

PRÁCTICA ODOO 2017
ACADEMIA FORTE



Grupo de Trabajo GP02

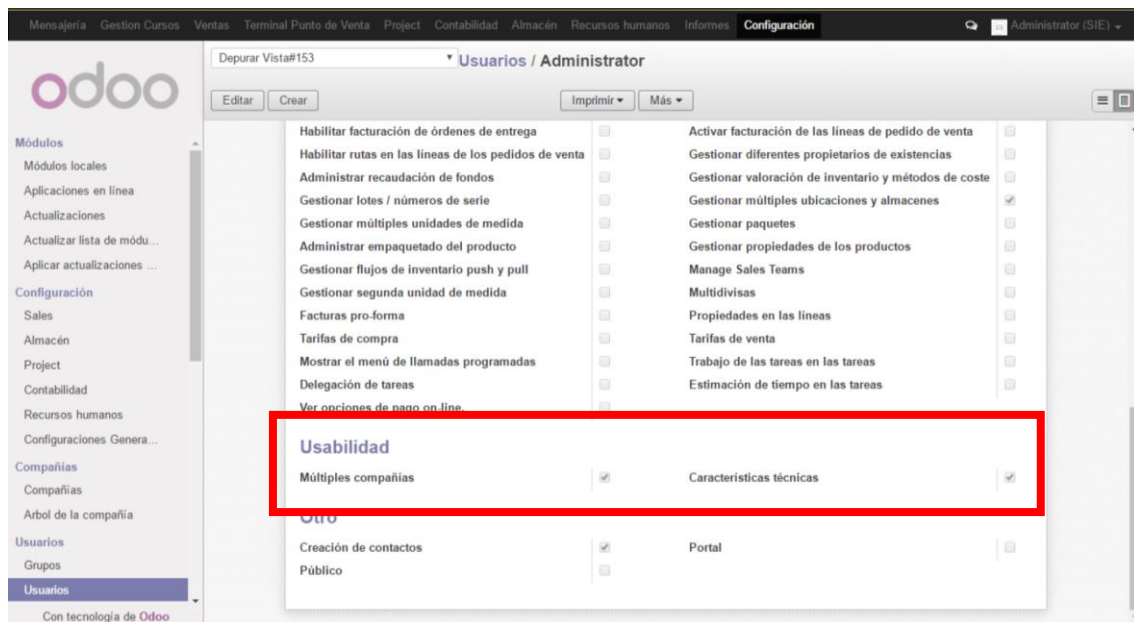
Rodrigo Díaz-Hellín Valera

Javier Martín Moreno

ÍNDICE

| | |
|--------------------------------|-----------|
| 1.- EJERCICIO 1 | 4 |
| 2.- EJERCICIO 2 | 7 |
| 3.- EJERCICIO 3 | 10 |
| 4.- EJERCICIO 4 | 11 |
| 5.- EJERCICIO 5 | 12 |
| 6.- EJERCICIO 6 | 13 |
| 7.- EJERCICIO 7 | 14 |
| 8.- EJERCICIO 8 | 15 |
| 9.- EJERCICIO 9 | 17 |
| 10.- EJERCICIO 10 | 19 |
| 11.- EJERCICIO 11 | 22 |
| 12.- EJERCICIO 12 | 23 |

(*) Antes de realizar cualquier ejercicio hemos realizado unos cambios respecto al usuario **Administrador**. Los cambios se encuentran en el apartado de *Configuración-Usuarios-Usuarios*. Una vez dentro, hemos editado el administrador en el contenido de usabilidad hemos seleccionado las dos opciones que nos proporciona tanto la de **Múltiples Compañías** como la de **Característica técnicas**.



1.- EJERCICIO 1

La Academia Forte posee tres centros de formación.

| CENTRO CIUDAD REAL | CENTRO VALDEPEÑAS | CENTRO MIGUELTURRA |
|---------------------------|---------------------------|---------------------------|
| Calle La Mata, 25 | Calle Quijote, 21 | Calle Ancha, 20 |
| Ciudad Real | Valdepeñas | Miguelturra |
| Responsable: Pedro Moreno | Responsable: Jorge Pietro | Responsable: María Moreno |
| NIF: V5770878F | NIF: D95114476 | NIF: U4523012E |

Para este primer ejercicio, lo primero que hemos hecho es crear la Academia Forte. Para ello, nos hemos dirigido a *Configuración-Compañías-Compañías*. Aquí, hemos hecho click en el botón *crear* y hemos rellenado los campos de nombre de compañía ya que solo teníamos ese dato.

