
                                              # (change requires restart)
port = 5432                                    # (change requires restart)
max_connections = 100                         # (change requires restart)

```

Figura 8.19
Fichero
postgresql.conf.

Al conectar a la base de datos que se va a copiar, con el botón derecho sobre su nombre, se obtiene el asistente de *backup* que ofrece la posibilidad de hacer cuatro tipos de copia de seguridad utilizando *pg_dump* con opciones distintas:

- *Custom*. El comando *pg_dump* utiliza *--format=c --blobs* como parámetro. Con la opción *Restore* de pgAdmin4 se recomponen las tablas.
- *Tar*. El comando *pg_dump* utiliza *--format=t --blobs* como parámetro. Almacena los binarios de cada tabla en archivos y añade un fichero *restore.sql* con información de la copia realizada para poder hacer la recuperación de las tablas. Con la opción *Restore* de pgAdmin4 se recupera el fichero *.tar*

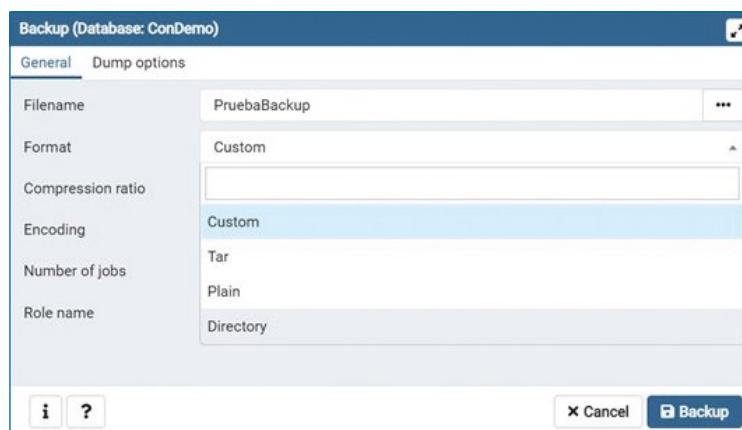


Figura 8.20
Tipos de backup con pgAdmin4.

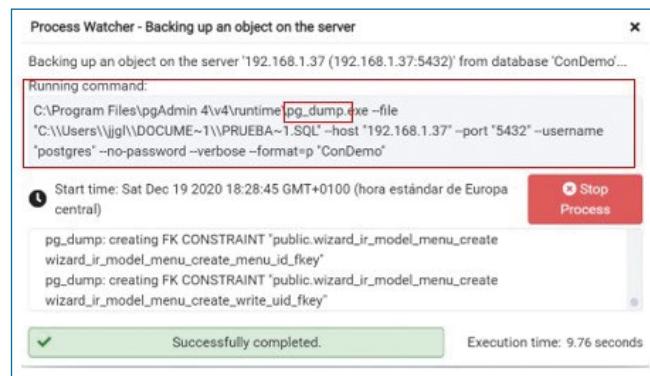


Figura 8.21
Realización de un backup
con pgAdmin4.

- *Plain*. El nombre del archivo debe tener la extensión *.sql*. El comando *pg_dump* utiliza *--format=p* como parámetro. En este caso no se puede utilizar la opción de *Restore* de pgAdmin4. Se crea la base de datos (importante hacerlo con usuario propietario) y después, seleccionada, en la pestaña con Query Tool () / Open File () se copia el contenido del fichero *.sql*. Este método no es fiable para ficheros grandes. Como opción, para este tipo de copia en fichero *.sql* se puede restaurar por consola mediante:

```
psql -h ip_servidor o localhost -p 5432 -U postgres -d dam-d -f
pruebacopia.sql
```

- *Directory.* El comando *pg_dump* utiliza *--format=d* como parámetro. Con la opción *Restore* de pgAdmin4 se recupera este tipo de copia.

En todos los casos, la base de datos sobre la que restaurar debe existir previamente. Es decir, en caso de pérdida de la base de datos original por borrado, hay que crear previamente una base de datos vacía con el mismo nombre (con el mismo usuario propietario antes de ejecutar el *Restore*). Si se quiere, también se puede restaurar sobre otra base de datos.

C) Comando *pg_dump* de Linux

En cualquier caso, no hay que olvidar que en entornos Linux, es posible utilizar el comando *pg_dump* para realizar el *backup*. Para ello, basta con crear un directorio para las copias, accesible por parte del usuario de instalación (por ejemplo, *odoo*) y posicionándose en él ejecutar:

```
# pg_dump "nombreBBDD" > "nombre_backup".sql
```

O, lo que es lo mismo:

```
# pg_dump "nombreBBDD" -f "nombre_backup".sql
```

Si se quiere especificar una de las opciones de salida para el tipo de fichero de backup creado:

