

SGE05.- Desarrollo de componentes.

1.- INTRODUCCIÓN.

En esta unidad de trabajo, en contraposición con la unidad de trabajo anterior, veremos como realizar modificaciones en el ERP utilizando el lenguaje de programación Python. De esta forma, desarrollaremos módulos que nos permitirán modificar el ERP de una forma automática y sencilla, simplemente instalando el módulo.

Las modificaciones que aprenderemos a realizar serán muy similares a las realizadas en la UT4 pero el procedimiento llevado a cabo para realizarlas será totalmente diferente.

2.- MODELO VISTA CONTROLADOR (MVC).

¿Utilizas habitualmente hojas de cálculo, como por ejemplo Calc de OpenOffice.org? En ese caso, sabrás que al introducir datos en las celdas, los mismos datos podemos verlos de varias formas. Es posible ver los datos en la hoja de cálculo donde los hemos introducido, o mediante un gráfico que puede ser de barras, circular, etc.

Esos programas implementan el patrón Modelo-Vista-Controlador (MVC), el modelo lo constituyen los datos que hemos introducido en las celdas. Las vistas se encargan de mostrar los datos de la forma que se seleccione.

El MVC divide una aplicación en tres componentes:

- Los datos de la aplicación (modelo).
- La interfaz del usuario (vista).
- El controlador, el cual define la forma en que la interfaz reacciona a la entrada del usuario.

Con esto se consigue separar los datos (modelo) de la interfaz de usuario (vista), de manera que los cambios en la interfaz no afectan a los datos y viceversa, es decir, que los datos pueden ser cambiados sin afectar a la interfaz de usuario.

En Odoo, el MVC se implementa de la siguiente forma:

- El modelo son las tablas de la base de datos. Vienen representadas por una clase en Python.
- Las vistas son los archivos XML que definen la interfaz de usuario del módulo.
- El controlador son los objetos creados en Python. En módulos sencillos esta capa puede no ser necesaria.

3.- DESARROLLANDO UN MÓDULO EN ODOO.

El primer concepto que tenemos que tener claro es que un módulo es un directorio dentro de la carpeta addons de nuestro servidor. Este directorio tendrá una serie de archivos que definirán el módulo. Es vital que sepamos realizar las siguientes tareas antes de comenzar con el desarrollo del módulo:

- Ser capaces de arrancar y parar el servicio de Odoo. Necesitaremos hacerlo para que se recarguen los archivos python de nuestro módulo.

Como nota adicional, os incluyo la referencia a la documentación oficial en la que aparece mucha información interesante necesaria para la creación de módulos en Odoo 10.

3.1.- ESTRUCTURA DEL MÓDULO.

En Odoo 10 la estructura básica del módulo puede ser creada automáticamente. Para ello utilizaremos el siguiente comando:

```
./odoo-bin scaffold libreria addons
```

Para saber más

Puedes ver el primer vídeo en el que se explica como generar la estructura básica de un módulo en Odoo 10 y que partes tiene.



([link: https://www.youtube.com/watch?v=E7kT6o_HCUE](https://www.youtube.com/watch?v=E7kT6o_HCUE))

Resumen textual alternativo ([link: documento.html](#))

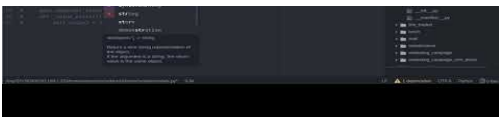
3.2.- CREAR MODELOS.

Los modelos van a corresponder con tablas de bases de datos y son el fundamento de nuestro módulo. En este ejemplo, vamos a realizar un módulo para una librería. Nuestro objetos modelo serán libro y categoria. Los archivos de modelo se escriben en python y siguen la arquitectura ORM que nos proporciona Odoo.

Todas las clases de modelo en Odoo deben extender de la clase Model de ORM.

Para saber más

En el siguiente vídeo puedes ver como crear los modelos y añadirles campos básicos.