The screenshot shows the Odoo web interface for configuring a company. The top navigation bar includes 'Mensajería', 'Gestión Cursos', 'Ventas', 'Terminal Punto de Venta', 'Project', 'Contabilidad', 'Almacén', 'Recursos humanos', 'Informes', and 'Configuración'. The user is logged in as 'Administrator (SIE)'. The main header shows 'Depurar Vista#111' and 'Compañías / Academia Forte'. The left sidebar lists various modules and configuration options. The main content area is titled 'Compañías / Academia Forte' and contains a form for creating a new company. The 'Nombre de la compañía' field is highlighted with a red box and contains the text 'Academia Forte'. Below this field is a dropdown menu for 'Compañía matriz'. The form also includes sections for 'Información general', 'Configuración', 'Configuración de informes', and 'Pagos fuera de plazo'. The 'Información general' section includes fields for 'Empresa', 'Dirección', 'Lema de la compañía', and 'Sitio web'. The 'Configuración' section includes fields for 'Teléfono', 'Fax', 'Email', 'ID de impuesto', and 'Registro de compañía'. The 'Configuración de informes' section includes a table for 'Cuentas bancarias' with columns for 'Número de cuenta', 'Nombre del banco', 'Mostrar en informes', and 'Propietario cuenta'. The 'Pagos fuera de plazo' section is currently empty.

Una vez realizado esto, procedemos a la creación de los distintos centros. De igual manera hacemos el click en *crear* e introducimos la información del centro como el nombre, la dirección y la ciudad.

El único detalle distinto de la creación de la compañía es que en los centros seleccionaremos la opción de **Compañía Matriz** y escogemos **Academia Forte**.

The screenshot shows the Odoo 'Compañías' configuration page for 'Centro Ciudad Real'. The page is titled 'Compañías / Centro Ciudad Real' and has a 'Depurar Vista#111' link. The left sidebar shows the 'Compañías' menu. The main content area has a red box highlighting the 'Nombre de la compañía' field, which is 'Centro Ciudad Real', and the 'Compañía matriz' dropdown, which is 'Academia Forte'. Below this, there are tabs for 'Información general', 'Configuración', 'Configuración de informes', and 'Pagos fuera de plazo'. The 'Información general' tab is active, showing fields for 'Empresa', 'Dirección', 'Teléfono', 'Fax', 'Email', 'ID de impuesto', 'Registro de compañía', 'Lema de la compañía', and 'Sitio web'. The 'Dirección' field is filled with 'Calle La Mata, 25', 'Ciudad Real', 'Ciudad', '13001', and 'España'. The 'Lema de la compañía' field is filled with 'Por ejemplo, Global Business Solutions'. The 'Sitio web' field is filled with 'Por ejemplo, www.odoo.com'. Below the tabs, there is a section for 'Cuentas bancarias' with a table showing 'Número de cuenta', 'Nombre del banco', 'Mostrar en informes', and 'Propietario cuenta'. The table has one row with 'Añadir un elemento' in the first column.

De igual manera para los otros dos centros restantes. Si seleccionamos *Árbol de la compañía*, la organización de la compañía tiene que quedar de la siguiente manera.

The screenshot shows the Odoo 'Árbol de la compañía' configuration page. The page is titled 'Árbol de la compañía' and has a 'Depurar Vista#112' link. The left sidebar shows the 'Compañías' menu. The main content area shows a table with two columns: 'Nombre de la compañía' and 'Empresa'. The table has five rows: 'Academia Forte', 'Centro Ciudad Real', 'Centro Miguelurra', 'Centro Valdepeñas', and 'Your Company'. The 'Empresa' column is filled with 'Academia Forte', 'Centro Ciudad Real', 'Centro Miguelurra', 'Centro Valdepeñas', and 'Your Company' respectively.