```
# pg_dump "nombreBBDD"--format="x" -f "nombre_backup"
```

Donde x puede ser c, d, t, p (custom, directory, tar o plaintext).

D) Automatización del backup

Todo lo expuesto hasta ahora trata de la realización de *backups* de forma manual, lo que no es solución en entornos profesionales. Es necesario entonces abordar la automatización de la copia de seguridad, para que, con la frecuencia deseada y utilizando una ventana de poca actividad, se realicen los *backups* sin intervención del administrador.

En lo que se refiere a las posibilidades que brinda el propio sistema operativo (en el caso de esta publicación, Ubuntu), una de ellas es la utilización conjunta del comando *date* y de *cron*.

Para obtener el momento exacto de la realización del *backup* se puede utilizar *date* y para crear el servicio, con la herramienta `# crontab -e` se edita el fichero del demonio *cron*, que es el gestor de tareas por usuario, donde se incluirán las instrucciones para ejecutar un fichero shell de forma automatizada.



Actividades propuestas

- 8.1.** Crea una copia de seguridad en formato tar de la base de datos principal de una instancia de Odoo y programala para que se realice todos los días de la semana a la 01:15 horas. La copias se guardarán en el directorio creado al efecto, llamado *backups*.
- 8.2.** Repite la actividad anterior automatizando la opción que ofrece la propia gestión de bases de datos de Odoo.

Figura 8.22
Detalle
de petición POST.



PARA SABER MÁS

Mantener el *backup* en el mismo equipo no es una opción aconsejable. Aumentar la seguridad del sistema implica, entre otras cosas, realizar la copia sobre otros soportes, como discos o cintas, e incluso exportar los backups realizados a otra ubicación.

Actualmente, el *Cloud* proporciona un amplio abanico de posibilidades para ello. Herramientas como GoodSync (de pago), rsync o rclone en entornos Linux o RaiDrive en entornos Windows ayudan en el proceso de guardar los backups realizados en otros equipos o en la nube, pues muchas de ellas son compatibles con la mayoría de servicios, como Dropbox, Google Drive, OneDrive, Amazon S3 u Office 365.

<https://rsync.samba.org/>
<https://www.raidrive.com/>
<https://www.goodsync.com/>
<https://rclone.org/>

8.2.2. Correo

La configuración del correo en Odoo se realiza desde las opciones que aparecen en el menú *Ajustes/Técnico*.

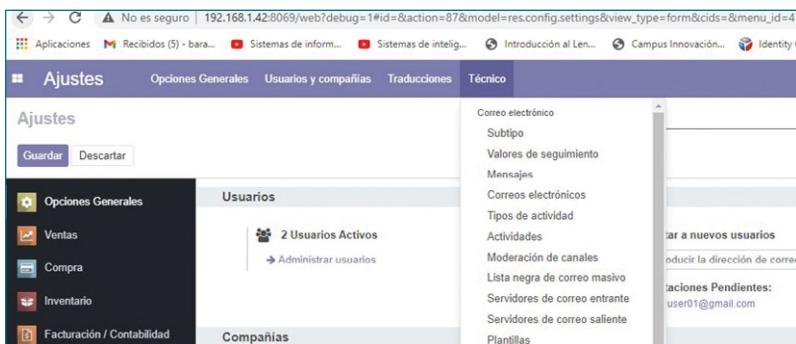


Figura 8.23
Acceso a la configuración de los servidores de correo.

Si se dispone de dominio propio, alojado en un ISP, basta con utilizar los datos de configuración proporcionados por el proveedor respecto a los servidores de entrada y salida del correo.

Se pueden tener varios servidores jerarquizados con prioridades (número menor implica mayor prioridad).

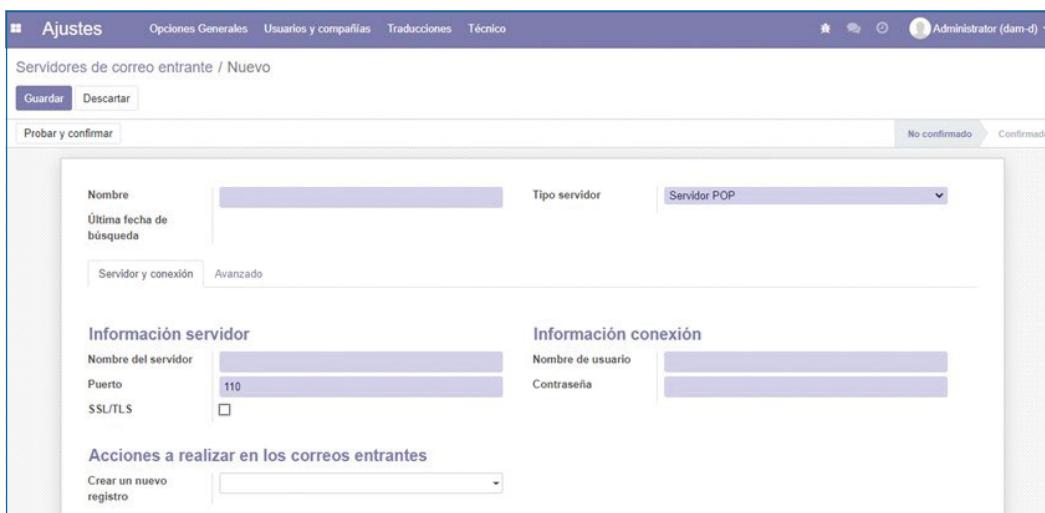


Figura 8.24
Pantalla de configuración de servidor de correo entrante (pop o imap).

Por defecto, la frecuencia de comprobación y envío de correos (la gestión de las colas del e-mail) es de 1 hora, dato que se puede modificar en *Técnico/Automatización-Acciones planificadas*.

Actividad propuesta 8.3



Configura con una cuenta de Gmail el correo electrónico corporativo de la instancia de Odoo instalada.

8.3. Monitorización

Las instancias de Odoo manejan grandes volúmenes de datos y transacciones. La infraestructura de hardware y software de cada implementación es distinta. Esto hace que la monitorización del rendimiento de las distintas tecnologías involucradas sea fundamental.

Existe una gran variedad de herramientas de monitorización de hardware y servicios que se pueden aplicar a una instalación de Odoo, como Munin, Nagios o Grafana. Además, es fácil encontrar, dependiendo de la versión instalada, módulos de monitorización creados por OCA o por empresas especializadas.

Actualmente odoo.sh incluye herramientas de monitorización basadas en Grafana.

Pero en instalaciones pequeñas y, como es el caso de este libro, en instancias educativas puede ser suficiente el uso de htop para diagnósticos buscando estabilidad y productividad.

Recurso web

www

La utilidad *pgtune* ayuda a configurar el fichero *postgresql.conf*.
Una versión en línea está disponible en
<https://pgtune.leopard.in.ua/#/>

8.4. Rendimiento

Odoo utiliza una serie de tecnologías como Python, PostgreSQL, HTML, CSS, XML, XML-rpc, JavaScript, Nodejs, Werkzeug, etc., cada una con sus particularidades y requerimientos. La optimización del rendimiento de todas y cada una de ellas es fundamental para una eficiente implementación de Odoo ERP.

