



3 Organización, consulta e tratamento da información

Sumario

3 Organización, consulta e tratamento da información.....	1
3.1 Convencións empregadas.....	5
3.2 Organización e consulta da información.....	6
3.2.1 Táboas e vistas da base de datos.....	6
3.2.2 Consultas de acceso a datos.....	8
3.3 Visualización da información.....	9
3.3.1 Interfaces de entrada de datos e de procesos. Formularios e Gráficos.....	11
3.3.2 Definición de campos.....	12
3.3.3 Menús.....	13
3.3.4 Procura de información.....	14
3.3.5 Informes e listaxes da aplicación.....	15
3.4 Tratamento da información.....	15
3.4.1 Cálculos: pedidos, albarás, facturas, asentos predefinidos, rastrexabilidade e producción.....	16
3.4.2 Procedementos almacenados de servidor.....	17
3.5 Extracción de datos en sistemas de ERP-CRM e almacéns de datos.....	18
3.5.1 Importar e exportar datos.....	19
3.6 Avaliación do rendemento e auditorías de acceso aos datos.....	19
3.6.1 Monitorización e avaliación do rendemento.....	20
3.6.2 Auditorías de control de acceso aos datos. Trazas do sistema (logs).....	21
3.6.3 Incidencias: identificación e resolución.....	23

Índice de figuras










Figura 1: Acceso a modelos.....	7
Figura 2: pgAdmin.....	9
Figura 3: Código da nova vista creada.....	10
Figura 4: Personalización de vistas.....	11
Figura 5: Personalización de menús.....	13
Figura 6: Procura de contactos en Odoo.....	14
Figura 7: Acceso a usuarios e compañías.....	16
Figura 8: Utilización da CPU.....	21
Figura 9: Modificación do ficheiro sysstat en linux.....	22
Figura 10: Ficheiro de rexistros de Odoo.....	23



Material docente elaborado a partir da base dos materiais formativos de FP en liña propiedade do Ministerio de Educación e Formación Profesional.

[Aviso Legal](#)

3.1 Convencións empregadas

	Notas de introdución
	Aclaración
	Arquivos de configuración, de rexistro...
	Casos de uso
	Código fonte
	Avisos ou advertencias
	Capturas de pantalla, imaxes
	Actividades.
	Enlace recomendado

	Iconas proporcionadas por Papyrus Development Team https://github.com/PapyrusDevelopmentTeam/papyrus-icon-theme
---	---

3.2 Organización e consulta da información


Ata o de agora, vimos as funcionalidades básicas de dúas ERPs moi estendidos no mercado. En diante centrarémonos principalmente nunha destas dúas aplicacións para poder explorar máis a fondo todas as posibilidades dun ERP. Basicamente por posuír un módulo basee un tanto máis sinxelo formado tan só por Empresas e Administración, centrarémonos en Odoo .

A base de datos dun sistema ERP é de gran envergadura. Almacena as táboas cos datos da aplicación, vistas das diferentes táboas e outros elementos como funcións ou disparadores que realizan operacións sobre os datos. Por iso, debido a esta gran cantidade de información almacenada, faise necesario unha organización entre os seus compoñentes.

O que se fai é establecer unha serie de normativas ou nomenclatura para organizar a información, que os desenvolvedores deben seguir á hora de modificar o código fonte ou o esquema da base de datos. Por exemplo, incluír un prefixo nos compoñentes da base de datos, para saber a que módulo pertencen, ou establecer unha serie de campos dentro dunha táboa como obrigatorios, para poder asegurar o funcionamento correcto da aplicación.

Nos sistemas de planificación empresarial desenvoltos nunha linguaxe orientada a obxectos, calquera dato é accesible a través de obxectos. Por exemplo, en Odoo temos un obxecto `res.partner` para acceder aos datos concernentes aos colaboradores ou socios, un obxecto `account.invoice` para os datos de factúralas etc. Como ves, ambos van precedidos dun prefixo que indica o módulo ao cal pertencen.

Calquera método que queira actuar sobre un obxecto deberá ter un parámetro que indique sobre que recurso ou rexistro dentro do obxecto quérese actuar. Por exemplo, se queremos enviar un correo electrónico aos colaboradores identificados como 1 e 5 dentro da táboa `res.partner`, utilizaremos a seguinte instrución:

	Enviar un correo electrónico
<pre>res.partner.send_email(..., [1,5], ...)</pre>	

Chámase así ao conxunto de ordes que se executan cando se cumpre unha condición determinada ao realizar unha operación nunha base de datos, para inserir, modificar ou eliminar información.

En resumo, un obxecto é, nun concepto amplo, todo elemento que forma parte da aplicación e que permite acceder aos datos da mesma.

3.2.1 Táboas e vistas da base de datos

Como explicamos, **un obxecto garda todo elemento que forma parte da aplicación**. Como elementos da aplicación temos a propia base de datos (táboas, disparadores ou

accións sobre esas táboas, etc), os formularios, os informes, as xanelas ou os asistentes, por exemplo.

En Odoo accédese aos obxectos a través da opción **Axustes-> Técnico-> Estrutura da base de datos-> Modelos**. Nesta opción podemos atopar toda a información que manexa a aplicación.