Para la creación de los responsables lo primero que hemos hecho es instalar el módulo de **Recursos Humanos** con el cual gestionaremos los empleados.

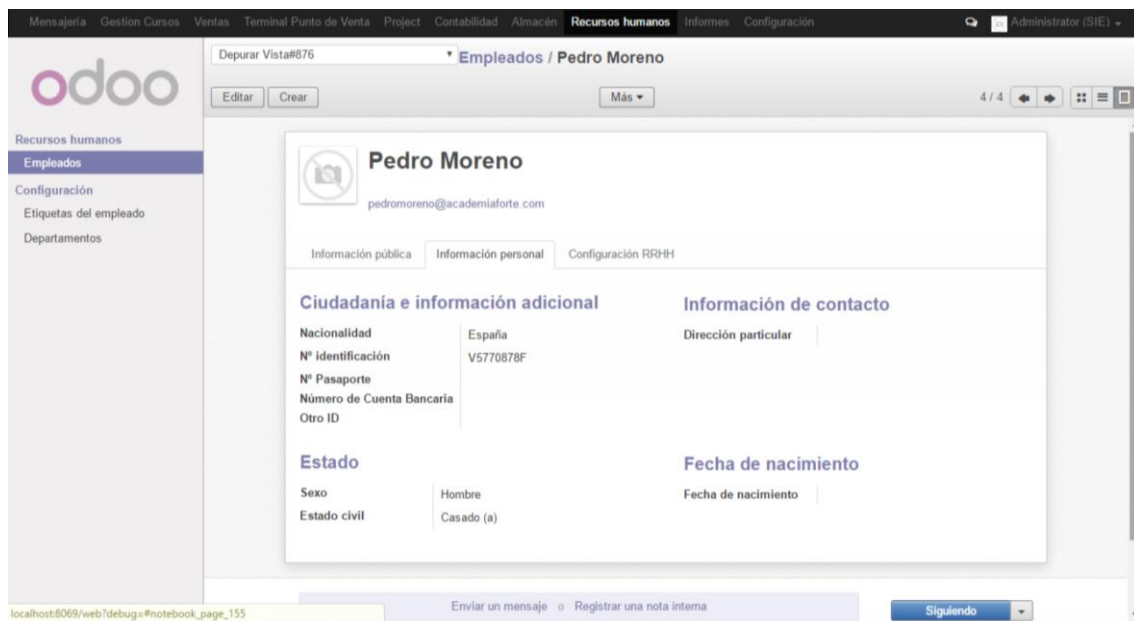


Una vez **instalado el módulo**, nos hemos dirigido a él y hemos seleccionado el apartado de *empleados*. A continuación hemos creado los tres empleados correspondientes a cada centro. La interfaz es similar a la de creación de los centros.

En la pestaña de *Información pública*, el apartado *Dirección de trabajo*, hay que escoger el centro, previamente creado, correspondiente al empleado. Además completar el resto de la información correspondiente. En el apartado de compañía hay que seleccionar **Academia Forte**. También en la zona de Cargo hay que asignarle el título de trabajo que en nuestro caso es **responsable** para los tres empleados



En la siguiente pestaña del empleado podemos completar la información con la nacionalidad, el NIF, el sexo y el estado civil.



2.- EJERCICIO 2

La Academia ofrece cursos y para cada curso puede haber varias ediciones.

| NOMBRE | PRECIO |
|----------|---------|
| ODOO | 70,00 € |
| DYNAMICS | 65,00 € |
| SAP | 120,00€ |

| CURSO | EDICIÓN | FECHA INICIO | FECHA FIN | LUGAR |
|----------|---------|--------------|------------|-------------|
| ODOO | Ed. 1 | 01/05/2016 | 31/05/2016 | C. Real |
| ODOO | Ed. 2 | 05/06/2016 | 25/05/2016 | Miguelturna |
| DYNAMICS | Ed. 1 | 03/05/2016 | 25/05/2016 | Valdepeñas |
| SAP | Ed. 1 | 03/05/2016 | 03/07/2016 | C. Real |

Para poder realizar este paso, hemos tenido que crear un nuevo módulo, el cual lo hemos llamado **Gestión de Cursos**. Dentro de este módulo tenemos varios archivos de los cuales solo modificaremos tres de ellos. Esos tres archivos son:

- `_openerp_`: Archivo Python.
- `curso_view`: Archivo xml.
- `module`: Archivo Python.

