

3. Organización e consulta da información¹

- Definición de campos.
- Consultas de acceso a datos.
- Interfaces de entrada de datos e de procesos. Formularios.
- Informes e listaxes da aplicación.
- Cálculos de pedidos, albarás, facturas, asentos predefinidos, rastrexabilidade, producción, etc.
- Gráficos.
- Procesos de extracción de datos en sistemas de ERP-CRM e almacéns de datos.
- Ferramentas de monitorización e de avaliación do rendemento.
- Incidencias: identificación e resolución.

¹ Nos basaremos en Odoo

1. Definición de campos. Consultas de acceso a dato

Los **objetos en Open ERP contienen campos** los cuales permiten introducir datos en la base de datos, estos campos van definidos en el atributo `columns`.

Campos básicos:

- **boolean:** Bolean (True o False)
 - **Sintaxis:** `fields.boolean('Field name', [, Optional Parameters])`
- **integer:** Numero entero
 - **Sintaxis:** `fields.integer('Field Name' [, Optional Parameters]),`
- **float:** Un numero de punto flotante
 - **Sintaxis:** `fields.float('Field Name' [, Optional Parameters]),`
- **char:** Un string de espacio limitado, su longitud es un parámetro requerido
 - **Sintaxis:** `fields.char('Field Name', size=n [, Optional Parameters])`
- **text:** Un campo de texto sin limite
 - **Sintaxis:** `fields.text('Field Name' [, Optional Parameters]),`
- **date:** Un dato
 - **Sintaxis:** `fields.date('Field Name' [, Optional Parameters]),`
- **datetime:** Permite asignar un dato de tiempo
 - **Sintaxis:** `fields.datetime('Field Name' [, Optional Parameters]),`
- **binary:** Dato binario
- **selection:** Esta campo permite seleccionar un dato de varios valores predeterminados
 - **Sintaxis:** `fields.selection(((('n','Unconfirmed'), ('c','Confirmed')), 'field Name' [, Optional Parameters])`

Campos Relacionales

- **one2many:** Este campo expresa una relación uno a muchos entre dos objetos, este campo es obsoleto utilizando una relación `many2one`
 - Sintaxis:** `fields.one2one('other.object.name', 'Field Name')`
`fields.one2many('other.object.name', 'Field relation id', 'Fieldname', optional parameter)`
** Optional parameters:*

- *invisible: True/False*

- *states: ?*

- *readonly: True/False*

- **many2one:** Asocia este objeto con un objeto padre en una relación muchos a uno, por ejemplo, de muchas marcas de autos existe una que pertenece a un vehículo en particular

- **Sintaxis:**

fields.many2one('other.object.name', 'Field Name', optional parameter)

Optional parameters:

- *ondelete: What should happen when the resource this field points to is deleted.*

+ *Predefined value: "cascade", "set null", "restrict", "no action", "set default"*

+ *Default value: "set null"*

- *required: True*

- *readonly: True*

- *select: True - (creates an index on the Foreign Key field)*

- **many2many:**

- **Sintaxis:**

fields.many2many('other.object.name', 'relationobject', 'actual.object.id', 'other.object.id', 'Field Name')

Donde:

- *other.object.name* is the other object which belongs to the relation
- *relation object* is the table that makes the link
- *actual.object.id* and *other.object.id* are the fields' names used in the relation table

Ejemplo:

```
'category_ids':fields.many2many(  
    'res.partner.category',  
    'res_partner_category_rel',  
    'partner_id',  
    'category_id',  
    'Categories')
```

- **one2one**: Este campo se encuentra obsoleto por que una relación uno a uno es realmente una relación muchos a uno (**many2one**).

Campos funcionales:

Un campo funcional es un campo cuyo valor se calcula por una función (en lugar de ser almacenado en la base de datos).

Parámetros:

```
fnct, arg = None, fnct_inv = None, fnct_inv_arg = None, type = "float",fnct_search = None, obj = None, method = False, store = False, multi = False
```

donde

- *fnct* es la función o el método que va a calcular el valor del campo. Debe de haber sido declarada antes de declarar el ámbito funcional.
- *fnct_inv* es la función o método que le permitirá escribir valores en ese campo.
- *Tipo* es el nombre del tipo de campo devuelto por la función. Puede ser cualquier nombre de tipo de campo, excepto la función.
- *fnct_search* le permite definir el comportamiento de búsqueda en ese campo.
- *Method* si el campo se calcula mediante un método (de un objeto) o una función global
- *Store (almacenar)* Si desea almacenar campo en la base de datos o no. El valor predeterminado es False.
- *Multi* es un nombre de grupo. Todos los campos con el mismo *varios* parámetros se calculan en una sola llamada de función.