Es muy importante que la carga de las *vistas* sea fluida, por lo que se hace imprescindible estudiar en cada instalación las cargas de trabajo que se producen en el entorno de producción, es decir, el número de peticiones concurrentes, el consumo de PostgreSQL, que suele ser alto y el aumento del número de usuarios (que implica generalmente el aumento de infraestructura).

Pero ¿cómo debe ser el servidor que gestione una instancia de Odoo? Cualquiera que haya tenido que dimensionar en alguna ocasión un servidor de forma que pueda prestar un servicio determinado, sabe la dificultad que eso conlleva. Se deben tener en cuenta múltiples factores tanto relacionados con el hardware como con el software y no es fácil encontrar una norma infalible.

En el caso de Odoo, hay que tener en cuenta, al menos, el sistema operativo, si la localización de la base de datos está en el mismo servidor, así como, por supuesto, la relación de aplicaciones que se pretenden instalar, el volumen de los datos y el número de usuarios concurrentes. El almacenamiento es otra variable a tener en cuenta.

Odoo trabaja en modo multihilo por defecto, como un gestor de contenidos haciendo peticiones (varias) para cada *vista* solicitada por cada usuario.

Odoo está escrito en Python y GIL hace que Python no tenga capacidad de subprocesamiento: Python está diseñado para que un solo *thread* se ejecute a la vez, por lo que no puede ejecutar diferentes tareas en diferentes procesadores. Por eso, Odoo utiliza el concepto de *worker* para activar el modo multiprocesamiento.

Sin *workers* habilitados, la instancia de Odoo solo utilizaría un *core* de la máquina, lo que se traduciría en tiempo de espera de los siguientes usuarios conectados para utilizar los recursos de Odoo. El sistema de utilizar *workers* produce el mismo resultado que si se lanzasen varias instancias de Odoo en la misma máquina.

En producción, es recomendable cambiar al servidor de multiprocesamiento que optimiza la utilización de recursos del sistema. Para ello, hay que establecer en *odoo.conf* el valor a un número de procesos distinto de cero.

Para dimensionar los servidores, se deben tener en cuenta los siguientes puntos:

- 1 *worker* = 5-6 usuarios concurrentes
- N° de *workers* soportados = 1 + N° Núcleos de procesador × 2
- Cron también necesita 1 *worker*
- Memoria RAM necesaria: en el supuesto de un sistema con 4 núcleos de CPU, 8 GB de memoria RAM y 30 usuarios simultáneos de Odoo, establecida una diferenciación entre solicitudes pesadas y ligeras, que se producen en una relación 20/80 y considerando que las solicitudes pesadas utilicen 1 GB de RAM, mientras que las ligeras utilicen 150 MB de RAM, un dimensionamiento a priori admisible sería:
 - Número máximo de *workers* soportados = 1+ (4 núcleos × 2) = 9
 - Número teórico de *workers* necesarios= 30 usuarios / 6 = 5
 - RAM necesaria = workers x [(light_worker_ratio x light_worker_ram_estimation) + (heavy_worker_ratio x heavy_worker_ram_estimation)]

El consumo de memoria RAM para este número de *workers* es, por tanto:

$$\text{RAM necesaria} = 6 \times [(0,8 \times 150) + (0,2 \times 1024)] \approx 2 \text{ GB de RAM}$$

Es decir, Odoo necesitará un mínimo de 2 GB de RAM de forma exclusiva.

Aunque debe tomarse fundamentalmente como una indicación, pues en la realidad, hay muchos factores que influyen, experiencias previas publicadas arrojan las siguientes conclusiones más o menos ajustadas:

- 1 *worker* = 5-6 usuarios (una CPU por cada 10-12 usuarios concurrentes)
- 1 cron thread por núcleo
- 512 MB RAM por worker
- maintenance_work_mem a 1GB mínimo
- proxy_mode = True
- Puntos medios del rango para los límites:

`limit_time_cpu, limit_time_real, limit_time_real_cron, limit_memory_hard, limit_memory_soft, limit_request.`

Recurso web

www

En Deploying Odoo se encuentra documentación detallada.



PARA SABER MÁS

Normalmente, el número de peticiones a la base de datos provocadas por los distintos usuarios concurrentes será muy elevado. Por eso es muy útil para mejorar el rendimiento de procesamiento en la capa de la base de datos tener en caché los registros de *vistas, ir.ui.view*, para mejorar la velocidad de carga de las páginas. Por ejemplo, *Pagespeed Insights* de Google puede ser utilizado.

Por eso, la utilización de un proxy inverso como Nginx es muy recomendable en instalaciones en producción.



Figura 8.25
Proceso de obtención de los datos de una vista.

8.4.1. Configuración del arranque

Los ficheros de configuración, de servicio y de log fueron introducidos en el capítulo 3. En la instalación realizada eran los ficheros *odoo.conf* en */opt/odoo/odoo*, *odoo.service* en */lib/systemd/system* y *odoo.log* en */var/log/odoo*.

Una vez instalado Odoo por el procedimiento seguido en dicho capítulo, su contenido es el que muestran las siguientes imágenes:

Figura 8.26
odoo.conf