El siguiente fragmento de código pertenece al archivo “module.py”, en el cual crearemos todas las clases necesarias para esta práctica. En este caso mostramos las dos primeras clases que vamos a utilizar “Cursos”, y “Ediciones”.

```

1  # -*- coding: utf-8 -*-
2  from openerp.osv import osv, fields
3
4  from types import *
5
6  #---CLASE CURSO---
7
8  class curso(osv.osv):
9
10     _name = "sie.curso"
11     _description = "Cursos"
12     _columns = {
13         'nombre':fields.char('Nombre', size=30, required=True),
14         'precio':fields.char('Precio €', size=10, required=True),
15     }
16
17     # Restriccion unica al campo NIF
18     _sql_constraints = [
19         ('nombre_unico', 'unique(nombre)', 'El nombre del curso debe ser unico'),
20     ]
21     curso()
22
23 #---CLASE EDICION ---
24 class edicion(osv.osv):
25     _name = "sie.edicion"
26     _description = "Ediciones"
27     _columns = {
28         'curso':fields.many2one('product.template','Curso', ondelete='cascade'),
29         # 'edicion':fields.char('Edicion',size=20,required=True),
30         'edicion':fields.many2one('product.template', 'Edicion', ondelete='set null'),
31         'fecha_inicio':fields.date('Fecha_inicio', size=20),
32         'fecha_fin':fields.date('Fecha_fin', size=64),
33         'centro':fields.many2one('res.company', 'Centro',ondelete='cascade',select=True),
34         'estado':fields.boolean('¿Finalizado?'),
35     }
36
37     _sql_constraints = [
38         ('curso_unico', 'unique(curso,edicion)', 'Curso con edicion ya seleccionada!'),
39     ]
40
41     edicion()

```

Además, también hemos tocado el archivo “curso_view.xml”, en el cual organizaremos el diseño del módulo para el correcto funcionamiento del mismo.

```

103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138

```

```

<record id="alumno_filter" model="ir.ui.view">
  <field name="name">alumno.select</field>
  <field name="model">sie.alumno</field>
  <field name="type">search</field>
  <field name="arch" type="xml">
    <search string="Buscar alumno">
      <group col="5" colspan="2">
        <field name="nombre" select="1"/>
        <field name="direccion" select="1"/>
        <field name="localidad" select="1"/>
        <field name="nif" select="1"/>
        <field name="curso" select="1"/>
      </group>
    </search>
  </field>
</record>
<record model="ir.ui.view" id="edicion_form">
  <field name="name">sie.edicion.form</field>
  <field name="model">sie.edicion</field>
  <field name="type">form</field>
  <field name="arch" type="xml">
    <form string="Ediciones">
      <group col="2">
        <field name="curso"/>
        <field name="edicion"/>
        <field name="fecha_inicio"/>
        <field name="fecha_fin"/>
        <field name="centro"/>
        <field name="estado"/>
      </group>
    </form>
  </field>
</record>

```

Para este Ejercicio, hemos creado dos modelos uno para **cursos** y otro para **ediciones**.

En cursos, hay dos campos uno para insertar el nombre y otro para el precio. Mientras que en el de ediciones podemos seleccionar el *ID del curso (previamente creado)*, la *edición*, la *fecha de comienzo*, la *fecha de fin* y el *centro*.

En el campo de edición, debemos crear la edición, hay que ponerle el nombre, el tipo de producto, que en nuestro caso hemos puesto servicio, el precio del producto además se puede añadir la compañía.