Las consultas de acceso a datos ya se plantean en otro módulo profesional.

Aún así en Odoon son similares que en cualquier otra herramienta y constan de asistentes para llevarlas a cabo. Se verán mejor en las actividades prácticas.

2. Interfaces de entrada de datos e de procesos. Formularios.

Odoon sigue una **arquitectura MVC**, es decir:

- **El modelo**, define la estructura de los datos.
- **La vista**, describe la interfaz con el usuario o la usuaria.
- **El controlador**, soporta la lógica de negocio de la aplicación.

La **capa modelo** es definida por objetos Python cuyos datos son almacenados en una base de datos PostgreSQL. El **mapeo de la base de datos** es gestionado automáticamente por Odoor, y el mecanismo responsable por esto es el modelo objeto relacional, (**ORM - object relational model**).

La **capa vista** describe la interfaz con el usuario o la usuaria. Las vistas son definidas usando XML, las cuales son usadas por el marco de trabajo (framework) **del cliente web para generar vistas HTML de datos**.

```
<record id="vista_película_form" model="ir.ui.view">
  <field name="name">motion.película.form</field>
  <field name="model">motion.película</field>
  <field name="type">form</field>
  <field name="arch" type="xml">
    <form string="Películas">
      <group col="2" colspan="2"><separator colspan="2" string="Información
      General"/>
        <field name="nombre" select="1"/>
        <field name="codigo" select="1"/>
        <field name="nombre_d" select="1"/>
      </group>
      <group col="2" colspan="2">
        <separator string="Información Secundaria" colspan="2"/>
        <field name="fecha"/>
        <field name="costo"/>
        <field name="numero"/>
        <field name="costo_a"/>
      </group>
      <notebook colspan="4">
        <page string="Notas">
          <field colspan="4" nolabel="1" name="notas"/>
        </page>
      </notebook>
    </form>
  </field>
</record>
```

El formulario resultante:

Las **vistas del cliente web** ejecutan acciones de datos persistentes a través de la **interacción con el servidor ORM**. Estas pueden ser operaciones básicas como escribir o eliminar, pero pueden también invocar métodos definidos en los objetos Python del ORM, ejecutando lógica de negocio más compleja. **A esto es a lo que nos referimos cuando se habla de la capa modelo.**

3. Informes y listados.

Básicamente un informe, de forma general, contiene las siguientes secciones:

- **Seccion 1: Report Header**

Cabecera del informe, donde se imprime una sola vez al inicio del reporte. Puede contener desde el logo de la empresa, fechas, nombre del informe ...

- **Seccion 2: Page Header**

Cabecera de pagina, donde se imprime al inicio de cada pagina impresa. Puede contener anotaciones generales, por ejemplo si es una factura, el nombre y dirección del cliente, número de factura.

- **Seccion 3: Details**

Detalle del reporte, donde las filas o registros que conforman el reporte. Es allí donde se alojan los campos del origen de datos

- **Seccion 4: Report Footer.**

Pie del reporte, donde se imprime una sola vez al finalizar el reporte. Se utiliza esta seccion para imprimir los totales generales, promedios.

- **Seccion 5: Page Footer.**

Pie de pagina, donde se imprime al final de cada pagina. Se utiliza esta seccion para imprimir la paginacion, los totales por pagina

The image shows a screenshot of a report layout with several sections. At the top, there is a header section titled 'Encabezado del informe' containing the title 'Libreta de teléfonos del cliente'. Below this is a page header section 'Encabezado de página' with fields for 'File As', 'Contact Name', and 'Teléfono'. This is followed by a section 'Archivo como encabezado' containing the text '=UC'. The main body of the report is a 'Detalle' section with a table containing columns for 'ID', 'Nombre del contacto', and 'Teléfono'. Below the table is a 'Pie de página' section, and at the very bottom is the 'Pie del informe' section.