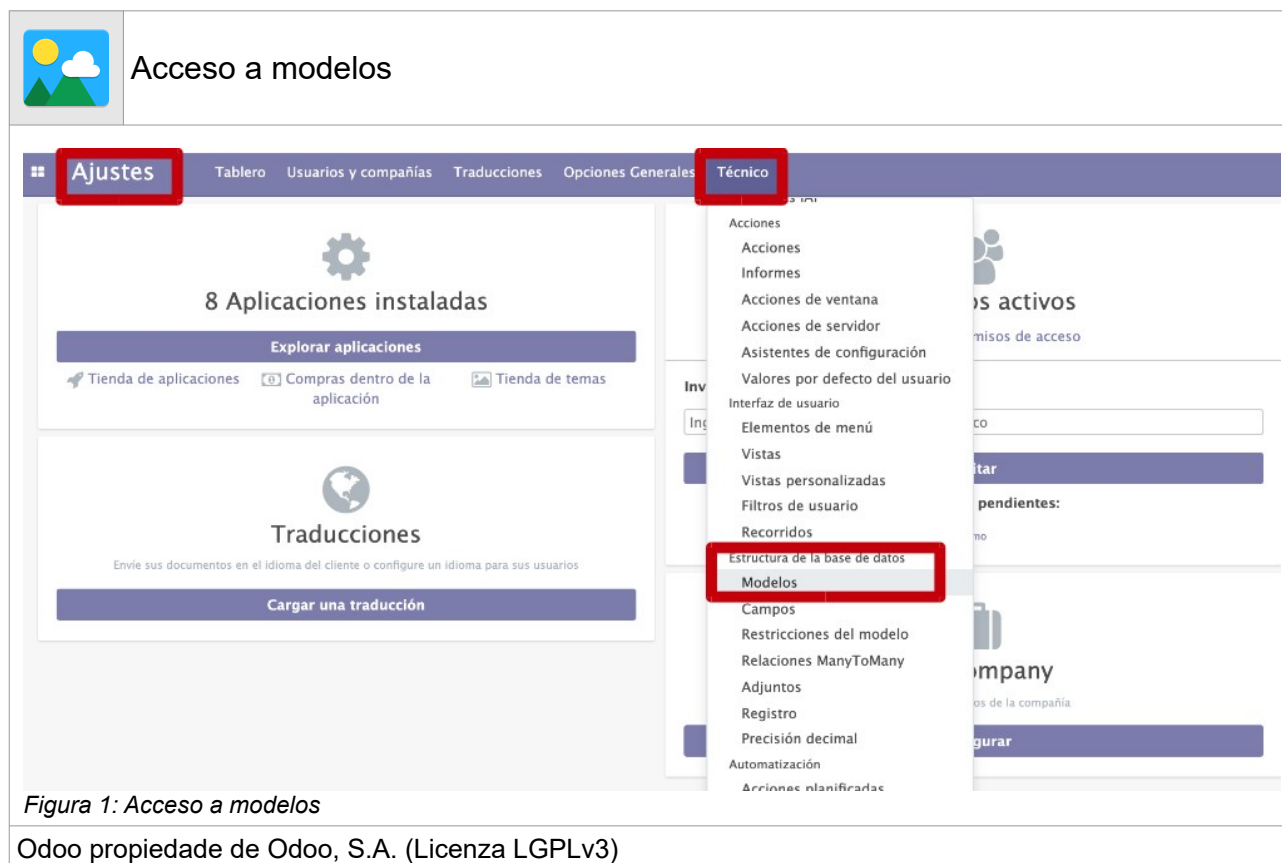


Figura 1: Acceso a modelos

Odoo propiedad de Odoo, S.A. (Licenza LGPLv3)

En Openbravo todos os obxectos cos que traballa a aplicación, chamados metadatos, están dispoñibles no módulo Diccionario de Datos, cuxa función é administrar estes metadatos dunha maneira centralizada. Neste módulo danse de alta os formularios, informes e procesos que se utilizan na aplicación, aos que se accederán mediante menús ou botóns en xanelas, así como as táboas e campos (ou columnas) que as forman.

Lembra que para entrar no Odoo debemos utilizar a dirección IP do servidor seguido do porto de comunicación, é dicir, se estamos a traballar en local (localhost:8069), é o porto por defecto. Se traballamos en remoto (a.b.c.d:8069) sendo as letras a, b, c e d fan referencia á dirección IP del ordenador onde está instalado o Odoo, loxicamente tamén se pode escribir o nome do computador seguido do porto.

Obxectos principais en Odoo

- **Modelos:** As táboas coas que traballa a aplicación
- **Vistas:** Os distintos tipos de formularios cos que mostra os datos dos modelos.

- **Menús:** A estrutura de menús coas que chamaremos ás accións para manexar os datos dos modelos da aplicación.
- **Accións:** Os métodos desde os que imos abrir nosas vistas para traballar na aplicación.

Dentro dos obxectos distinguimos as táboas. **Unha táboa é unha estrutura de datos organizada en filas e columnas, de maneira que cada columna é un campo (ou atributo) e cada fila un rexistro.** Por exemplo, no modelo `res.users` cada rexistro correspóndese cun usuario, e para cada usuario gárdanse unha serie de atributos como nome, login ou password.

En ocasións, a base de datos está formada por tantas táboas e obxectos que se volve complexa e difícil de manexar. Neses casos, interesa que algúns grupos ou perfís de usuarios teñan unha vista parcial deses datos. Para estes casos utilízanse as vistas. **Unha vista é basicamente unha "táboa virtual" á que se pode acceder coma se fose unha táboa do esquema, pero que realmente non o é.** Teñen a mesma estrutura que as táboas: filas e columnas ou campos, e pódese acceder a elas da mesma forma, a través de consultas de acceso a datos como veremos posteriormente.

Para administrar a base de datos podemos conectarnos directamente ao xestor da base de datos, aínda que o normal é utilizar ferramentas gráficas que nos faciliten o traballo. Entre as ferramentas para administración de bases de datos PostgreSQL atópase **PGAdmin 4**.

