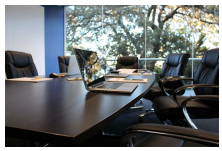
Organización, consulta y tratamiento de la información.

Caso práctico

El equipo de **BK Programación** está trabajando a fondo en el estudio de las aplicaciones de planificación empresarial. Creen que puede ser una nueva vía de negocio el ofrecer servicios de consultoría para implantar este tipo de aplicaciones. Para ello necesitan seguir estudiando sus características. **María** y **Juan** intentan sacar tiempo para dedicarle al proyecto, así que han decidido dejar instalada la aplicación en un equipo para poder ir instalando los demás programas conforme los vayan necesitando.



pigsels (CC0)

—Podemos utilizar este equipo que tiene el sistema operativo recién instalado —comenta **María**.

—De acuerdo —responde **Juan**-. Después tendremos que instalar un programa para administrar las bases de datos de la aplicación —señala **Juan**.

—Sí —responde **María**—, poder acceder directamente a las bases de datos nos ayudará a hacernos una idea más completa de cómo funciona la aplicación por dentro.

—Exacto —dice **Juan**— ¿cuál utilizamos?

Mientras instalan la aplicación, **Juan** empieza a buscar una herramienta de código abierto para acceso a bases de datos PostgreSQL. Los principales objetivos son:

- Que facilite la administración de la base de datos.
- Que se puedan hacer la mayor parte de las tareas de una forma gráfica e intuitiva.

1.- Organización y consulta de la información.

Pigsels (CC0)

Hasta ahora, hemos visto las funcionalidades básicas de dos ERPs muy extendidos en el mercado. En adelante nos centraremos principalmente en una de estas dos aplicaciones para poder explorar más a fondo todas las posibilidades de un ERP. Básicamente por poseer un módulo base un tanto más sencillo formado tan sólo por **Empresas** y **Administración**, nos centraremos en Odoo .

La base de datos de un sistema ERP es de gran envergadura. Almacena las tablas con los datos de la aplicación, vistas de las diferentes tablas y otros elementos como funciones o disparadores que realizan operaciones sobre los datos. Por ello, debido a esta gran cantidad de información almacenada, se hace necesario una organización entre sus componentes.

Lo que se hace es establecer una serie de normativas o nomenclatura para organizar la información, que los desarrolladores deben seguir a la hora de modificar el código fuente o el esquema de la base de datos. Por ejemplo, incluir un prefijo en los componentes de la base de datos, para saber a qué módulo pertenecen, o establecer una serie de campos dentro de una tabla como obligatorios, para poder asegurar el funcionamiento correcto de la aplicación.

En los sistemas de planificación empresarial desarrollados en un lenguaje orientado a objetos, cualquier dato es accesible a través de objetos. Por ejemplo, en Odoo tenemos un objeto **res.partner** para acceder a los datos concernientes a los colaboradores o socios, un objeto **account.invoice** para los datos de las facturas, etc. Como ves, ambos van precedidos de un prefijo que indica el módulo al cual pertenecen.

Cualquier método que quiera actuar sobre un objeto deberá tener un parámetro que indique sobre qué recurso o registro dentro del objeto se quiere actuar. Por ejemplo, si queremos enviar un correo electrónico a los colaboradores identificados como 1 y 5 dentro de la tabla **res.partner**, utilizaremos la siguiente instrucción:

```
res.partner.send_email(..., [1,5], ...)
```

En resumen, un objeto es, en un concepto amplio, todo elemento que forma parte de la aplicación y que permite acceder a los datos de la misma.

Para saber más

En este apartado vamos a encontrarnos con cierta terminología de base de datos al acceder a los datos del ERP. Además del enlace del Portal de español de PostgreSQL proporcionado en unidades anteriores, en el siguiente enlace se ofrecen una serie de tutoriales sobre PostgreSQL. Puede ser interesante consultar alguno de ellos para tener una idea de cómo funciona este gestor de base de datos o para refrescar conocimientos:

Tutoriales PostgreSQL.

Puedes repasar conceptos sobre el lenguaje de consulta estructurado SQL en el siguiente enlace:

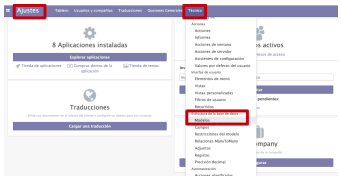
Tutorial del lenguaje SQL.

1.1.- Tablas y vistas de la base de datos.

Como hemos explicado, **un objeto guarda todo elemento que forma parte de la aplicación**. Como elementos de la aplicación tenemos la propia base de datos (tablas, disparadores o acciones sobre esas tablas, etc), los formularios, los informes, las ventanas o los asistentes, por ejemplo.

Ross (CC BY-SA)

En Odoo se accede a los objetos a través de la opción **Ajustes-> Técnico-> Estructura de la base de datos-> Modelos**. En esta opción podemos encontrar toda la información que maneja la aplicación.



Captura de pantalla de Odoo propiedad de Odoo, S.A. (Licencia LGPLv3)

En **Openbravo** todos los objetos con los que trabaja la aplicación, llamados **metadatos**, están disponibles en el módulo **Diccionario de Datos**, cuya función es administrar estos metadatos de una manera centralizada. En este módulo se dan de alta los formularios, informes y procesos que se utilizan en la aplicación, a los que se accederán mediante menús o botones en ventanas, así como las tablas y campos (o columnas) que las forman.