Aunque no obligatoriamente debe contener todas las secciones si es aconsejable que se mantenga una estructura por secciones por su sencillez y posibilidad a la hora de utilizar dicha estructura en otros informes.

Los datos que aparecen en el informe finalizado dependen de las **opciones de organización**. En particular, los datos del informe varían según las secciones en las que desee insertar objetos de informe concretos. Por ejemplo, si inserta un objeto de gráfico en la sección *Encabezado de informe*, el gráfico sólo aparecerá una vez al principio del informe y resumirá los datos que contiene el informe. Además, si un objeto de gráfico se añade a la sección *Encabezado de grupo*, aparecerá un gráfico individual al principio de cada grupo de datos y sólo se resumirán los datos relacionados con dicho grupo.

Los informes necesitarán **separar los datos en grupos para facilitar la lectura y el análisis**. Se puede agrupar los datos de un informe para que muestren **relaciones jerárquicas**. Al agrupar datos jerárquicamente, se ordena la información según la relación entre campos.

- Para que el programa reconozca una relación entre los campos principal y secundario, ambos deben pertenecer al **mismo tipo de datos**.
- Los datos del campo principal deben tener un subconjunto de los datos del campo secundario.

- Para que el nivel superior de la jerarquía aparezca en el informe, el valor debe aparecer en los datos secundarios y la fila correspondiente de los datos principales debe estar vacía.
- No puede existir lógica circular en los datos (es decir, A no puede estar relacionado con B si B está relacionado con C y C está a su vez relacionado con A).

Por ejemplo, si se desea mostrar la estructura jerárquica de un departamento, se puede agrupar datos por Id. de empleado y especificar la jerarquía usando el campo de datos que muestra a quién informa el empleado. Otro agrupamiento sería listado de empleados por departamento siendo el campo principal el id del departamento y el secundario los empleados que a él pertenecen.

Se puede en una región de datos para seleccionar o excluir partes del conjunto de datos que utiliza la región de datos.

Los filtros limitan los datos que se muestran al usuario tras la recuperación de todos los datos. Dado que se recupera el conjunto completo de datos y luego se filtra cuando se procesa el informe, es posible que el informe no sea tan bueno como cuando el informe obtiene datos filtrados de otra manera (específicamente, si escribe un código que filtra datos antes de que pasen al informe).

De esta forma en los agrupamientos pueden ser interesantes filtrados de datos, como por ejemplo listado de empleados por departamentos que obtuvieron un mínimo de X ventas o que pertenezcan a tal o cual ciudad

4. Cálculos de pedidos, albaranes, facturas, asientos predefinidos, rastreabilidad, producción, etc.

Se verá en las actividades prácticas.

5. Gráficos

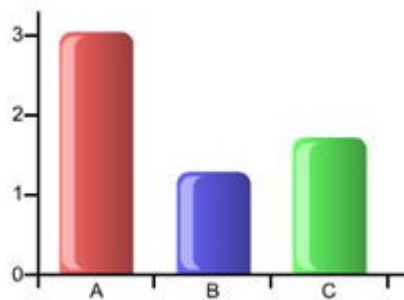
Los **gráficos y las tablas** representan e interpretan información procedente de diferentes fuentes, de forma clara, precisa y ordenada. Casi todos los tipos de información puede organizarse en una tabla de datos y ser representada en algún tipo de gráfico.

Según las características y la cantidad de datos, conviene utilizar uno u otro gráfico.. Los gráficos permiten visualizar la información contenida en las tablas de manera rápida y sencilla, demostrando con mayor claridad la relación que estos datos tienen entre sí.

Los más conocidos son:

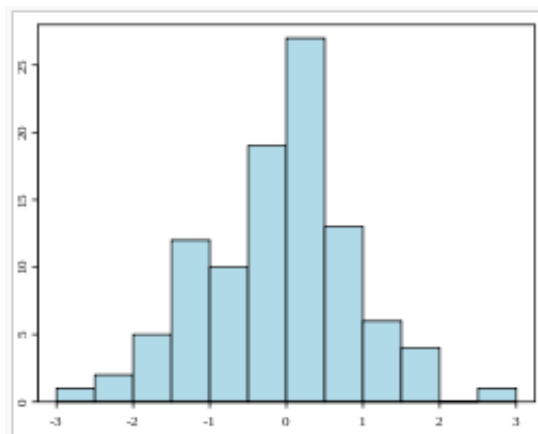
A- Gráficos de barras

Son aquellos que emplean rectángulos (barras) que se colocan paralelamente. La altura indica la frecuencia de ese dato. Los gráficos de barras, permiten representar información numérica en forma clara y ordenada, para comunicarla a otras personas. Con la información representada en los gráficos puedes interpretar rápidamente y de manera visual la información, facilitando su posterior análisis.



