

# OpenAcademy

Curso	Programación de OpenERP/Odoo
Departamento	Organización de Empresas
Profesor	Jose Zambudio Bernabeu
Descripción	Ejercicio Realización de un módulo Odoo/OpenERP

## Contenido

- 1 Introducción
- 2 Ejercicios

## 1. Introducción

OpenAcademy es un módulo Odoo/OpenERP orientado a la gestión de una academia en la que se tiene un cierto número de profesores y alumnos, y en la que se imparten cursos de varios tipos.

En nuestro módulo podremos:

- Gestionar todos los cursos creados, ver los actualmente activos.
- Gestionar los profesores dados de alta. Ver que cursos tienen estos.
- Gestionar los alumnos dados de alta.
- Podremos designar un número de sesiones a un curso.
- Podremos dar de alta a un alumno (asistente) en una sesión de un curso, para poder mantener un control de los asientos de la clase.

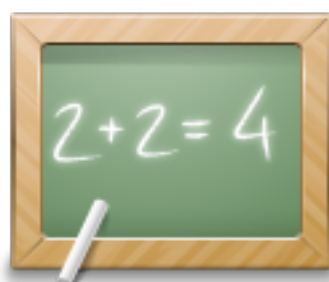
## 2. Ejercicios

(A) Realizar un módulo Odoo/OpenERP llamado **openacademy**.

(B) La estructura de nuestro módulo será:

- **openacademy/**
  - data/
  - demo/
  - i18n/
  - report/
  - security/
  - static/
  - views/
  - wizards/
  - workflows/

(C) Añadir el siguiente icono para el módulo anteriormente creado:



(D) Para el control de profesores y de alumnos vamos a utilizar el modelo “**res.partner**” que nos proporciona el módulo “**base**”.

Para ello deberemos realizar ciertas modificaciones en dicho modelo.

Por tanto, se pide que nuestro módulo OpenERP dependa del módulo “**base**” para poder realizar las modificaciones oportunas.

(E) Dicho módulo tendrá que contener los siguientes modelos:

(E)(1) **Attendee** que debe heredar de `orm.Model`.

Este modelo será utilizado para el control de los asistentes.

(E)(2) **Course** que debe heredar de `orm.Model`.

Este modelo será utilizado para el control de los cursos de nuestra academia.

(E)(3) **Session** que debe heredar de `orm.Model`.

Este modelo será utilizado por el modelo **course** para especificar las sesiones que va a contener, y cuales van a ser sus profesores.

(F) Para cada uno de los modelos creados en el punto anterior, deberemos indicar una descripción de estos.

(G) Deberemos añadir los siguientes fields (columnas):

Course		
<b>image</b> (imagen)	fields.binary	<ul style="list-style-type: none"> <li>- Con nombre del label “<i>Image</i>”.</li> <li>- Añadimos también una <b>ayuda</b> para el usuario con el siguiente texto: “<i>This field holds the image used as image for the product, limited to 1024x1024px.</i>”</li> </ul>
<b>name</b> (nombre)	fields.char	<ul style="list-style-type: none"> <li>- Con nombre del label “<i>Title</i>”.</li> <li>- Como máximo va a tener un tamaño de 128 caracteres.</li> <li>- Vamos a permitir al usuario traducir dicho campo a varios idiomas.</li> <li>- Es obligatorio introducir este campo.</li> </ul>
<b>description</b> (descripción)	fields.text	<ul style="list-style-type: none"> <li>- Con nombre del label “<i>Description</i>”.</li> <li>- Vamos a permitir al usuario traducir dicho campo a varios idiomas.</li> </ul>
<b>active</b> (activo)	fields.boolean	<ul style="list-style-type: none"> <li>- Con nombre del label “<i>Active</i>”.</li> </ul>