```
\Local\Temp\d48c2774\192.168.1.41\opt\odoo\odoo — Atom
Packages Help
odoo.service x odoo.log
1 [options]
2 admin_passwd = postgres
3
4 xmlrpc = True
5 xmlrpc_port = 8069
6
7 db_host = 127.0.0.1
8 db_port = 5432
9 db_user = odoo
10 db_password = odoo
11
12 addons_path = /opt/odoo/odoo/addons
13
14
```

```

odoo.service -- C:\Users\jigl\AppData\Local\Temp\d48c2774\192.168.1.41\lib\systemd\system -- Atom
File Edit View Selection Find Packages Help
Project Remote untitled odoo.service x
1 [Unit]
2 Description=Odoo 13
3 After=network.target
4
5 [Service]
6 Type=simple
7 User=odoo
8 Group=odoo
9 ExecStart=/opt/odoo/odoo-bin -c /opt/odoo/odoo/odoo.conf --logfile /var/log/odoo/odoo.log
10
11 [Install]
12 WantedBy=multi-user.target
13
14

```

Figura 8.27
odoo.service.

```

odoo.log -- C:\Users\jigl\AppData\Local\Temp\d48c2774\192.168.1.41\var\log\odoo -- Atom
File Edit View Selection Find Packages Help
Project Remote untitled odoo.service x odoo.log x
187 2020-12-20 17:36:47,435 1373 INFO ? werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:47] "GET /web/static/src/xml/
188 2020-12-20 17:36:47,556 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:47] "POST /mail/init_mess
189 2020-12-20 17:36:47,772 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:47] "POST /web/dataset/ci
190 2020-12-20 17:36:47,779 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:47] "POST /web/dataset/ci
191 2020-12-20 17:36:47,832 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:47] "GET /web/image?mode=
192 2020-12-20 17:36:47,899 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:47] "POST /web/dataset/ci
193 2020-12-20 17:36:47,972 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:47] "POST /calendar/notif
194 2020-12-20 17:36:48,028 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:48] "POST /web/action/lo
195 2020-12-20 17:36:48,240 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:48] "POST /web/dataset/ci
196 2020-12-20 17:36:48,326 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:48] "POST /web/dataset/ci
197 2020-12-20 17:36:48,472 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:48] "POST /web/dataset/ci
198 2020-12-20 17:36:49,329 1373 INFO ? werkzeug: 192.168.1.39 - - [20/Dec/2020 17:36:49] "GET /web_tour/static/sr
199 2020-12-20 17:37:28,714 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:37:28] "POST /longpolling/po
200 2020-12-20 17:37:37,870 1373 INFO dam-d werkzeug: 192.168.1.39 - - [20/Dec/2020 17:37:37] "POST /longpolling/po
201 2020-12-20 17:38:06,147 1373 INFO ? odoo.service.server: Initiating shutdown
202 2020-12-20 17:38:06,147 1373 INFO ? odoo.service.server: Hit CTRL-C again or send a second signal to force the
203 2020-12-20 17:38:08,565 1445 INFO ? odoo: Odoo version 13.0
204 2020-12-20 17:38:08,567 1445 INFO ? odoo: Using configuration file at /opt/odoo/odoo/odoo.conf
205 2020-12-20 17:38:08,567 1445 INFO ? odoo: addons paths: ['/opt/odoo/odoo/odoo/addons', '/opt/odoo/.local/share/
206 2020-12-20 17:38:08,568 1445 INFO ? odoo: database: odoo@127.0.0.1:5432
207 2020-12-20 17:38:08,937 1445 INFO ? odoo.addons.base.models.ir_actions_report: Will use the Wkhtmltopdf binary

```

Figura 8.28
odoo.log.

El fichero `odoo.conf` es fundamental para asegurar un rendimiento adecuado del servidor de Odoo.

El comando `./odoo-bin -h` lista los parámetros de configuración del sistema.

La información acerca de los parámetros disponibles de la versión instalada por el procedimiento manual del capítulo 3 se encuentra en `/opt/odoo/odoo/odoo/tools/config.py`

A) Parámetros de inicio

Algunos de los más importantes son:

- `admin_passwd`. Contraseña maestra de la bbdd.
- `init (-i)`. Contiene lista para instalar módulos por consola.
- `update (-u)`. Contiene lista de módulos para actualizar.
- `pidfile`. Ruta pid de Odoo.
- `addons_path`. Todas las rutas absolutas donde se guardan carpetas de módulos separadas por comas.

- *data_dir (-D)*. Ruta donde alojar directorios de datos.
- *config (-c), save (-s), without_demo, import_partial...*

B) Parámetros de log

Algunos de los más importantes son:

- *logfile*. Por defecto none. Se coloca la ruta donde guardar el log de ejecución (en vez de por consola).
- *logrotate*. Rotación diaria por defecto. Habilita la rotación de registros. Configura el módulo de registro para archivar los registros del servidor a diario y para mantener los registros antiguos durante 30 días para evitar que los registros pudieran consumir eventualmente todo el espacio disponible en disco.
- *syslog*. Por defecto, 'false'. Si se pone a 'true' guarda en el log del sistema operativo /var/log/syslog en vez de en la ruta marcada.
- *log_handler* y *log_level*. Configura el nivel de registro. Controlador de la capa mostrada del log para ver quién proporciona información y sobre qué capa. Por ejemplo:

```
log_level = warn
log_handler = :WARNING,werkzeug:CRITICAL,odoo.service.server:INFO
```

solo registrará mensajes con al menos el nivel de ADVERTENCIA (WARNING), excepto para werkzeug (CRITICAL) y odoo.service.server (INFO).

- *log_db*. Habilita los log de la BBDD.
- *log_db_level*. Especifica el nivel de log de la BBDD.

C) Parámetros del servicio XML-RPC/HTTP

Algunos de los más importantes son:

- *xmlrpc-interface*. Vacío= localhost. Si no, ip del servidor donde está instalado Odoo.
- *xmlrpc-port*. Por defecto 8069. Puerto de conexión a Odoo.
- *xmlrpc*. Por defecto True.
- *longpolling-port*. Por defecto 8072. Servicio push para notificaciones y chat.
- *proxy-mode*. Por defecto False.

D) Parámetros relativos a los workers

Por defecto, Odoo maneja las peticiones de los clientes en subprocessos separados. En entornos de producción es bastante usual configurar Odoo para trabajar con *workers*, procesos de trabajo para manejar solicitudes HTTP. Esto tiene fundamentalmente dos ventajas respecto a la

configuración predeterminada: las solicitudes se pueden manejar en paralelo haciendo un mejor uso cuando se dispone de múltiples núcleos, y, además, es posible terminar un worker de forma independiente al resto, si superara un determinado consumo de recursos.

- *limit_memory_hard=4294967296*. Cantidad de memoria máxima asignada a un *worker*. Por defecto es un valor bajo. Es recomendable utilizar 4 GB.

`limit-memory-hard = workers x 768MB` (valor por defecto).

- *limit_memory_soft=671088640*. Si un worker consume más del límite(640 MB), finalizará el proceso después de que se procese la petición actual.

`limit-memory-soft = workers x 640MB` (valor por defecto).

- *limit_request=8192*. Se finalizará el trabajo si se han procesado este número de peticiones.
- *limit_time_cpu=120*. Cantidad máxima de tiempo de CPU para procesar una petición. Valor por defecto bajo.
- *limit_time_real=300*. Cantidad máxima en tiempo real (de reloj) permitido para procesar una solicitud.Valor por defecto bajo.

E) Parámetros de la BBDD si PostgreSQL está instalado en el mismo host

Algunos de los más importantes son:

- *db_host = False*
- *db_maxconn = 64*
- *db_name = odoo*
- *db_password = False*
- *db_port = False*
- *db_template = template1*
- *db_user = False*

En caso contrario, habrá que sustituir FALSE por la configuración de acceso al servidor donde esté instalada PostgreSQL.

La configuración predeterminada de PostgreSQL es generalmente muy conservadora tratando de evitar que la base de datos sature los recursos del sistema pero se puede aumentar de forma segura el valor de algunos parámetros en el archivo postgresql.conf para obtener un mejor rendimiento. Aunque no es objeto de este texto, se relacionan algunos parámetros (PostgreSQL 9.6) interesantes:

- *max_connections = 100*
- *shared_buffers = 256 MB*
- *effective_cache_size = 768 MB*
- *work_mem = 10MB*
- *maintenance_work_mem = 64 MB*
- *checkpoint_segments = 16*

- *min_wal_size* = 2 GB
- *max_wal_size* = 4GB
- *checkpoint_completion_target* = 0.9
- *wal_buffers* = 8MB

F) Otros parámetros clave de PostgreSQL

- *work_mem*: memoria para ordenaciones. Recomendado 64 MB.
- *shared_buffers*: memoria para caché. Recomendado un 25 % de RAM
- *effective_cache_size*: caché de disco. Entre el 50 % y el 75 % de la RAM

Ejemplo: un fichero *odoo.conf* en una instalación puesta en producción podía contener las siguientes líneas:

```
# configuración del log
logfile = /var/log/odoo/odoo.log
logrotate = True
log_level = warn
log_handler = :WARNING,werkzeug:CRITICAL,openerp.service.server:INFO

# parámetros de conexión a la BBDD
db_host = False
db_maxconn = 64
db_name = odoo
db_password = False
db_port = False
db_template = template1
db_user = False

max_connections = 100
shared_buffers = 256 MB
effective_cache_size = 768 MB
work_mem = 10MB
maintenance_work_mem = 64 MB
checkpoint_segments = 16
min_wal_size = 2GB
max_wal_size = 4GB
checkpoint_completion_target = 0.9

#workers
workers = 4
limit_memory_hard = 4294967296 # 4 GB (por defecto 2048x1024x1024)
limit_memory_soft = 671088640 # 640MB (por defecto 2560x1024x1024)
limit_request = 8192
limit_time_cpu = 600
limit_time_real = 1200
```

```
max_cron_threads = 1
# Para que escuche solo en localhost
xmlrpc_interface = 127.0.0.1
netrpc_interface = 127.0.0.1
```

Actividad propuesta 8.4



Cambia el fichero `odoo.conf` para adecuarlo a una instalación del ERP en un servidor Ubuntu con un procesador de cuatro núcleos y 16 GB de RAM que alberga también la BBDD PostgreSQL, incluyendo los parámetros que se estimen oportunos.

8.4.2. Diagnóstico

Para realizar un diagnóstico que permita averiguar en qué capa del sistema (base de datos, procesamiento de datos-rpc , acceso a disco o tráfico de red) se localiza una posible latencia, en `netsvc.py` se encuentran las palabras reservadas que se pueden utilizar para cambiar el nivel de los logs correspondientes a cada capa.