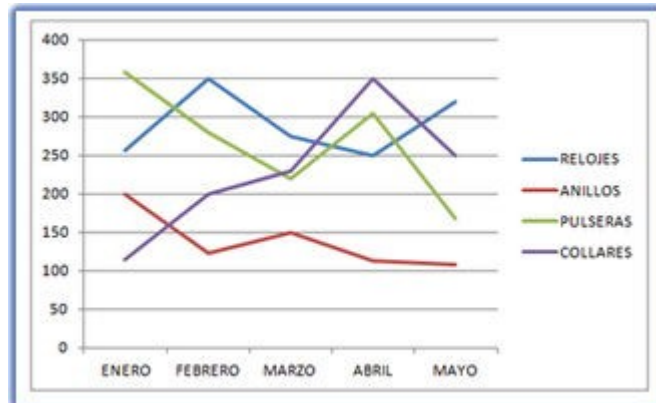
Para construir un gráfico de barras los **datos numéricos van en el eje vertical** (determinando la altura de las barras) y las **categorías en el eje horizontal**. También pueden representarse alternando esta orientación. Los gráficos de barras son usados para comparar dos o más valores.

Una variación es el **histograma** que es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados, ya sea en forma diferencial o acumulada. Sirven para obtener una "primera vista" general de la distribución de la muestra, respecto a una característica, cuantitativa y continua, de la misma.



B- Gráficos de líneas o lineal

Es un conjunto de puntos conectados por una línea en un sistema cartesiano, que **muestran tendencias** de una variable a lo largo de un período de tiempo u otra variable contiuna.



C.- Gráficos de sectores o “tarta”

Es un diagrama en círculo que representa visualmente información en porciones imaginarias de una tarta.



Es muy útil para la representación de **porcentajes o proporciones**. Generalmente deben de ser superiores a cuatro. El empleo de tonalidades o colores facilita su visualización. A pesar de su extenso uso no suele ser el más recomendable ya que no facilita la percepción de las proporciones y sus diferencias siendo preferible para ello el diagrama de barras. Su construcción se realiza con la siguiente fórmula.

$$N^{\circ} = \frac{360^{\circ} \times f_a}{N}$$

D- Pictogramas

Son los más llamativos, ya que se representan por medio de dibujos, se reemplaza las barras por dibujos. Se usan **para lograr el interés masivo del público** y muy utilizados en los medios de comunicación para representar estudios estadísticos.



6. Procesos de extracción de datos en sistemas de ERP-CRM y almacenes de datos.

Se verá en las actividades prácticas.

7. Herramientas de monitorización y de evaluación del rendimiento.

Las **herramientas de monitoreo** de aplicaciones proporcionan a los administradores la información que necesitan para **descubrir rápidamente, aislar y resolver problemas que impactan negativamente en el rendimiento de una aplicación**. Tales herramientas pueden ser específicas para una aplicación en particular, o monitorizar varias aplicaciones en la misma red, recogiendo datos sobre el uso del CPU cliente, demandas de memoria, rendimiento de la base de datos y ancho de banda. Estas herramientas no son específicas para sistemas ERP sino que permiten la monitorización de otros tipos de sistemas informáticos.

Entre las herramientas que podemos citar:

- **ManageEngine Applications Manager:** presente módulos de monitoreo para SAP y Oracle E-Business Suite.
- **Ingeo:** para Odoo y otras de software libre
- **PandoraCMS:** para monitorización específica de sistemas CRM

- **SocialMention:** además del monitoreo de CRM de la empresa, las posibles cuentas de la misma en Twitter, Google+, Facebook y demás redes sociales.

Además las propias aplicaciones ERP tienen sus propias herramientas de auditoría, monitoreo y control del rendimiento.

8. Incidencias: identificación e resolución

Se verá en las actividades prácticas.