The screenshot shows the 'Ediciones' form in Odoo. The form is titled 'Ediciones / sie.edicion,1 / Ed. 1 ODOO'. It has a sidebar with navigation links: 'Gestionar Cursos', 'Cursos', 'Ediciones' (selected), 'Gestionar Profesores', 'Profesores', 'Gestionar Alumnos', 'Alumnos', 'Gestionar Inscripciones', and 'Inscripciones'. The main form area has a 'Nombre del producto' field with the value 'Ed. 1 ODOO'. Below this is a checkbox 'Puede ser vendido' which is checked. There are tabs for 'Información', 'Abastecimientos', 'Inventario', 'Ventas', 'Variantes', and 'Contabilidad'. The 'Información' tab is active, showing fields for 'Tipo de producto' (set to 'Servicio'), 'Precio de venta' (70.00), 'Activo' (checked), 'Código EAN13', 'Establecer un código EAN personalizado', 'Referencia interna', and 'Compañía' (Academia Forte). There is also a text area for 'Describe aquí las características del producto...'. At the bottom, there are buttons for 'Enviar un mensaje', 'Registrar una nota interna', and 'Siguiendo'.

Una vez realizado este procedimiento para todas las ediciones, nos quedará algo tal que así.

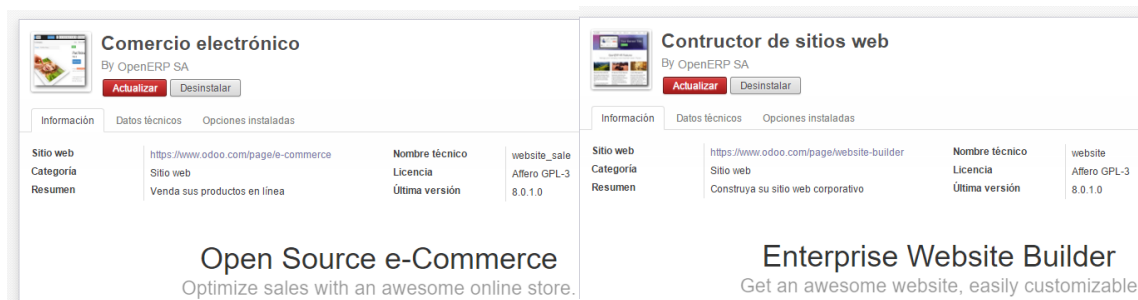
The screenshot shows the 'Ediciones' list view in Odoo. The table has columns: ID, Edición, Fecha_inicio, Fecha_fin, Centro, and ¿Finalizado?. The data is as follows:

| ID | Edición | Fecha_inicio | Fecha_fin | Centro | ¿Finalizado? |
|-------------|----------|--------------|------------|--------------------|--------------------------|
| sie.curso,1 | Edicion1 | 01/05/2016 | 31/05/2017 | Centro Ciudad Real | <input type="checkbox"/> |
| sie.curso,1 | Edicion2 | 05/06/2016 | 04/07/2016 | Centro Miguelterra | <input type="checkbox"/> |
| sie.curso,2 | Edicion1 | 03/05/2016 | 25/05/2017 | Centro Valdepeñas | <input type="checkbox"/> |
| sie.curso,3 | Edicion1 | 04/05/2017 | 03/07/2017 | Centro Ciudad Real | <input type="checkbox"/> |