Unha vez instalado PgAdmin 4, conectaremonos á base de datos introducindo os seguintes datos:

- **Nome:** nome que queiramos darlle á conexión.
- **Servidor:** Dirección IP ou nome do servidor.
- **Porto:** Número de porto para a conexión, normalmente o 5432.
- **Base de datos de Mantemento:** base de datos inicial coa que nos conectamos, normalmente chamada "postgres".
- **Nome de usuario:** usuario co que queiramos conectarnos, no noso caso, debemos introducir o usuario que creamos na segunda unidade, concretamente, o usuario "userbd".
- **Contrasinal:** clave do usuario.

3.2.2 Consultas de acceso a datos

As consultas de acceso a datos permítennos acceder á información que gardan as táboas e vistas da base de datos. As consultas de acceso a datos serven para indicar ao sistema de xestión da base de datos que devolva un extracto da información en forma dun conxunto de rexistros.



Logotipo de pgAdmin, ferramenta para o manexo de bases de datos.



Figura 2: pgAdmin

pgAdmin Community (CC BY-SA)

Os pasos para crear unha consulta son:

- Seleccionar as táboas ou vistas sobre as que vai actuar a consulta.
- Establecer a relación entre as táboas e vistas, no caso de que a aplicación non a proporcione.
- Seleccionar os campos para mostrar na consulta.
- Executar a consulta.

As consultas poden actuar sobre unha ou varias táboas ou vistas, e pódense gardar para ser utilizadas posteriormente. En ocasións a aplicación permite realizar consultas de acceso a datos, ou ben podemos conectarnos directamente ao sistema xestor de base de datos.

As consultas de acceso a datos pódense construír escribindo o código na linguaxe de consulta utilizada, por exemplo SQL, ou ben mediante asistentes e construtores gráficos se se trata de consultas pouco complexas.

3.3 Visualización da información

Dado que todos os datos do programa están almacenados en obxectos, como se mostran devanditos obxectos ao usuario? **Tanto a información das táboas como a de calquera outro obxecto da aplicación móstrase a través de interfaces.**

Cada obxecto ten a súa propia interface, por exemplo, non se mostran da mesma maneira os datos das Empresas que os datos dunha Factura. As interfaces poden ser:

- Estáticas: créanse dentro do código da aplicación e non poden ser modificadas.
- Dinámicas: poden ser modificadas por parte do usuario, para o que se almacena a descrición da vista nunha linguaxe de descrición de datos que permita a súa modificación, por exemplo **XML**.

Por tanto, as interfaces dinámicas son construídas de forma dinámica pola descrición XML da pantalla do cliente. Para iso non é necesario ser uns expertos nesa linguaxe, podemos facer obxectos sinxelos simplemente tomando como exemplo outros obxectos que haxa creados na aplicación. Con todo, moitas aplicacións provén a forma de crear as descricións de maneira gráfica sen necesidade de introducir código manual.



O seguinte código é un exemplo de descrición dun obxecto en Odoo:

```
<?xml version="1.0" encoding="utf-8"?>
<form string="Partner">
  <field name="name"/>
  <field name="title"/>
  <field name="ref"/>
  <field name="lang"/>
  <field name="customer"/>
  <field name="supplier"/>
</form>
```

Co código anterior estamos a crear unha interfaz para a introdución e consulta de datos. En concreto, móstranse seis campos correspondentes ao nome, título ou tratamento en caso de ser unha persoa física, código, idioma, e por último dous campos booleanos que gardan se se trata dun cliente ou dun provedor. Esta definición almacenarase nun arquivo XML que, ao abrilo desde a aplicación, mostrará o obxecto resultante da definición anterior:



Aplicación onde se pode ver a nova vista creada

Figura 3: Código da nova vista creada

Odoo propiedad de Odoo, S.A. (Licenza LGPLv3)

3.3.1 Interfaces de entrada de datos e de procesos. Formularios e Gráficos.

Podemos definir unha ou varias interfaces para cada obxecto ou tipo de recurso, para describir que campos móstranse e como. Dependendo de como se distribúan os campos atopámonos, entre outras, cos seguintes tipos de interfaces:

- **Formularios:** neste tipo de interfaces móstrase un só rexistro, cuxos campos se distribúen na pantalla seguindo sempre o mesmo criterio, normalmente de esquerda a dereita e de arriba a abaixo, de acordo con a orde no que son descritos na vista.
- **Árbores:** este tipo de interfaces utilízase cando queremos mostrar un conxunto de rexistros en modo lista, e é útil para mostrar varios rexistros á vez e realizar procuras sobre eles.
- **Gráficos:** os gráficos son outra forma de ver os datos dun formulario, e poden mostrarse en varios formatos e tipos para unha mellor visualización da información.

Odoo chama ás interfaces **Vistas**, e desde a aplicación podémolas personalizar de dúas maneiras diferentes:

- Escribindo código XML. Editamos a vista e escribimos directamente o código. Isto facémolo a desde o menú Axustes/Técnico/Interface de usuario/Vistas. (Lembra que tes que ser superusuario)
- Accedendo ao módulo e modificando o ficheiro XML correspondente. O roteiro onde se atopan as aplicacións é diferente de linux a windows, en linux o roteiro é `/usr/lib/python3/dist-packages/odoo/addons` en windows atópase en `/archivos de programa/odoo/addons`

Unha vez modificada a vista, é recomendable actualizar a aplicación desde o menú Aplicacións, buscas a túa aplicación e nos tres puntos da súa parte superior dereita pulsas e seleccionas Actualizar.



Figura 4: Personalización de vistas

3.3.2 Definición de campos

Ao principio do apartado vimos un exemplo de definición dun obxecto. Diciamos que os obxectos están definidos en arquivos XML. Un obxecto está formado por campos. Os campos son a información que se mostra en pantalla. Por exemplo, unha vista de tipo formulario con dous campos nomee e título definiríamola así:



Vista de tipo formulario con dous campos nomee e título

```
<?xml version="1.0"?>
<form string="Empresa">
  <field name="name"/>
  <field name="title"/>
</form>
```

No exemplo anterior, utilizamos distintos tipos de etiquetas para describir a vista, A etiqueta form indica que imos describir un formulario e dentro dela utilízase a etiqueta field para cada un dos campos que o compoñen. As etiquetas empezan co símbolo < e terminan co símbolo >. Nas vistas de tipo formulario, os campos van precedidos por unha etiqueta co seu nome e colócanse de esquerda a dereita, na orde con que son declarados no arquivo XML .