Recuerda que para entrar en el Odoo debemos utilizar la dirección IP del servidor seguido del puerto de comunicación, es decir, si estamos trabajando en local (`localhost:8069`), es el puerto por defecto. Si trabajamos en remoto (`xxx.xxx.xxx.xxx:8069`) siendo las X's la dirección IP dl ordenador donde está instalado el Odoo, lógicamente también se puede escribir el nombre del ordenador seguido del puerto.

Objetos principales en Odoo

- Modelos: Las tablas con las que trabaja la aplicación
- Vistas: Los distintos tipos de formularios con los que muestra los datos de los modelos.
- Menús: La estructura de menús con las que llamaremos a las acciones para manejar los datos de los modelos de la aplicación.
- Acciones: Los métodos desde los que vamos a abrir nuestras vistas para trabajar en la aplicación.

Dentro de los objetos distinguimos las tablas. **Una tabla es una estructura de datos organizada en filas y columnas, de manera que cada columna es un campo (o atributo) y cada fila un registro**. Por ejemplo, en el modelo `res.users` cada registro se corresponde con un usuario, y para cada usuario se guardan una serie de atributos como `nombre`, `login` o `password`.

En ocasiones, la base de datos está formada por tantas tablas y objetos que se vuelve compleja y difícil de manejar. En esos casos, interesa que algunos grupos o perfiles de usuarios tengan una vista parcial de esos datos. Para estos casos se utilizan las vistas. **Una vista es básicamente una "tabla virtual" a la que se puede acceder como si fuera una tabla del esquema, pero que realmente no lo es**. Tienen la misma estructura que las tablas: filas y columnas o campos, y se puede acceder a ellas de la misma forma, a través de consultas de acceso a datos como veremos posteriormente.

Para administrar la base de datos podemos conectarnos directamente al gestor de la base de datos, aunque lo normal es utilizar herramientas gráficas que nos faciliten el trabajo. Entre las herramientas para administración de bases de datos PostgreSQL se encuentra **PGAdmin 4**.

Debes conocer

En el siguiente enlace puedes consultar cómo se instala la herramienta gráfica PgAdmin 4 para administrar bases de datos PostgreSQL:

Instalar y configurar PgAdmin 4.

Una vez instalado PgAdmin 4, nos conectaremos a la base de datos introduciendo los siguientes datos:

- **Nombre**: nombre que queramos darle a la conexión.
- **Servidor**: Dirección IP o nombre del servidor.
- **Puerto**: Número de puerto para la conexión, normalmente el `5432`.

- **Base de datos de Mantenimiento**: base de datos inicial con la que nos conectamos, normalmente llamada "`postgres`".
- **Nombre de usuario**: usuario con el que queramos conectarnos, en nuestro caso, debemos introducir el usuario que creamos en la segunda unidad, concretamente, el usuario "`userbd`".
- **Contraseña**: clave del usuario.

Autoevaluación

Las vistas no se sustentan en datos almacenados físicamente, sino que están construidas en base a otras tablas.

○ Verdadero ○ Falso

Verdadero
Efectivamente, las vistas no guardan datos en sí mismas, sino que se basan en los datos que contienen las tablas.

1.2.- Consultas de acceso a datos.

Las consultas de acceso a datos nos permiten acceder a la información que guardan las tablas y vistas de la base de datos. Las consultas de acceso a datos sirven para indicar al sistema de gestión de la base de datos que devuelva un extracto de la información en forma de un conjunto de registros.

Los pasos para crear una consulta son:

- Seleccionar las tablas o vistas sobres las que va a actuar la consulta.
- Establecer la relación entre las tablas y vistas, en caso de que la aplicación no la proporcione.
- Seleccionar los campos a mostrar en la consulta.
- Ejecutar la consulta.

Las consultas pueden actuar sobre una o varias tablas o vistas, y se pueden guardar para ser utilizadas posteriormente. En ocasiones la aplicación permite realizar consultas de acceso a datos, o bien podemos conectarnos directamente al sistema gestor de base de datos.

Las consultas de acceso a datos se pueden construir escribiendo el código en el lenguaje de consulta utilizado, como por ejemplo SQL, o bien mediante asistentes y constructores gráficos si se trata de consultas poco complejas.

Debes conocer

El siguiente enlace, muestra cómo visualizar gráficamente una consulta de acceso a datos con PgAdmin 4, ahora bien no se puede crear con PgAdmin 4 una consulta gráficamente como sí era posible en su versión anterior pgAdmin III. Puedes ver en esta página web cómo se hace:

Visualizar gráficamente una consulta de acceso a datos en pgAdmin 4.

2.- Visualización de la información.

Caso práctico

María está reunida con **Juan** para hablar sobre sus avances en el estudio de ERP. Tras instalar el programa para acceder a la base de datos, ha podido consultar cómo están formadas las tablas y demás objetos que maneja la aplicación. Esto es un punto de partida para cualquier adaptación que quieran hacer a la aplicación.

Pigsels (CC0)

Ahora están en disposición de crear los formularios que les interesen, informes o gráficos.

—Los objetos se describen utilizando un lenguaje de marcas estándar conocido como XML ¿qué tal andas en este lenguaje? —le pregunta **María**.

—Lo he utilizado para compartir información entre aplicaciones y servidores web —responde **Juan**—. La verdad es que es un lenguaje que no requiere de muchos conocimientos previos.