## PROGRAMACIÓN DE OPENERP/ODOO

Session		
<b>name</b> (nombre)	fields.char	<ul style="list-style-type: none"> <li>- Con nombre del label "<i>Title</i>".</li> <li>- Como máximo va a tener un tamaño de 128 caracteres.</li> <li>- Es obligatorio introducir este campo.</li> </ul>
<b>date_start</b> (fecha de inicio)	fields.date	<ul style="list-style-type: none"> <li>- Con nombre del label "<i>Start date</i>".</li> </ul>
<b>duration</b> (duración)	fields.float	<ul style="list-style-type: none"> <li>- Con nombre del label "<i>Duration</i>".</li> <li>- Añadimos también una ayuda para el usuario con el siguiente texto: "<i>Duration in days</i>"</li> </ul>
<b>seats</b> (asientos)	fields.integer	<ul style="list-style-type: none"> <li>- Con nombre del label "<i>Number of Seats</i>".</li> </ul>
<b>active</b> (activo)	fields.boolean	<ul style="list-style-type: none"> <li>- Con nombre del label "<i>Active</i>".</li> </ul>
<b>state</b> (estado)	fields.selection	<ul style="list-style-type: none"> <li>- Con nombre del label "<i>State</i>".</li> <li>- Es obligatorio introducir este campo.</li> <li>- Vamos a indicar que dicho campo no puede modificarlo el usuario, es decir, sólo podrá leer su valor.</li> <li>- Estados:                         <ul style="list-style-type: none"> <li>• draft -&gt; Con nombre del label "<i>Draft</i>" (Borrador)</li> <li>• confirmed -&gt; Con nombre del label "<i>Confirmed</i>" (Confirmado)</li> <li>• done -&gt; Con nombre del label "<i>Done</i>" (Hecho)</li> </ul> </li> </ul>

(H) Para el modelo **Session** debemos indicar que los registros deben ordenarse por el campo "**date\_start**".

(I) Para el modelo **Session** debemos indicar a OpenERP que los siguientes fields deben tener por defecto los siguientes valores:

active	True
state	draft
date_start	(Fecha actual)

- (J) Para el modelo **Course** debemos crear 2 restricciones para la creación de nuevos registros. (Esto en clase lo hemos explicado pero no lo hemos hecho, os lo añado como ejercicio para que busquéis en otros módulos como están implementados y como podéis resolver esto)
- (J)(A) A nivel de base de datos, debemos indicar a OpenERP que en el modelo **Course**, no pueden existir 2 cursos con el mismo nombre, es decir, no podemos tener 2 registros con el mismo valor para el field “name”.
- (J)(B) A nivel de aplicación, debemos indicar a OpenERP que en el modelo **Course**, no podemos crear un registro el cual su nombre y su descripción sean iguales.
- (K) Creación de menús. Debemos crear la siguiente estructura de menús:
- Un menú principal llamado “**Open Academy**”  
(id: openacademy\_menu\_root)
    - Un menú folder llamado “**Open Academy**”  
(id: openacademy\_menu)
      - ➔ Un menú llamado “**Courses**”  
(id: menu\_courses)  
(action: action\_courses)
      - ➔ Un menú llamado “**Sessions**”  
(id: menu\_sessions)  
(action: action\_sessions)
    - Un menú folder llamado “**Personal**”  
(id: openacademy\_menu)
      - ➔ Un menú llamado “**Instructors**”  
(id: menu\_instructors)  
(action: action\_instructors)

## (L) Creación de actions:

### (L)(A) action\_courses

Action para mostrar registros del modelo: **course**

Tipo de vista: **form**

Vistas que incluye: **tree,form**

Añadir un mensaje de ayuda cuando no tengan registros.

### (L)(B) action\_sessions

Action para mostrar registros del modelo: **session**

Tipo de vista: **form**

Vistas que incluye: **tree,form,calendar**

### (L)(C) action\_instructors

Action para mostrar registros del modelo: **res.partner**

Tipo de vistas: **form**

Vistas que incluye: **kanban,tree,form**

Debe mostrar únicamente los registros que no son compañía.

**[('is\_company', '=', False)]**

Por favor, para no causar confusión, vamos a seguir todos los mismos nombres de fields, y por tanto, programando TODO en inglés. Además se recomienda seguir las convenciones nombradas en clase.