Hai un tipo especial de campo que son os campos de relación one2many. Estes campos utilízanse para reflectir unha relación dun-a-moitos entre dous obxectos. Por exemplo, o campo address dentro do obxecto res.partner é un campo one2many o cal quere dicir que ese campo está enlazado con outro obxecto, en leste caso res.partner.address, isto é, que un rexistro de res.partner pode ter moitos rexistros relacionados en res.partner.address. Noutras palabras, unha empresa pode ter moitas direccións tamén chamadas contactos da empresa, e esas direccións gárdanse na táboa res.partner.address. Á hora de describir este tipo de campos no arquivo XML utilizaremos a sintaxe:



Campos

```
<field mode="form,tree" name="address" >
<dentro poñeremos os campos que forman parte de
res.partner.address>
</field>
```

No menú **Axustes/Técnico/Estrutura da base de datos/Modelos** podemos ver os campos do obxecto Empresa(res.partner), e veremos que un deles é o campo address de tipo One2many.

Ademais dos campos, existen varios elementos de deseño que nos permiten personalizar as vistas form e tree. Algúns deles son:

- **mode:** tipo de vistas que vai permitir o obxecto.


- **name:** nome do campo tal e como aparece no obxecto.
- **separator:** agrega unha liña de separación no formato, exemplo:
 - `<separator string="link" colspan="4"/>`
 - O atributo **string** define a etiqueta do separador e o atributo **colspan** define o seu tamaño .
- **notebook:** permite distribuír os campos da vista en diferentes pestanas ou tabs que van definidos por paginas, exemplo:
 - `<notebook colspan="4">....</notebook>`

Por outra banda, á etiqueta field pódenselle engadir atributos. Algúns deles son:

- **colspan="4":** el numero de columnas polas que se pode estender un campo .
- **readonly="1":** establece un campo como só lectura .
- **invisible="True":** ouculca o campo e a súa etiqueta .
- **password="True":** reemplaza a entrada dun campo cun símbolo "•" .
- **string="":** la etiqueta que vai aparecer xunto ao campo.

3.3.3 Menús

Para acceder aos obxectos que creamos na aplicación, deben definirse previamente os menús así como as accións asociadas aos mesmos.



Personalización de m enús

Ajustes

Tablero

Usuarios y compañías

Traducciones

Opciones Generales

Técnico

Modelos / Contacto

Editar

Crear

Descripción del modelo

Modelo

Modelo transitorio

Hilo de mensajes

Contacto

res.partner

☐

☒

Campos

Permisos de acceso

Reglas de

Nombre de campo	Etiqueta de cam
__last_update	Última modificaci
active	Activo
activity_date_deadline	Siguiente plazo d
activity_ids	Actividades
activity_state	Estado de la activ

Seudónimos

Canales

Canales/socios

Correos de Digesto

IAP

Cuentas IAP

Acciones

Acciones

Informes

Acciones de ventana

Acciones de servidor

Asistentes de configuración

Valores por defecto del usuario

Interfaz de usuario

Elementos de menú

Vistas

Vistas personalizadas

Filtros de usuario

Recorridos

Estructura de la base de datos

Modelos

Tipo

En las aplicaciones

Objeto base

account, auth_signi

calendar, crm, mail

payment, portal, pr

sms, snailmail, stor

Tipo de campo	Requerido
Fecha y hora	<input type="checkbox"/>
booleano	<input type="checkbox"/>
fecha	<input type="checkbox"/>
one2many	<input type="checkbox"/>
Selección	<input type="checkbox"/>

Os menús pódense editar desde Axustes/Técnico/Interface de usuario/Elementos de Menú. Desde aí podemos modificar o menú seleccionado ou crear novos menús. Para crear menú secundario seguiremos o mesmo proceso, tan só teremos que indicar cal é o menú pai do cal depende.

Os menús teñen asociada a acción que deben realizar. Hai diferentes tipos de accións, as máis importantes:

- `ir.actions.act_window`: Abre unha vista nunha nova xanela.
- `ir.actions.report`: Imprime un informe.

Por tanto, cando creamos un menú, debemos indicar o tipo de acción que ten asociada. Se a acción non está creada debemos utilizar a ligazón **Crear...** para creala. Os campos para introducir variarán segundo a acción de que se trate. Por exemplo, para crear unha nova acción que abra unha vista debemos introducir polo menos os seguintes datos:

- **Nome da acción.**
- **Tipo da acción**, neste caso, `ir.actions.act_window`.
- **Datos da vista**: tipo (árbore, formulario etc.)/ etc.) e nome.

3.3.4 Procura de información

Un aspecto moi importante dunha aplicación ERP é como leva a cabo a procura de información. Os motores de procura deben ser o suficientemente potentes para que o proceso sexa rápido e efectivo.

Normalmente inclúense procuras básicas sobre os campos principais, e procuras avanzadas para buscar rexistros por campos máis específicos.



A opción de procura está dispoñible en cada xanela da aplicación. No caso de Odoo, os criterios de procura introdúcense na parte superior da xanela e debaixo sitúase unha vista de tipo árbore coa listaxe dos rexistros do obxecto.

A listaxe reducirase aos rexistros coincidentes cos criterios de procura. Tamén é posible aplicar filtros á selección. Os rexistros poden ser seleccionados e cambiando ao modo formulario poderemos ver todos os seus datos, así como editar o seu contido.