([link: https://www.youtube.com/watch?v=DurV9im-PdM](https://www.youtube.com/watch?v=DurV9im-PdM))

Resumen textual alternativo ([link: documento.html](#))

Es importante que recuerdes añadir siempre un campo **name** a cada uno de los modelos que crees. Gracias a este campo podrás definir relaciones sin problemas y la búsqueda en las vistas funcionará directamente.

3.3.- CREACIÓN DE LAS VISTAS.

Las vistas se definen de manera muy similar a como habíamos visto en la UT4. Tendremos vistas de tipo formulario, árbol (tree), kamban, búsqueda, etc. En nuestro módulo las meteremos en un directorio concreto (views), de cara a tenerlas identificadas. Podremos meter varias vistas dentro de un mismo fichero xml o bien separarlas en varios ficheros xml.

Para saber más

En el siguiente vídeo puedes ver como definir las vistas y los menús para el módulo librería que estamos creando:



([link: https://www.youtube.com/watch?v=Q7ey7eeE494](https://www.youtube.com/watch?v=Q7ey7eeE494))

Resumen textual alternativo ([link: documento.html](#))

3.4.- CREACIÓN DE RELACIONES.

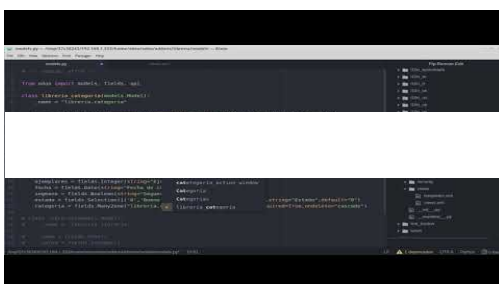
Generalmente los modelos que creamos tendrán relaciones entre si o con otros objetos del sistema. Las relaciones que podemos crear en Odoo son:

- one2many
- many2one
- any2many

Las tres son análogas a las relaciones SQL que ya conocéis del mundo de las bases de datos. Es importante destacar que para crear una relación one2many en un modelo debemos crear la relación many2one en el modelo con el que establecemos la relación. Por otra parte, es vital que los modelos tengan un campo name para que las relaciones funcionen.

Para saber más

En el siguiente vídeo puedes ver como crear relaciones en Odoo 10 entre los objetos libro y categoría:



([link: https://www.youtube.com/watch?v=HNsTUG-MWlc](https://www.youtube.com/watch?v=HNsTUG-MWlc))

Resumen textual alternativo ([link: documento.html](#))

3.5.- VISTAS DE BÚSQUEDA.

Las vistas de búsqueda permiten redefinir la forma en la que buscamos en Odoo. Tenemos dos opciones a la hora de realizar una vista de búsqueda:

- campos de búsqueda: Simplemente son campos del modelo por los que es posible buscar. Por ejemplo, **descripción** dentro del objeto libro.
- filtros. Son búsquedas ya definidas con un criterio definido por nosotros. Digamos que el cliente está interesado en buscar siempre los libros con un precio menor o igual que cinco. Podríamos escribir entonces el siguiente filtro:

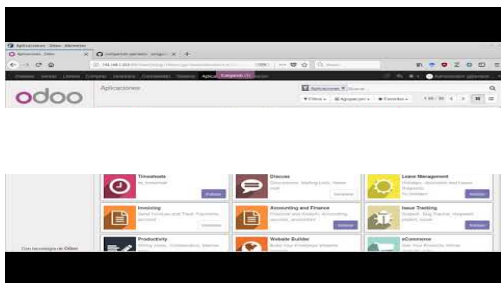
```
<filter name="baratos" domain="[('precio','<=',5)]"/>
```

Para saber más

La lista de operaciones a la hora de escribir un dominio son las podéis encontrar en la siguiente página. Utilizando dominios es posible escribir expresiones lógicas complejas que se adapten a cualquier necesidad.

github.com (link: <https://github.com/sotogarcia/odoo-development/wiki/Comparison-operators/>)

En el siguiente vídeo podéis ver como crear una vista de búsqueda para el módulo librería:



(link: <https://www.youtube.com/watch?v=PblyhtM0k8Q>)

Resumen textual alternativo (link: [documento.html](#))

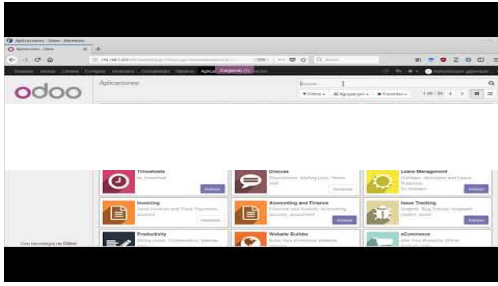
3.6.- CAMPOS CALCULADOS.

Normalmente en un modelo tendremos campos que el usuario deberá introducir a través de las vistas. De todas formas, puede existir la necesidad de realizar un campo calculado. Por ejemplo, supongamos que en el objeto libro tenemos los campos importe y numejemplares. Podríamos crear el campo calculado importetotal en el cual tendríamos el producto de los dos campos anteriores. Este campo podría ser introducido por parte del usuario y tan solo se le mostraría en solo lectura.

Para crear un campo calculado en python debemos escribir una función dentro de la clase con el modelo, con una estructura determinada.

Para saber más

En el siguiente vídeo puedes ver los detalles de como crear un campo calculado en python:



(link: <https://www.youtube.com/watch?v=fqHLAG9K4RA>)

Resumen textual alternativo (link: [documento.html](#))

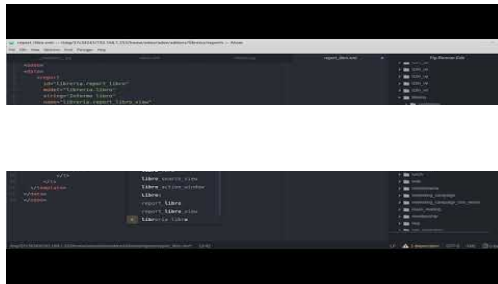
3.7.- CREACIÓN DE INFORMES.

Al igual que en la UT4, podemos crear informes para nuestros objetos en Odoo. Un informe se crea a través del motor de informes **Qweb** y el resultado suele ser un PDF (aunque podría tener otro formato), que se le permite descargar al usuario.

El lenguaje utilizado para crear los informes es idéntico al utilizado en la UT4.

Para saber más

En el siguiente vídeo puedes ver todos los detalles sobre como crear un informe, en este caso para imprimir los detalles de un libro:



(link: <https://www.youtube.com/watch?v=JR6OVqt4oko>)

Resumen textual alternativo (link: [documento.html](#))

3.8.- PERMISOS.

Una vez creado el módulo, es interesante dejar creados grupos que gestionen los accesos a las partes de nuestro módulo. En el caso de la librería crearemos un único grupo Librería- Responsable que tendrá todos los privilegios sobre el módulo librería. De manera análoga podríamos crear otros grupos que diesen permisos más restrictivos sobre el modulo.

Para ello debemos editar dos archivos, un security.xml y un archivo .csv donde indicaremos que permisos tiene cada grupo sobre cada objeto de Odoo individual.

Para saber más

En el siguiente vídeo puedes revisar como realizar esta tarea siguiendo el ejemplo que estamos desarrollando:



(link: <https://www.youtube.com/watch?v=6miubFgiqmg>)

Resumen textual alternativo (link: [documento.html](#))

3.9.- RETOQUES FINALES: ICONO Y DATOS PRECARGADOS.

Este último apartado hace referencia a como establecer un icono para nuestro módulo y como precargar datos de cara a que el módulo no aparezca totalmente vacío.

Para saber más

En el siguiente vídeo puedes ver como realizar estos últimos pasos de manera sencilla y de esta manera retocar tu módulo y dejarlo preparado para su distribución:



(link: https://www.youtube.com/watch?v=Qcf_50hxnVU)

Resumen textual alternativo (link: [documento.html](#))