```
210 ✘ DEFAULT_LOG_CONFIGURATION = [
211     'odoo.http.rpc.request:INFO',
212     'odoo.http.rpc.response:INFO',
213     ':INFO',
214 ]
215 ✘ PSEUDOCONFIG_MAPPER = {
216     'debug_rpc_answer': ['odoo:DEBUG', 'odoo.sql_db:INFO', 'odoo.http.rpc:DEBUG'],
217     'debug_rpc': ['odoo:DEBUG', 'odoo.sql_db:INFO', 'odoo.http.rpc.request:DEBUG'],
218     'debug': ['odoo:DEBUG', 'odoo.sql_db:INFO'],
219     'debug_sql': ['odoo.sql_db:DEBUG'],
220     'info': [],
221     'warn': ['odoo:WARNING', 'werkzeug:WARNING'],
222     'error': ['odoo:ERROR', 'werkzeug:ERROR'],
223     'critical': ['odoo:CRITICAL', 'werkzeug:CRITICAL'],
224 }
```

Figura 8.29
Fichero `netsvc.py`.

Con el siguiente comando aparece el nivel INFO (por defecto):

```
# tail -f /var/log/odoo/odoo.log
```

En `odoo.conf` se puede cambiar el nivel del log (log level) por cualquiera de ellos (no olvidar reiniciar el servicio odoo con `#service restart odoo` después de hacer los cambios).