3.3.5 Informes e listaxes da aplicación

Os informes e listaxes da aplicación son unha forma de mostrar os datos para mellorar a súa visualización. Moitos informes e listaxes están creados, e son accesibles desde os distintos menús da aplicación.

Algúns destes documentos poden ser:

- Informes contables.
- Albarás.
- Pedidos.
- Recibos.
- Reclamacións a provedores e/ou clientes.
- Informes estatísticos.
- Xeración de etiquetas para un conxunto de rexistros.
- Informes de agrupamento: permiten mostrar os rexistros existentes para un mesmo valor dun campo.
- Impresión de documentos do sistema.

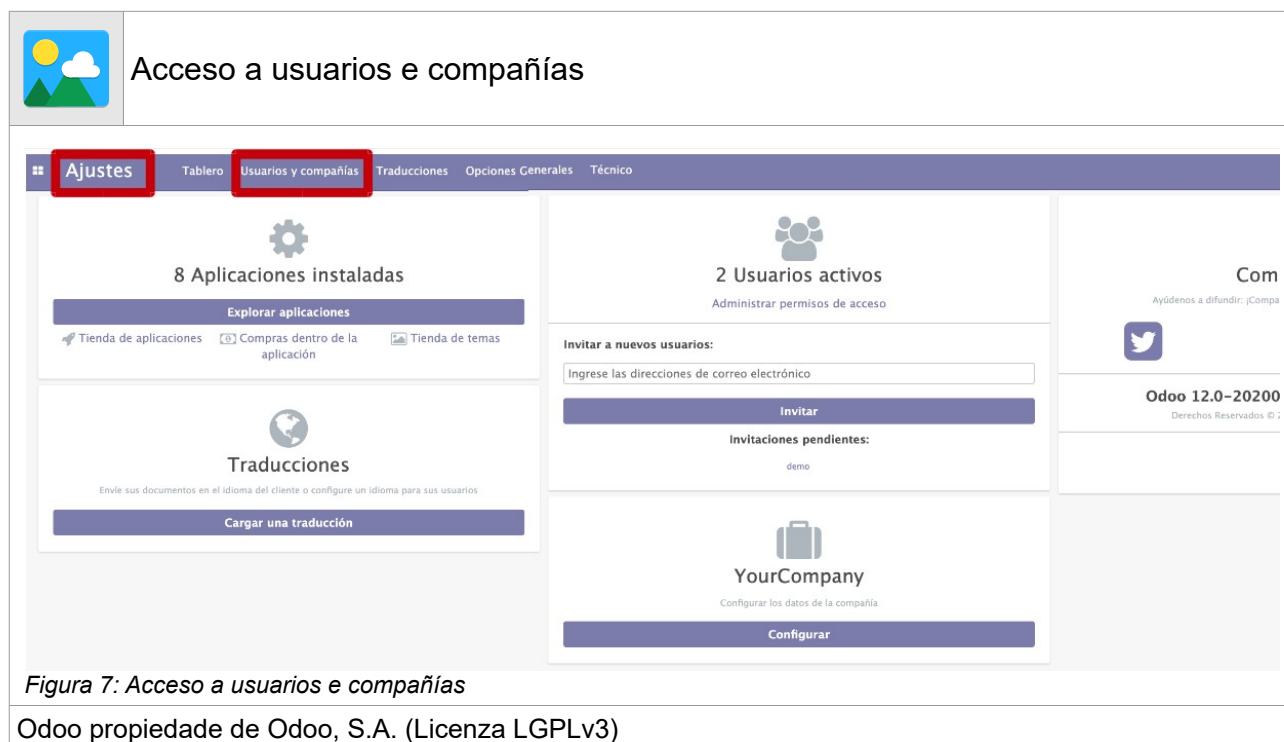
Os informes e listaxes que incorpora a aplicación pódense utilizar tal cal se proporcionan, ou ben adaptar o seu deseño á imaxe corporativa da empresa. Tamén se poden engadir novos informes instalándoos como módulos independentes. No caso de Odoo, os módulos extra que conteñen exclusivamente informes dalgún tipo levan no seu nome a etiqueta "report".

3.4 Tratamento da información

O tratamento da información leva a cabo a través de distintos procesos. Antes de poder realizar ningún proceso é necesario introducir a información propia da compañía. Por exemplo, se imos levar a contabilidade da empresa haberán de configurarse as contas contables, teranse que definir as contas bancarias e os diarios de compras, vendas e caixa ou banco. Tamén haberán de definirse os impostos.

Por exemplo, se temos instalado o módulo de contabilidade, cando cremos unha empresa cliente ou provedor, deberemos asociarlle unha serie de contas contables, como as de cobranza ao cliente e pago ao provedor, polo que deberemos ter creadas ditas contas antes de poder dar de alta aos nosos clientes e provedores, e igual cos produtos.

Dentro da información propia da compañía deberemos crear os clientes, provedores e produtos. En calquera dos casos, existe unha información mínima que teremos que introducir antes de poder empezar a traballar.



Ao instalar o perfil mínimo de Odoo deberemos comprobar se existen datos, e se non, introducilos, de polo menos os seguintes obxectos ou táboas:

- **Título:** Fai referencia ao tipo de empresa ou ao tratamento que se lle debe outorgar a unha persoa de contacto. Se é un ou outro se distingue a través do campo Dominio que hai na táboa, que será igual a "Empresa" no primeiro dos casos ou a "Contacto" no segundo caso.
- **Función:** Listaxe de cargos ou funcións das persoas de contacto.
- **Provincias:** Listaxe de provincias por país.

Esta información é necesaria para poder introducir os datos das empresas, e podemos acceder a ela desde o menú **Axustes/Usuarios e Compañías/Compañías**.