—Sí —comenta **María**—, y aunque no sea un lenguaje de programación en sí mismo, sino un meta-lenguaje que permite describir secuencias de datos, practicando es como mejor lograremos familiarizarnos con él y con las tecnologías a él asociadas.

Dado que todos los datos del programa están almacenados en objetos, ¿cómo se muestran dichos objetos al usuario? **Tanto la información de las tablas como la de cualquier otro objeto de la aplicación se muestra a través de interfaces**.

dullhunk (CC BY)

Cada objeto tiene su propia interfaz, por ejemplo, no se muestran de la misma manera los datos de las Empresas que los datos de una Factura. Las interfaces pueden ser:

- **Estáticas**: se crean dentro del código de la aplicación y no pueden ser modificadas.
- **Dinámicas**: pueden ser modificadas por parte del usuario, para lo cual se almacena la descripción de la vista en un lenguaje de descripción de datos que permita su modificación, como por ejemplo **XML**.

Por tanto, las interfaces dinámicas son construidas de forma dinámica por la descripción XML de la pantalla del cliente. Para ello no es necesario ser unos expertos en ese lenguaje, podemos hacer objetos sencillos simplemente tomando como ejemplo otros objetos que haya creados en la aplicación. No obstante, muchas aplicaciones proveen la forma de crear las descripciones de manera gráfica sin necesidad de introducir código manual.

El siguiente código es un ejemplo de descripción de un objeto 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>
```

Con el código anterior estamos creando una interface para la introducción y consulta de datos. En concreto, se muestran seis campos correspondientes al nombre, título o tratamiento en caso de ser una persona física, código, idioma, y por último dos campos booleanos que guardan si se trata de un cliente o de un proveedor. Esta definición se almacenará en un archivo XML que, al abrirlo desde la aplicación, mostrará el objeto resultante de la definición anterior:

Captura de pantalla de Odoo propiedad de Odoo, S.A. (Licencia LGPLv3)

2.1.- Interfaces de entrada de datos y de procesos. Formularios y Gráficos.

Podemos definir una o varias interfaces para cada objeto o tipo de recurso, para describir qué campos se muestran y cómo. Dependiendo de cómo se distribuyan los campos nos encontramos, entre otras, con los siguientes tipos de interfaces:

Captura de pantalla de Odoo propiedad de Odoo, S.A. (Licencia LGPLv3)

- **Formularios**: en este tipo de interfaces se muestra un sólo registro, cuyos campos se distribuyen en la pantalla siguiendo siempre el mismo criterio, normalmente de izquierda a derecha y de arriba a abajo, de acuerdo con el orden en el que son descritos en la vista.
- **Árboles**: este tipo de interfaces se utiliza cuando queremos mostrar un conjunto de registros en modo lista, y es útil para mostrar varios registros a la vez y realizar búsquedas sobre ellos.
- **Gráficos**: los gráficos son otra forma de ver los datos de un formulario, y pueden mostrarse en varios formatos y tipos para una mejor visualización de la información.

Odoo llama a las interfaces **Vistas**, y desde la aplicación las podemos personalizar de dos maneras diferentes:

- Escribiendo código XML. Editamos la vista y escribimos directamente el código. Esto lo hacemos a desde el menú **Ajustes/Técnico/Interfaz de usuario/Vistas**. (Recuerda que tienes que ser superusuario)
- Accediendo al módulo y modificando el fichero XML correspondiente. La ruta donde se encuentran las aplicaciones es diferente de linux a windows, en linux la ruta es */usr/lib/python3/dist-packages/odoo/addons* en windows se encuentra en */archivos de programa/odoo/addons*

Una vez modificada la vista, es recomendable actualizar la aplicación desde el menú **Aplicaciones**, buscas tu aplicación y en los tres puntos de su parte superior derecha pulsas y seleccionas Actualizar.

Para saber más

En el siguiente enlace puedes consultar información sobre los distintos elementos que se pueden utilizar dentro de la definición de una vista:

Vistas en Odoo.

Comprueba la versión, puede cambiar.

Autoevaluación

En las vistas de tipo formulario los datos se distribuyen en formato de lista, y se van situando los registros uno debajo del otro y así sucesivamente.

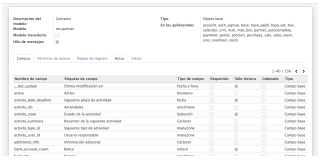
○ Verdadero ○ Falso

Falso
Las vistas de tipo formulario sólo muestran un registro a la vez.

2.2.- Definición de campos.

Al principio del apartado veíamos un ejemplo de definición de un objeto. Decíamos que los objetos están definidos en archivos XML. Un objeto está formado por campos. Los campos son la información que se muestra en pantalla. Por ejemplo, una vista de tipo formulario con dos campos nombre y título la definiríamos así:

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



Captura de pantalla de Odoo propiedad de Odoo, S.A.
(Licencia LGPLv3)

En el ejemplo anterior, hemos utilizado distintos tipos de etiquetas para describir la vista, La etiqueta **form** indica que vamos a describir un formulario y dentro de ella se utiliza la etiqueta **field** para cada uno de los campos que lo componen. Las etiquetas empiezan con el símbolo < y terminan con el símbolo />. En las vistas de tipo formulario, los campos van precedidos por una etiqueta con su nombre y se colocan de izquierda a derecha, en el orden con que son declarados en el archivo XML .

Hay un tipo especial de campo que son los campos de relación **one2many**. Estos campos se utilizan para reflejar una relación de uno-a-muchos entre dos objetos. Por ejemplo, el campo **address** dentro del objeto **res.partner** es un campo **one2many** lo cual quiere decir que ese campo está enlazado con otro objeto, en este caso **res.partner.address**, esto es, que un registro de **res.partner** puede tener muchos registros relacionados en **res.partner.address**. En otras palabras, una empresa puede tener muchas direcciones también llamadas contactos de la empresa, y esas direcciones se guardan en la tabla **res.partner.address**. A la hora de describir este tipo de campos en el archivo XML utilizaremos la sintaxis:

```
<field mode=”form,tree” name=”address” >
    <dentro pondremos los campos que forman parte de res.partner.address>
</field>
```

En el menú **Ajustes/Técnico/Estructura de la base de datos/Modelos** podemos ver los campos del objeto Empresa(res.partner), y veremos que uno de ellos es el campo address de tipo One2many.

Además de los campos, existen varios elementos de diseño que nos permiten personalizar las vistas form y tree. Algunos de ellos son:

- **mode**: tipo de vistas que va a permitir el objeto.
- **name**: nombre del campo tal y como aparece en el objeto.
- **separator**: agrega una línea de separación en el formato, ejemplo:

  <separator string="link" colspan="4"/>

  El atributo **string** define la etiqueta del separador y el atributo colspan define su tamaño .

- **notebook**: Permite distribuir los campos de la vista en diferentes pestañas o tabs que van definidos por paginas, ejemplo:

  <notebook colspan="4">....</notebook>

Por otra parte, a la etiqueta **field** se le pueden añadir atributos. Algunos de ellos son:

- **colspan="4"**:El numero de columnas por las que se puede extender un campo .
- **readonly="1"**: Establece un campo como solo lectura .
- **invisible="True"**: Oculta el campo y su etiqueta .
- **password="True"**: Reemplaza la entrada de un campo con un símbolo **"•"** .
- **string=""**: La etiqueta que va a aparecer junto al campo.

Autoevaluación

Señala cuáles son elementos de diseño para la creación de vistas.

- [ ] **mode.**
- [ ] **readonly.**
- [ ] **separator.**
- [ ] **res.partner.**

Mostrar retroalimentación

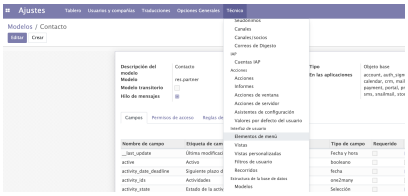
Solución

1. Correcto
2. Incorrecto
3. Correcto
4. Incorrecto

2.3.- Menús.

Para acceder a los objetos que creamos en la aplicación, deben definirse previamente los menús así como las acciones asociadas a los mismos.

Los menús se pueden editar desde **Ajustes/Técnico/Interface** de **usuario/Elementos de Menú**. Desde ahí podemos modificar el menú seleccionado o crear nuevos menús. Para crear menú secundario seguiremos el mismo proceso, tan sólo tendremos que indicar cuál es el menú padre del cual depende.



Captura de pantalla de Odoo propiedad de Odoo, S.A. (Licencia LGPLv3)

Los menús tienen asociada la acción que deben realizar. Hay diferentes tipos de acciones, las más importantes:

- ir.actions.act_window: Abre una vista en una nueva ventana.
- ir.actions.report: Imprime un informe.

Por tanto, cuando creamos un menú, debemos indicar el tipo de acción que tiene asociada. Si la acción no está creada debemos utilizar el enlace **Crear**... para crearla. Los campos a introducir variarán según la acción de que se trate. Por ejemplo, para crear una nueva acción que abra una vista debemos introducir al menos los siguientes datos:

- **Nombre de la acción**.
- **Tipo de la acción**, en este caso, **ir.actions.acc_window.**
- **Datos de la vista**: tipo (árbol, formulario, etc.) y nombre.

Citas para pensar

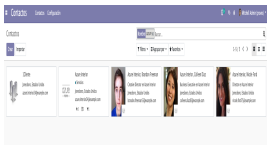
Nadie es consciente de su capacidad, hasta que la pone a prueba.

Thomas Alva Edison.

2.4.- Búsqueda de información.

Un aspecto muy importante de una aplicación ERP es cómo lleva a cabo la búsqueda de información. Los motores de búsqueda deben ser lo suficientemente potentes para que el proceso sea rápido y efectivo.

Normalmente se incluyen búsquedas básicas sobre los campos principales, y búsquedas avanzadas para buscar registros por campos más específicos.



Captura de pantalla de Odoo propiedad de Odoo, S.A. (Licencia LGPLv3)

La opción de búsqueda está disponible en cada ventana de la aplicación. En el caso de Odoo, los criterios de búsqueda se introducen en la parte superior de la ventana y debajo se sitúa una vista de tipo árbol con el listado de los registros del objeto.

El listado se reducirá a los registros coincidentes con los criterios de búsqueda. También es posible aplicar filtros a la selección. Los registros pueden ser seleccionados y cambiando al modo formulario podremos ver todos sus datos, así como editar su contenido.

Debes conocer

En el siguiente vídeo se muestra una explicación sobre los menús personalizados de Odoo:

http://www.youtube.com/embed/OB5hwe1o6cY

2.5.- Informes y listados de la aplicación.

Los informes y listados de la aplicación son una forma de mostrar los datos para mejorar su visualización. Muchos informes y listados están creados, y son accesibles desde los distintos menús de la aplicación.

Pxfuel (CC0)

Algunos de estos documentos pueden ser:

- Informes contables.
- Albaranes.
- Pedidos.
- Recibos.
- Reclamaciones a proveedores y/o clientes.
- Informes estadísticos.
- Generación de etiquetas para un conjunto de registros.
- Informes de agrupamiento: permiten mostrar los registros existentes para un mismo valor de un campo.
- Impresión de documentos del sistema.

Los informes y listados que incorpora la aplicación se pueden utilizar tal cual se proporcionan, o bien adaptar su diseño a la imagen corporativa de la empresa. También se pueden **añadir nuevos informes** instalándolos como módulos independientes. En el caso de Odoo, los módulos extra que contienen exclusivamente informes de algún tipo llevan en su nombre la etiqueta `"report"`.

Autoevaluación

Las aplicaciones ERP incorporan de forma predefinida la mayor parte de los informes que necesita cualquier empresa.

○ Verdadero ○ Falso

Falso
Algunos informes se encuentran de manera predefinida cuando se instala la aplicación, pero la mayor parte de ellos se realizan con posterioridad a la instalación, en la etapa de implantación.

3.- Tratamiento de la información.

Caso práctico

María y **Juan** son los tutores laborales de **Ana**, una becaria en prácticas del ciclo de **Desarrollo de Aplicaciones Multiplataforma**. Ellos tienen que salir de viaje y le han pedido que investigue en la aplicación dónde se introducen los datos iniciales para poder empezar a trabajar con la aplicación.