```

2021-01-08 18:42:19,546 810 INFO ? odoo: database: odoo@127.0.0.1:5432
2021-01-08 18:42:20,135 810 INFO ? odoo.addons.base.models.ir_actions_report: Will use the wkhtmltopdf binary at /usr/local/bin/wkhtmltopdf
2021-01-08 18:42:20,883 810 INFO ? odoo.service.server: HTTP service (werkzeug) running on srv10x:80
69
Jjgl-admin@srv10x:~$ tail -f /var/log/odoo/odoo.log
2021-01-08 18:25:21,893 792 INFO ? odoo.addons.base.models.ir_actions_report: Will use the wkhtmltopdf binary at /usr/local/bin/wkhtmltopdf
2021-01-08 18:25:22,368 792 INFO ? odoo.service.server: HTTP service (werkzeug) running on srv10x:80
69
2021-01-08 18:27:14,805 792 INFO ? odoo.service.server: Initiating shutdown
2021-01-08 18:27:14,806 792 INFO ? odoo.service.server: Hit CTRL-C again or send a second signal to force the shutdown.
2021-01-08 18:42:19,544 810 INFO ? odoo: Odoo version 13.0
2021-01-08 18:42:19,545 810 INFO ? odoo: Using configuration file at /opt/odoo/odoo/odoo.conf
2021-01-08 18:42:19,545 810 INFO ? odoo: addons paths: ['/opt/odoo/odoo/odoo/addons', '/opt/odoo/local/share/odoo/addons/13.0']
2021-01-08 18:42:19,546 810 INFO ? odoo: database: odoo@127.0.0.1:5432
2021-01-08 18:42:20,135 810 INFO ? odoo.addons.base.models.ir_actions_report: Will use the wkhtmltopdf binary at /usr/local/bin/wkhtmltopdf
2021-01-08 18:42:20,883 810 INFO ? odoo.service.server: HTTP service (werkzeug) running on srv10x:80
69

```

Figura 8.30
Nivel Info de odoo.log.

Supuestos prácticos

1. Los permisos se aplican a modelos y se asignan a grupos de usuarios. La Aplicación *Compra* de Odoo tiene originalmente dos grupos de usuarios: *Administradora* y *Usuario*.
El ejercicio consiste en crear un nuevo grupo de usuarios llamado *Becario*. De esta manera, sus integrantes solo podrán leer y modificar *Pedidos de compra*, teniendo restrin-gidas las opciones de crear y borrar.
2. *Backup* y recuperación de Odoo con *pgAdmin4*.
 - a) Realiza una copia de seguridad de una BBDD que tenga datos (por ejemplo, los datos demo que proporciona Odoo al crear una nueva BBDD o cualquiera otra no vacía de la que se disponga).
 - b) Entra en la opción “*More Details*” y explicar el comando que se ejecuta y sus parámetros.
 - c) Crea una nueva BBDD vacía y restaura la copia de seguridad sobre ella.
 - d) Comprueba que ambas operaciones se han realizado correctamente. Consulta la documentación de ayuda.

Como opción, se puede realizar el mismo *backup* con otras modalidades.
3. *rclone* es una herramienta de línea de comandos para sistemas Unix y Windows, de código abierto, que permite sincronizar datos con los proveedores de alojamiento en la nube.
 - a) Descarga la aplicación *rclone* desde *rclone.org*.
 - b) Haz pruebas de configuración y funcionamiento con algunos ficheros y carpetas.
 - c) Con las explicaciones proporcionadas en “Para saber más” del apartado 8.2.1 y los conocimientos adquiridos sobre el funcionamiento de *rclone* en el apartado anterior, realiza una configuración para que las copias de seguridad automatizadas realizadas con *pg_dump* se sincronicen en la nube, en una cuenta de Dropbox, GoogleDrive, Mega, Owncloud, OneDrive o similar.

Resumen

- Es parte de la administración básica la gestión de usuarios y de plantillas.
- Un usuario es cualquiera que acceda a la aplicación, ya sea empleado, contacto, cliente o proveedor.
- El control de acceso a los distintos modelos y vistas (permisos) se asocia a los grupos de usuarios.
- Los permisos pueden ser heredados.
- Mediante las *Reglas* se afinan los permisos.
- En Odoo existe un superusuario al que no se le aplica ninguna restricción de acceso.
- Todos los módulos de Odoo proporcionan plantillas de documentos prediseñadas.
- Las tareas relacionadas con el mantenimiento de las bases de datos, la gestión del correo corporativo y la optimización para un mayor rendimiento se consideran parte de una administración avanzada.