3.4.1 Cálculos: pedidos, albarás, facturas, asentos predefinidos, rastrexabilidade e produción.

Entre os procesos para realizar por un sistema de planificación empresarial destacan os seguintes:

- **Contabilidade.** Inclúe os procesos onde se reflicten as operacións económicas realizadas pola empresa, a determinación dos custos da empresa e os orzamentos do exercicio fiscal. Proporcionanse asentos predefinidos para a introdución rápida de asentos sen necesidade de ter coñecementos de contabilidade.

- **Operacións de compra:**
 - Crear unha orde de compra ou pedido de compra.
 - Recibir os bens.
 - Controlar a factura de compra.
 - Rexistrar o pago ao proveedor.
- **Operacións de venda:**
 - Crear unha orde de venda ou pedido de venda e recibir a conformidade do cliente.
 - Preparar os bens para enviar ao cliente e realizar o albará e a entrega.
 - Realizar a factura de venda.
 - Rexistrar a cobranza ao cliente ou pago do cliente.
- **Rastrexabilidade:** chámase así ao proceso da entrada do produto ata a saída do mesmo.

É posible que polas necesidades da empresa non sexa necesario utilizar todos os procesos do ERP, por exemplo, pode ser usado só como un CRM, ou só como un programa de contabilidade, pero a súa verdadeira potencia alcánzase coa integración de todas as súas funcións.

3.4.2 Procedementos almacenados de servidor

Dentro do tratamento que lle damos á información, ás veces é necesario que leven a cabo determinadas tarefas de forma automática, en resposta a algo que pasa na aplicación. Por exemplo, podemos querer enviar un correo electrónico ao cliente de maneira automática, cada vez que nos fai un pedido, ou ben rexistrar na ficha do cliente o pedido realizado. Ambas as accións: enviar o correo ou rexistrar o pedido no histórico do cliente, son accións que se desencadean automaticamente polo sistema, e a acción que as desencadea é a realización dun pedido por parte dun cliente.

Para facer isto na maioría dos sistemas podemos utilizar:

- **Procedementos almacenados de servidor.** Un procedemento almacenado é un programa, procedemento ou función almacenado nunha base de datos e listo para ser usado. Poden ser executados directamente polo usuario ou ben cando se cumpra unha determinada condición a través de disparadores.
- **Eventos de servidor.** Un evento de servidor consiste en detectar que pasa algo na aplicación e facer que o sistema responda a este suceso de forma automática. Os eventos de servidor son creados a nivel de obxectos e non de base de datos, como ocorre nos procedementos. Podemos definir os eventos de servidor a través dos menús da aplicación, non sendo necesario introducirmos na base de datos para programar a acción desexada.

Un procedemento almacenado en PostgreSQL pódese escribir en múltiples linguaxes de programación. Para definir un procedemento en PgAdmin utilizamos a seguinte sintaxe:



Procedemento almacenado

```
CREATE [ OR REPLACE ] FUNCTION  
nome_funcion([ argumentos ])  
RETURNS tipo AS  
$BODY$  
    codigo  
$BODY$  
LANGUAGE linguaxe;
```

No código podemos escribir calquera instrución da linguaxe SQL para o manexo de base de datos. Os elementos entre corchetes son opcionais, non é necesario poñelos se non os imos a utilizar. O procedemento execútase cunha instrución SELECT:



Execución de procedemento almacenado

```
SELECT nomee_funcion();
```

3.5 Extracción de datos en sistemas de ERP-CRM e almacéns de datos

O proceso de extracción de datos podemos definilo como a operación de sacar datos dunha aplicación para ser tratados noutra aplicación. A extracción de datos pode realizarse utilizando diferentes sistemas.

Un uso moi común é utilizar ferramentas ofimáticas que se conectan á aplicación ERP, para obter información da base de datos e envorcala na aplicación ofimática como un procesador de textos, unha folla de cálculo etc.

Existen procesos máis complicados e potentes de extraer información. Son os chamados procesos de Business Intelligence. Este tipo de solucións deben realizar tres tarefas:

- transformar e combinar os datos para extraer a información,
- convertela en potentes indicadores e
- mostrala en distintos formatos gráficos.

Segundo a orixe dos datos e o tipo de información que queiramos obter, pódense utilizar:

- **Consultas e Informes.** Úsanse cando todos os datos están nunha soa base de datos, e extráense a partir dunha consulta SQL. A aplicación facilita informes e consultas predefinidos, aínda que como vimos tamén se poden xerar consultas e informes personalizados utilizando a propia aplicación, mesmo ferramentas externas como Jasper Reports.

- **Almacéns de datos.** A extracción de datos faise desde diferentes sistemas e distintas bases de datos, creando almacéns de datos co obxectivo de homoxeneizar e integrar a información.
- **Cubos multidimensionales.** Un cubo n-dimensional é un conxunto de datos multidimensionales organizados en eixos e celas, que manexa a información dunha base de datos relacional. Tamén existen bases de datos multidimensionales, como contraposición á operación de gardar os datos en bases de datos relacionales e logo manexalos con cubos.

En ocasións o proceso de extracción e manipulación da información non se realiza en tempo real debido ao gran volume de información que hai que manexar, para evitar unha diminución na velocidade de resposta á hora de presentar os datos. Isto quere dicir que primeiro se extrae a información e logo é manipulada, o cal significa que pode haber unha leve diferenza entre a información manipulada e o verdadeiro contido da base de datos.

3.5.1 Importar e exportar datos

Unha forma de extraer información da aplicación é exportando os datos. Tamén é posible introducir información de maneira masiva mediante a importación de datos. A importación e exportación de datos realízase a través dos mecanismos estándar que prové a aplicación.