—¿Qué te parece si cuando volvamos **Juan** y yo nos cuentas lo que hayas averiguado? —pregunta **María**.

—¡Estupendo! —exclama **Ana**.

—De acuerdo —comenta **María**—, además podrías mirar en la base de datos los procedimientos almacenados de servidor, puede que después los necesitemos.

María está convencida de que **Ana** podrá averiguar muchas cosas, tiene mucha iniciativa y ganas de aprender, que es lo fundamental.

Elaboración propia (Uso educativo no comercial)

El tratamiento de la información se lleva a cabo a través de distintos procesos. Antes de poder realizar ningún proceso es necesario introducir la información propia de la compañía. Por ejemplo, si vamos a llevar la contabilidad de la empresa habrán de configurarse las cuentas contables, se tendrán que definir las cuentas bancarias y los diarios de compras, ventas y caja o banco. También habrán de definirse los impuestos.

Stephen Craven (CC BY-SA)

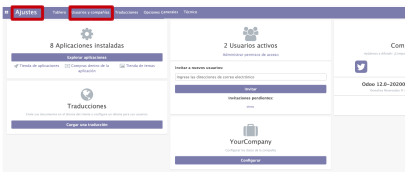
Por ejemplo, si tenemos instalado el módulo de contabilidad, cuando creemos una empresa cliente o proveedor, deberemos asociarle una serie de cuentas contables, como las de cobro al cliente y pago al proveedor, por lo que deberemos tener creadas dichas cuentas antes de poder dar de alta a nuestros clientes y proveedores, e igual con los productos.

Dentro de la información propia de la compañía deberemos crear los clientes, proveedores y productos. En cualquiera de los casos, existe una información mínima que tendremos que introducir antes de poder empezar a trabajar.

Al instalar el perfil mínimo de Odoo deberemos comprobar si existen datos, y si no, introducirlos, de al menos los siguientes objetos o tablas:

- **Título**: Hace referencia al tipo de empresa o al tratamiento que se le debe otorgar a una persona de contacto. Si es uno u otro se distingue a través del campo `Dominio` que hay en la tabla, que será igual a `"Empresa"` en el primero de los casos o a `"Contacto"` en el segundo caso.
- **Función**: Listado de cargos o funciones de las personas de contacto.
- **Provincias**: Listado de provincias por país.

Esta información es necesaria para poder introducir los datos de las empresas, y podemos acceder a ella desde el menú **Ajustes/Usuarios y Compañías/Compañías**.



Captura de pantalla de Odoo propiedad de Odoo, S.A. (Licencia LGPLv3)

3.1.- Cálculos: pedidos, albaranes, facturas, asientos predefinidos, trazabilidad y producción.

Entre los procesos a realizar por un sistema de planificación empresarial destacan los siguientes:

Peter Gerken (CC BY-SA)

- Contabilidad. Incluye los procesos donde se reflejan las operaciones económicas realizadas por la empresa, la determinación de los costes de la empresa y los presupuestos del ejercicio fiscal. Se proporcionan asientos predefinidos para la introducción rápida de asientos sin necesidad de tener conocimientos de contabilidad.
- Operaciones de compra:
  - Crear una orden de compra o pedido de compra.
  - Recibir los bienes.
  - Controlar la factura de compra.
  - Registrar el pago al proveedor.
- Operaciones de venta:
  - Crear una orden de venta o pedido de venta y recibir la conformidad del cliente.
  - Preparar los bienes a enviar al cliente y realizar el albarán y la entrega.
  - Realizar la factura de venta.
  - Registrar el cobro al cliente o pago del cliente.
- Trazabilidad: se llama así al proceso de la entrada del producto hasta la salida del mismo.

Es posible que por las necesidades de la empresa no sea necesario utilizar todos los procesos del ERP, por ejemplo, puede ser usado sólo como un CRM, o sólo como un programa de contabilidad, pero su verdadera potencia se alcanza con la integración de todas sus funciones.

Autoevaluación

Los procesos del sistema son en realidad las operaciones que lleva a cabo la aplicación ERP.

○ Verdadero ○ Falso

Verdadero

Efectivamente, y tratan de cubrir todas las áreas de la empresa.

3.2.- Procedimientos almacenados de servidor.

pxfuel (CC0)

Dentro del tratamiento que le damos a la información, a veces es necesario que se lleven a cabo determinadas tareas de forma automática, en respuesta a algo que pasa en la aplicación. Por ejemplo, podemos querer enviar un correo electrónico al cliente de manera automática, cada vez que nos hace un pedido, o bien registrar en la ficha del cliente el pedido realizado. Ambas acciones: enviar el correo o registrar el pedido en el histórico del cliente, son acciones que se desencadenan automáticamente por el sistema, y la acción que las desencadena es la realización de un pedido por parte de un cliente.

Para hacer esto en la mayoría de los sistemas podemos utilizar:

- **Procedimientos almacenados de servidor**. Un procedimiento almacenado es un programa, procedimiento o función almacenado en una base de datos y listo para ser usado. Pueden ser ejecutados directamente por el usuario o bien cuando se cumpla una determinada condición a través de disparadores.
- **Eventos de servidor**. Un evento de servidor consiste en detectar que pasa algo en la aplicación y hacer que el sistema responda a este suceso de forma automática. Los eventos de servidor son creados a nivel de objetos y no de base de datos, como ocurre en los procedimientos. Podemos definir los eventos de servidor a través de los menús de la aplicación, no siendo necesario introducirnos en la base de datos para programar la acción deseada.

Un procedimiento almacenado en PostgreSQL se puede escribir en múltiples lenguajes de programación. Para definir un procedimiento en PgAdmin 4 utilizamos la siguiente sintaxis:

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

En el código podemos escribir cualquier instrucción del lenguaje SQL para el manejo de base de datos. Los elementos entre corchetes son opcionales, no es necesario ponerlos si no los vamos a utilizar. El procedimiento se ejecuta con una instrucción SELECT:

```
SELECT nombre_funcion();
```

Ejercicio resuelto

Crear un procedimiento almacenado en Odoo que muestre las empresas que sean sociedades limitadas.

Mostrar retroalimentación

Primero creamos el procedimiento almacenado, escribiendo el siguiente código en el botón de "Ejecutar consultas SQL arbitrarias" de pgAdmin 4:

```
CREATE OR REPLACE FUNCTION PRUEBA()
  RETURNS setof res_partner AS
$BODY$
select * from res_partner where title='Ltd'
$BODY$
LANGUAGE sql;
```

Para crear el procedimiento hemos utilizado una instrucción SELECT que selecciona los registros de la tabla res_partner cuyo campo title coincida con el código "Ltd", que es el utilizado internamente para referirse a las empresas de tipo Sociedad Limitada.

Le damos al botón de **"Ejecutar consulta"** y para comprobar que se ha creado la función, salimos del editor de consultas y le damos a refrescar del menú contextual de **Funciones()**.

Volvemos al editor de consultas dándole al botón "**Ejecutar consultas SQL arbitrarias**" y utilizamos la instrucción SELECT para ejecutar la función:

```
SELECT prueba();
```

Si todo ha ido bien, el resultado de ejecutar el procedimiento almacenado debe ser el listado de todas las sociedades limitadas. Cada registro aparecerá en una fila de la tabla, con los campos encerrados entre paréntesis y separados unos de otros por comas, en nuestro caso algo así como:

```
"(4,1,"2012-01-14 16:23:10.856227","2012-01-21 18:17:55.646055",1,,,t,,es_ES,t,3,,"Comercial Riabu",ltd,,,f,,)"
```

"(8,1,"2012-01-21 18:30:18.675727","2012-01-21 18:51:24.376119",1,,,t,,es_ES,t,2,,"Olivencia Consultores",ltd,,,f,,)"

Autoevaluación

Ejercicio de relacionar

Término	Relación	Definición
Evento de servidor.	☐	1. Programa almacenado en la base de datos.
Procedimiento almacenado.	☐	2. Acción definida en la aplicación.

Enviar

Ambos términos son similares en cuanto a la acción que realizan, pero no en cómo la realizan.

4.- Extracción de datos en sistemas de ERP-CRM y almacenes de datos.

Caso práctico

Pigsels (CC0)

María y **Juan** siguen de viaje y **Ana** ya ha realizado las tareas encomendadas. En lugar de perder el tiempo, quiere aprovecharlo investigando más sobre la aplicación. Ahora está leyendo en Internet cómo extraer datos para utilizarlos en otras aplicaciones. Recuerda que en el ciclo estudiaron algunas tecnologías de extracción de datos que se habían venido utilizando en aplicaciones que no permitían cambios, y una forma de adaptarse a las necesidades de los clientes era extrayendo los datos y creando informes en otras aplicaciones.

El proceso de extracción de datos podemos definirlo como la operación de sacar datos de una aplicación para ser tratados en otra aplicación. La extracción de datos puede realizarse utilizando diferentes sistemas.

Un uso muy común es utilizar herramientas ofimáticas que se conectan a la aplicación ERP, para obtener información de la base de datos y volcarla en la aplicación ofimática como un procesador de textos, una hoja de cálculo, etc.

Existen procesos más complicados y potentes de extraer información. Son los llamados procesos de Business Intelligence. Este tipo de soluciones deben realizar tres tareas:

- transformar y combinar los datos para extraer la información,
- convertirla en potentes indicadores y
- mostrarla en distintos formatos gráficos.

Según el origen de los datos y el tipo de información que queramos obtener, se pueden utilizar:

- **Consultas e Informes**. Se usan cuando todos los datos están en una sola base de datos, y se extraen a partir de una consulta SQL. La aplicación facilita informes y consultas predefinidos, aunque como hemos visto también se pueden generar consultas e informes personalizados utilizando la propia aplicación, incluso herramientas externas como Jasper Reports.
- **Almacenes de datos**. La extracción de datos se hace desde diferentes sistemas y distintas bases de datos, creando almacenes de datos con el objetivo de homogeneizar e integrar la información.
- **Cubos multidimensionales**. Un cubo n-dimensional es un conjunto de datos multidimensionales organizados en ejes y celdas, que maneja la información de una base de datos relacional. También existen bases de datos multidimensionales, como contraposición a la operación de guardar los datos en bases de datos relacionales y luego manejarlos con cubos.

En ocasiones el proceso de extracción y manipulación de la información no se realiza en tiempo real debido al gran volumen de información que hay que manejar, para evitar una disminución en la velocidad de respuesta a la hora de presentar los datos. Esto quiere decir que primero se extrae la información y luego es manipulada, lo cual significa que puede haber una leve diferencia entre la información manipulada y el verdadero contenido de la base de datos.

4.1.- Importar y exportar datos.

Michael Mandiberg (CC BY-SA)

Una forma de extraer información de la aplicación es exportando los datos. También es posible introducir información de manera masiva mediante la importación de datos. La importación y exportación de datos se realiza a través de los mecanismos estándar que provee la aplicación.

El formato usual de importación y exportación de datos es el CSV , que es un formato de texto utilizado para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma en donde la coma es el separador decimal: España, Francia, Italia...) y las filas por saltos de línea.

La aplicación debe incorporar una herramienta de importación, que permita seleccionar los campos a importar y volcar dicha información en el objeto que deseemos. Se pueden importar datos hacia una sola tabla o hacia varias tablas. En el caso de varias tablas se utiliza un separador para indicar a qué tabla pertenece cada uno de los datos existentes en el archivo a importar.

El proceso de exportación es igual de sencillo, desde el objeto que queramos exportar elegimos la opción de **Exportar** que aparecerá en el formulario correspondiente. El archivo generado podrá ser abierto en cualquier aplicación ofimática o en el caso de CSV por un editor de textos sencillo.

En Odoo, si exportamos desde la vista formulario, nos permitirá más campos a exportar ya que esta vista contiene más campos que la vista árbol.

Debes conocer

Consulta el siguiente enlace para conocer cómo se pueden importar datos a Odoo:

Importación de datos a Odoo.

5.- Evaluación del rendimiento y auditorías de acceso a los datos.

Caso práctico

María tiene que realizar unos informes de cómo funciona el servidor con el ERP instalado. Para ello es necesario dejar ejecutándose las herramientas de monitorización durante unos días, y posteriormente consultar los información capturada para poder sacar conclusiones. La información la consultarán con alguna herramienta que permita mostrar gráficos que ayuden a interpretar mejor los datos.

Por otra parte, le ha pedido a **Ana** que identifique dónde se guardan las trazas del sistema y le recuerda que para acceder a esos ficheros necesitará tener permisos de administrador en el sistema.

towo (CC BY)

Elaboración propia (Uso educativo no comercial)

En este apartado vamos a ver cómo consultar y tratar la información a nivel de administración de sistemas. Cuando somos los encargados de administrar un sistema, debemos tener herramientas que nos permitan hacer un seguimiento de los datos que arroja el equipo servidor donde se encuentran las aplicaciones.

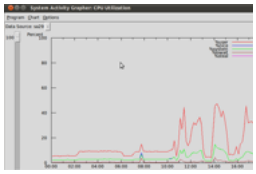
El rendimiento del servidor puede disminuir o ser inexistente debido a diversos motivos. Para investigar qué ocurre en cada caso, es necesario buscar información en los ficheros de registro o mensajes del sistema llamados **logs**, o ejecutar herramientas que permitan realizar un análisis y monitorización del rendimiento.

Podemos obtener datos instantáneos del rendimiento del sistema relativo al funcionamiento de los procesadores, de la memoria, de los dispositivos de entrada y salida, etc., pero también podemos recoger datos periódicamente y almacenarlos en ficheros históricos para consultarlos posteriormente. Estos datos nos proporcionan información muy importante sobre las posibles carencias y cuellos de botella de nuestro sistema.

Existen diversas utilidades para recopilar y hacer un histórico del rendimiento y la actividad de los datos. Estas utilidades recopilan la información del sistema, la almacenan por un periodo de tiempo y calculan los valores medios. En cualquier momento se pueden tomar lecturas de los parámetros del servidor que se determinen, para la resolución de problemas o bien simplemente para consultar el estado de nuestro servidor.

5.1.- Monitorización y evaluación del rendimiento.

Entre las herramientas de monitorización y evaluación del rendimiento disponibles para un servidor Linux se encuentra la herramienta **sar**, incluida en el paquete **sysstat**.



Captura de pantalla del sistema operativo Ubuntu, que se distribuye bajo licencia GNU GPL. (GNU/GPL)

La forma de instalar el **sar** es instalando el paquete **sysstat**, desde Synaptic o bien ejecutando desde el terminal la siguiente orden para Debian o Ubuntu:

```
$ sudo apt-get install sysstat
```

A partir de ahí ya podemos ver datos instantáneos, tan sólo tenemos que tener en cuenta que todos los comandos de la herramienta aceptan como parámetros el número de valores que queremos obtener (intervalo) y cada cuánto tiempo queremos que los capture (segundos). Por ejemplo, para obtener información de 3 valores sobre el uso del procesador, cada segundo, utilizamos la siguiente orden:

```
$ sar 1 3
```

Y el resultado:

```
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
```

Los parámetros "`1 3`" indican que **sar** se ejecutará cada segundo un total de 3 veces, los valores de ejecución se muestran en filas separadas, y la última fila es la media aritmética de todos los valores.

Para conocer la lista completa de los parámetros que podemos utilizar con **sar**, podemos consultar la ayuda del comando (`man sar`). Algunos ejemplos son los siguientes:

- Procesador: `sar -P ALL`
- Memoria: `sar -r`
- Interfaces de red: `sar -n DEV`
- Discos: `sar -d`

Autoevaluación

Señala cuál de las siguientes líneas no es un uso adecuado del comando sar.

- ○ `sudo apt-get install sar`
- ○ `sar 2 5`
- ○ `sar -P ALL`
- ○ `sar`

La instalación de sar se hace a través del paquete sysstat.

Incorrecto

Incorrecto

Incorrecto

Solución

1. Opción correcta
2. Incorrecto
3. Incorrecto
4. Incorrecto

5.2.- Auditorías de control de acceso a los datos. Trazas del sistema (logs).

Captura de pantalla de terminal de Ubuntu.
(GNU/GPL)

La actividad de los programas, sobre todo si se trata de programas que se ejecutan en servidores, queda registrada en ficheros del sistema llamados **logs**. En ocasiones podemos querer examinar los ficheros de trazas del sistema para realizar un control de acceso a los datos, ya que estos ficheros de trazas van almacenando toda la actividad y eventos que ocurren en el equipo: quién entra, qué comandos ejecuta, qué errores muestran las aplicaciones, etc.

En la mayoría de las distribuciones Linux, estos ficheros se guardan en el directorio **/var/log**. Para visualizar su contenido necesitamos permisos de **root**, o pertenecer a un grupo de usuarios con permisos para ver esos ficheros. En este directorio por ejemplo se guarda el fichero **syslog**, que guarda mensajes de trazas de demonios y otros programas como **cron**, **init**, **dhclient**, y algunos mensajes relacionados con el núcleo del sistema operativo. Si deseas consultar estos ficheros de manera gráfica en el entorno gráfico de **Gnome**, puedes ver cómo se hace en esta sencilla guía: Consultar los sucesos del sistema en la interfaz gráfica de Ubuntu 18.04 LTS Desktop

Volviendo al ejemplo de la herramienta **sar** para monitorización de servidores Linux, para hacer que **sar** capture información periódica debemos editar el fichero **/etc/default/sysstat** y la línea que empieza por **ENABLED** ponerla con el valor **ENABLED="true"**. Una vez hecho, reiniciamos el servidor de **sysstat** con la orden:

$/etc/init.d/sysstat start

A partir de ahí el script **/usr/lib/sysstat/sa1**, que se ejecuta por defecto cada 10 minutos empezará a recoger datos de rendimiento (procesador, memoria, disco, red, etc) que se guardarán en el fichero de sistema **/var/log/sysstat/saXX**, donde **XX** indica el día del mes en el que nos encontramos. Por defecto, se guardan los datos de la última semana.

Para consultar de manera gráfica las trazas del sistema de **sar** podemos utilizar la herramienta Isag, disponible a través de Synaptic.

5.3.- Incidencias: identificación y resolución.

Además del control de acceso a los datos, lo interesante del registro de trazas del sistema es consultar la información posteriormente para resolver posibles problemas, ya que generalmente si una aplicación no funciona, o no puede inicializarse, lo que hace es imprimir una traza de error, que puede poner sobre aviso de lo que está ocurriendo.

Dentro del directorio `/var/log` podemos encontrar las trazas del sistema de una aplicación en Ubuntu. Por ejemplo, podemos querer visualizar la actividad del servidor de Odoo con el siguiente comando:

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

El resultado de la instrucción serán las diez primeras líneas del fichero de 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 l
ibrary 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 servero
doo: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:~#
```

Captura de pantalla de terminal de Ubuntu. (GNU/GPL)

Donde podemos interpretar que el servidor de **Odoo** se encuentra ejecutándose y esperando conexiones en el puerto `8069`. Cualquier incidencia quedará reflejada en estos archivos, y será cuestión nuestra identificar la causa y una posible solución.

Autoevaluación

Los mensajes del sistema en Ubuntu los podemos encontrar en archivos almacenados en el directorio `/var/log`.

○ Verdadero ○ Falso

Verdadero
Efectivamente, ese es el directorio donde suelen encontrarse los archivos que guardan información de la actividad del sistema.