- La gestión de la base de datos se puede hacer desde el Database Manager accesible desde la pantalla inicial de conexión.
- Las copias de seguridad pueden realizarse desde el propio Database Manager.
- pgAdmin4 también proporciona una herramienta de backup que permite hacer hasta cuatro tipos distintos de backup utilizando el comando pg_dump.
- Desde el propio sistema operativo se pueden hacer los *backups*. En el caso de Ubuntu se puede utilizar el comando pg_dump y mediante cron programar la automatización de las copias en el tiempo.
- Existen herramientas de terceros que permiten exportar los *backups* realizados a otra ubicación.
- Odoo permite la configuración del correo corporativo para su uso en distintas actividades de los módulos instalados del ERP.
- La puesta en producción de una instancia de Odoo y su uso por un grupo más o menos numeroso de usuarios hace necesario un importante trabajo de monitorización y optimización del rendimiento.
- El fichero de configuración de odoo es fundamental para el rendimiento del servidor
- En odoo.conf (o el nombre que se haya decidido para este fichero durante la instalación) se pueden incluir parámetros de inicio, de log, de PostgreSQL, de workers, etc., buscando la optimización del rendimiento del ERP.

ACTIVIDADES DE AUTOEVALUACIÓN

1. En Odoo los usuarios pueden ser (señala todas las respuestas correctas):

- a) Público.
- b) De portal.
- c) Proveedor.
- d) Cliente.

2. El control de acceso de usuarios está en el modelo:

- a) models.py.
- b) ir.model.security.
- c) ir.model.access.
- d) res.partner.access.

3. Los permisos que permiten cambiar botones y acciones de ventana al modelo Producto son los asignados:

- a) product.product
- b) product.product.access
- c) product.attribute
- d) product.template

4. ¿Cualquier usuario puede insertar datos en una aplicación de Odoo?:

- a) Solo puede insertar datos el usuario Administrador.
- b) Cualquiera, siempre que pertenezca a un grupo que tenga permisos para ello.
- c) Sí, salvo si es el usuario invitado.
- d) Para insertar datos hay que activar el modo desarrollador.

5. En versiones de Odoo superiores a la 13, el usuario System(OdooBot) es:

- a) El superusuario.
- b) El usuario del modo desarrollador.
- c) El usuario Master.
- d) El usuario Administrador.

6. Los *backup* de las bases de datos de Odoo se pueden hacer:

- a) Desde la página de acceso a la instancia de Odoo instalada.
- b) Con comandos y utilidades del sistema operativo.
- c) Con pgAdmin4.
- d) Todas las opciones son correctas.

7. Para utilizar el correo corporativo de una empresa en Odoo:

- a) Es necesario instalar Odoo en un ISP.
- b) Hay que configurar los servidores de correo.
- c) Se utilizan servidores de correo compatibles con Odoo como Microsoft Exchange, Zimbra y similares.
- d) Ninguna de las opciones es correcta.

8. Con el uso de los *workers*:

- a) La instancia de Odoo solo utiliza 1 core del procesador.
- b) Se reduce el número de peticiones que recibe el servidor.
- c) Se obtiene el mismo resultado de lanzarse varias instancias de la aplicación.
- d) Se corre el peligro de aumentar los tiempos de espera.

9. ¿Un usuario interno puede administrar al resto de usuarios en un sistema Odoo?:
- a) Puede modificarlos pero no crearlos.
 - b) Puede crearlos con permisos básicos, pero no modificarlos.
 - c) Solo el usuario *Administrator*.
 - d) Cualquier usuario puede administrar al resto entrando en la opción “Modo desarrollador”.
10. Por defecto, la frecuencia de comprobación y envío de correos en Odoo es de:
- a) 10 minutos.
 - b) 1 hora.
 - c) 1 minuto.
 - d) A demanda.

SOLUCIONES:

1. **a** **b** **c** **d**
2. **a** **b** **c** **d**
3. **a** **b** **c** **d**
4. **a** **b** **c** **d**

5. **a** **b** **c** **d**
6. **a** **b** **c** **d**
7. **a** **b** **c** **d**
8. **a** **b** **c** **d**

9. **a** **b** **c** **d**
10. **a** **b** **c** **d**