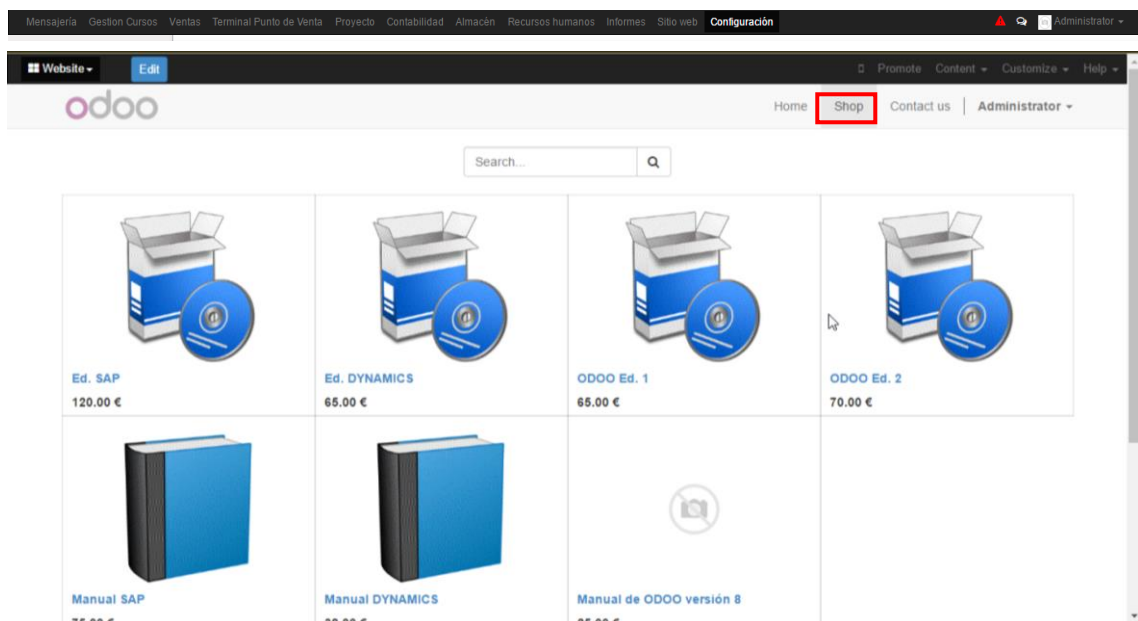
3.- EJERCICIO 3

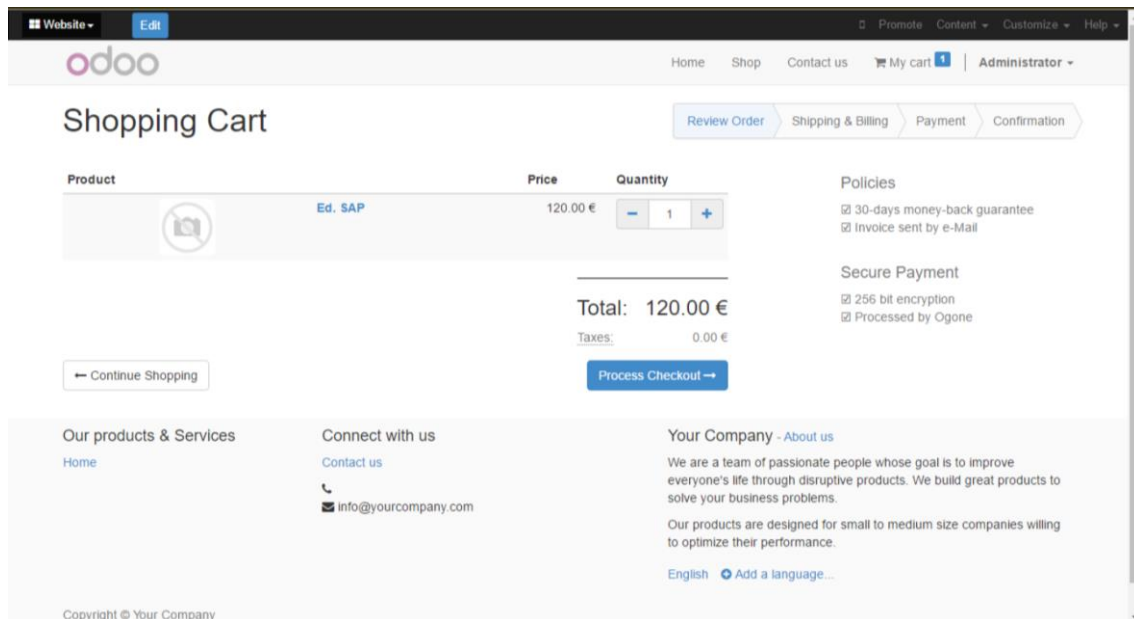
La aplicación debe permitir vender los cursos y otros productos desde el portal on-line de los centros de formación y también en los propios centros de formación.

Para poder realizar este apartado, hemos tenido que instalar dos módulos ya existentes en Odoo, los cuales son **Constructor de Sitios Web**, y **Comercio Electrónico**.



A continuación, cómo en el punto anterior ya habíamos creado los productos, podemos comprobar que se podrá comprar dichos productos en el Sitio Web. Por tanto, pinchamos en "Sitios Web", seguidamente pinchamos en la pestaña "Tienda", y nos mostrará los productos.