O formato usual de importación e exportación de datos é o CSV , que é un formato de texto utilizado para representar datos en forma de táboa, nas que as columnas se separan por comas (ou punto e coma onde a coma é o separador decimal: España, Francia, Italia...) e as filas por saltos de liña.

A aplicación debe incorporar unha ferramenta de importación, que permita seleccionar os campos para importar e envorcar dita información no obxecto que desexemos. Pódense importar datos cara a unha soa táboa ou cara a varias táboas. No caso de varias táboas utilízase un separador para indicar a que táboa pertence cada un dos datos existentes no arquivo para importar.

O proceso de exportación é igual de sinxelo, desde o obxecto que queiramos exportar eliximos a opción de Exportar que aparecerá no formulario correspondente. O arquivo xerado poderá ser aberto en calquera aplicación ofimática ou no caso de CSV por un editor de textos sinxelo.

En Odoo, se exportamos desde o vista formulario, permitíranos máis campos a exportar xa que esta vista contén máis campos que a vista árbore.

3.6 Avaliación do rendemento e auditorías de acceso aos datos

Neste apartado imos ver como consultar e tratar a información a nivel de administración de sistemas. Cando somos os encargados de administrar un sistema, debemos ter

ferramentas que nos permitan facer un seguimento dos datos que arroxa o equipo servidor onde se atopan as aplicacións.

O rendemento do servidor pode diminuír ou ser inexistente debido a diversos motivos. Para investigar que ocorre en cada caso, é necesario buscar información nos ficheiros de rexistro ou mensaxes do sistema chamados logs, ou executar ferramentas que permitan realizar unha análise e monitorización do rendemento.


Podemos obter datos instantáneos do rendemento do sistema relativo ao funcionamento dos procesadores, da memoria, dos dispositivos de entrada e saída etc., pero tamén podemos recoller datos periodicamente e almacenalos en ficheiros históricos para consualos posteriormente. Estes datos proporcionannos información moi importante sobre as posibles carencias e pescozos de botella do noso sistema.

Existen diversas utilidades para recompilar e facer un histórico do rendemento e a actividade dos datos. Estas utilidades recompilan a información do sistema, almacénana por un período de tempo e calculan os valores medios. En calquera momento pódense tomar lecturas dos parámetros do servidor que se determinen, para a resolución de problemas ou ben simplemente para consultar o estado do noso servidor.

3.6.1 Monitorización e avaliación do rendemento


Entre as ferramentas de monitorización e avaliación do rendemento dispoñibles para un servidor Linux atópase a ferramenta **sar**, incluída no paquete **sysstat**.

A forma de instalar o **sar** é instalando o paquete **sysstat**, desde Synaptic ou ben executando desde o terminal a seguinte orde para Debian ou Ubuntu:

	Instalación de sar
<pre>\$ sudo apt-get install sysstat</pre>	

A partir de aí xa podemos ver datos instantáneos, tan só temos que ter en conta que todos os comandos da ferramenta aceptan como parámetros o número de valores que queremos obter (intervalo) e cada canto tempo queremos que os capture (segundos).

Por exemplo, para obter información de 3 valores sobre o uso do procesador, cada segundo, utilizamos a seguinte orde:

	Información de 3 valores sobre o uso do procesador
<pre>\$ sar 1 3</pre>	

E o resultado:



Resultado de ejecución de sar

Linux 2.6.35-28-generic (ubuntu)			21/12/11	_x86_64_	(1 CPU)		
16:38:02	CPU	%user	%nice	%system	%iowait	%steal	
%idle							
16:38:03	all	70,00	0,00	18,00	0,00	0,00	12,00
16:38:04	all	67,00	0,00	26,00	0,00	0,00	7,00
16:38:05	all	52,00	0,00	28,00	0,00	0,00	20,00
Media:	all	63,00	0,00	24,00	0,00	0,00	13,00

Os parámetros "1 3" indican que sar executarase cada segundo un total de 3 veces, os valores de execución móstranse en filas separadas, e a última fila é a media aritmética de todos os valores.

Para coñecer a lista completa dos parámetros que podemos utilizar con sar, podemos consultar a axuda do comando (man sar). Algúns exemplos son os seguintes:

- Procesador: **sar -P ALL**
- Memoria: **sar -r**
- Interfaces de rede: **sar -n DEV**
- Discos: **sar -d**



Gráfico da utilización do procesador

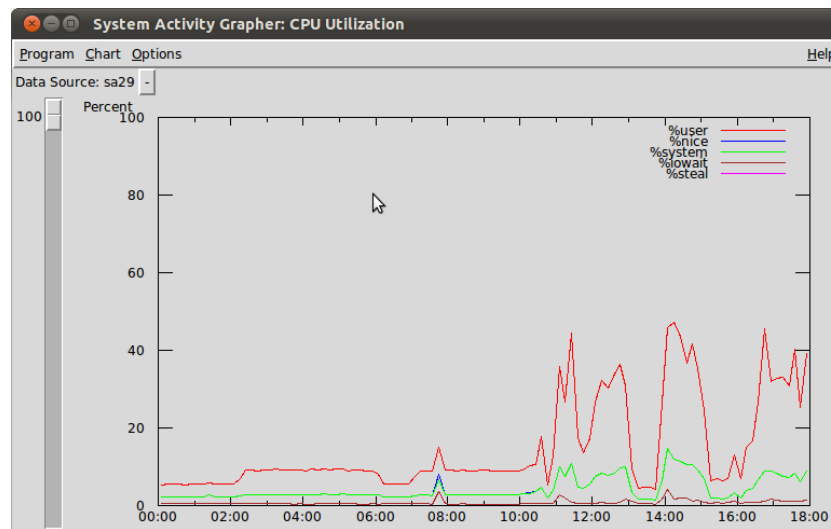


Figura 8: Utilización da CPU

Sistema operativo Ubuntu, que se distribúe baixo licenza GNU GPL. (GNU/GPL)


3.6.2 Auditorías de control de acceso aos datos. Trazas do sistema (logs)

A actividade dos programas, sobre todo se se trata de programas que se executan en servidores, queda rexistrada en ficheiros do sistema chamados logs. En ocasións podemos querer examinar os ficheiros de trazas do sistema para realizar un control de

acceso aos datos, xa que estes ficheiros de trazas van almacenando toda a actividade e eventos que ocorren no equipo: quen entra, que comandos executa, que erros mostran as aplicacións etc.

Na maioría das distribucións Linux, estes ficheiros gárdanse no directorio /var/log. Para visualizar o seu contido necesitamos permisos de root, ou pertencer a un grupo de usuarios con permisos para ver eses ficheiros. Neste directorio por exemplo gárdase o ficheiro syslog, que garda mensaxes de trazas de demos e outros programas como cron, init, dhclient, e algunhas mensaxes relacionadas co núcleo do sistema operativo. Se desexas consultar estes ficheiros de maneira gráfica na contorna gráfica de Gnome, podes ver como se fai nesta sinxela guía: Consultar os sucesos do sistema na interface gráfica de Ubuntu 18.04 LTS Desktop


Volvendo ao exemplo da ferramenta sar para monitorización de servidores Linux, para facer que sar capture información periódica debemos editar o ficheiro /etc/default/sysstat e a liña que empeza por ENABLED poñela co valor ENABLED="true". Unha vez feito, reiniciamos o servidor de sysstat coa orde:



Reiniciar servidor sysstat

```
$/etc/init.d/sysstat start
```

A partir de aí o script /usr/lib/sysstat/sa1, que se executa por defecto cada 10 minutos empezará a recoller datos de rendemento (procesador, memoria, disco, rede, etc) que se gardarán no ficheiro de sistema /var/log/sysstat/saXX, onde XX indica o día do mes no que nos atopamos. Por defecto, gárdanse os datos da última semana.



Modificación do ficheiro sysstat en linux

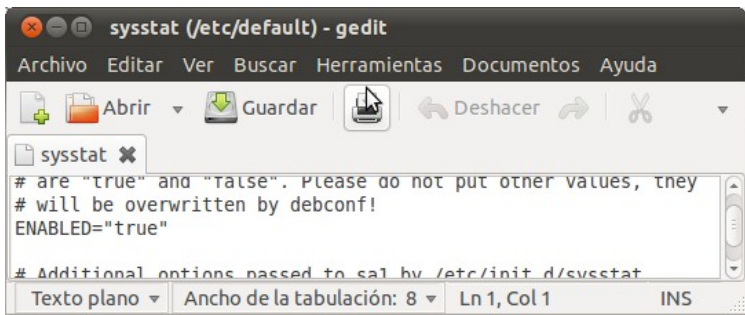


Figura 9: Modificación do ficheiro sysstat en linux


Terminal de Ubuntu. (GNU/GPL)

Para consultar de maneira gráfica as trazas do sistema de sar podemos utilizar a ferramenta Isag, dispoñible a través de Synaptic.

3.6.3 Incidencias: identificación e resolución

Ademais do control de acceso aos datos, o interesante do rexistro de trazas do sistema é consultar a información posteriormente para resolver posibles problemas, xa que xeralmente se unha aplicación non funciona, ou non pode inicializarse, o que fai é imprimir unha traza de erro, que pode poñer sobre aviso do que está a ocorrer.


Dentro do directorio `/var/log` podemos atopar as trazas do sistema dunha aplicación en Ubuntu. Por exemplo, podemos querer visualizar a actividade do servidor de Odoo co seguinte comando:



Actividade do servidor Odoo

```
$ sudo head /var/log/odoo/odoo-server.log
```

O resultado da instrución serán as dez primeiras liñas do ficheiro de log:



Ficheiro de rexistros odoo-server.log

```
root@serverodoo:~# head /var/log/odoo/odoo-server.log
2020-07-30 10:21:16,978 23859 INFO ? odoo: Odoo version 12.0-20200730
2020-07-30 10:21:16,978 23859 INFO ? odoo: Using configuration file at /etc/odoo/odoo.conf
2020-07-30 10:21:16,978 23859 INFO ? odoo: addons paths: ['/var/lib/odoo/.local/share/Odoo/addons/12.0', '/usr/lib/python3/dist-packages/odoo/addons']
2020-07-30 10:21:16,978 23859 INFO ? odoo: database: odoo@default:default
2020-07-30 10:21:17,083 23859 INFO ? odoo.addons.base.models.ir_actions_report: You need Wkhtmltopdf to print a pdf version of the reports.
2020-07-30 10:21:17,173 23859 WARNING ? odoo.addons.base.models.res_currency: The num2words python library is not installed, amount-to-text features won't be fully available.
2020-07-30 10:21:17,209 23859 INFO ? odoo.service.server: HTTP service (werkzeug) running on server odoo:8069
2020-07-30 10:23:51,062 23859 INFO ? odoo.http: HTTP Configuring static files
2020-07-30 10:23:51,076 23859 INFO ? odoo.http: Generating nondb routing
2020-07-30 10:23:51,088 23859 INFO ? werkzeug: 192.168.1.52 -- [30/Jul/2020 10:23:51] "GET / HTTP/1.1" 200 - 1 0.003 0.020
root@serverodoo:~#
```

Figura 10: Ficheiro de rexistros de Odoo

Terminal de Ubuntu. (GNU/GPL)

Onde podemos interpretar que o servidor de Odoo atópase executándose e esperando conexións no porto 8069. Calquera incidencia quedará reflectida nestes arquivos, e será cuestión nosa identificar a causa e unha posible solución.