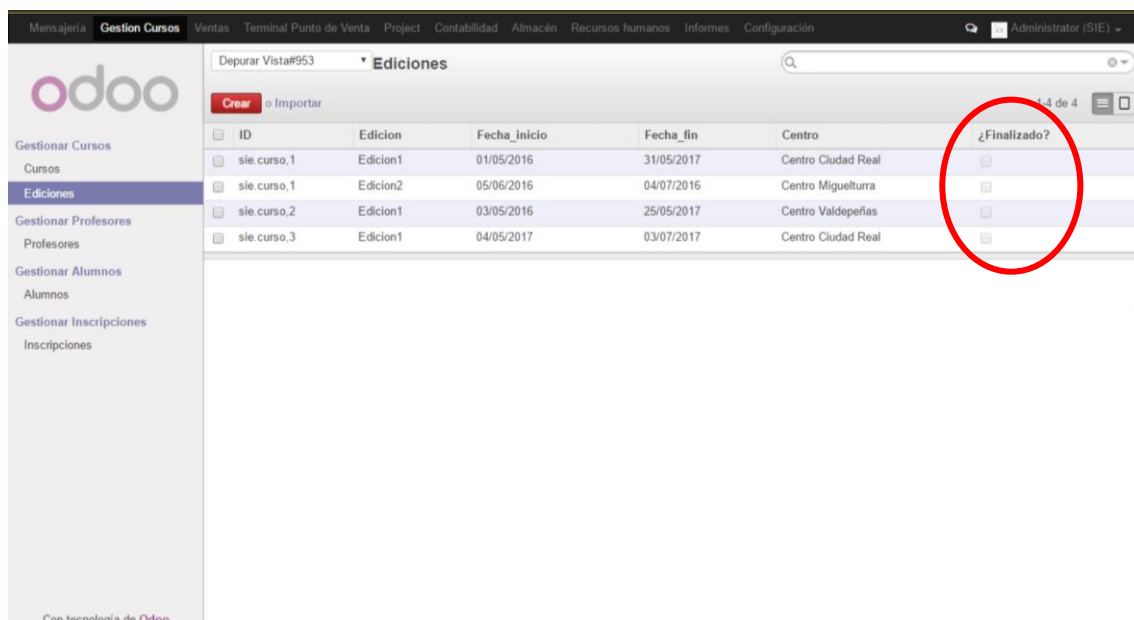




4.- EJERCICIO 4

Una edición de cada curso puede estar en activo o finalizada.

Para este apartado, podemos comprobar en el ejercicio 2 que está ya modelado, no obstante, en la siguiente imagen lo podemos comprobar.



El código utilizado para ello es el siguiente.

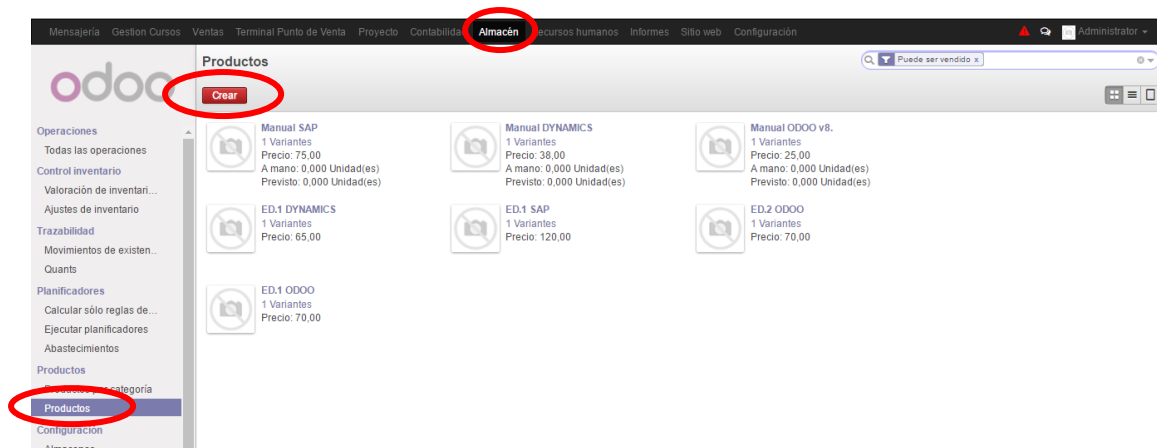
```
22
23 #---CLASE EDICION ---
24 class edicion(osv.osv):
25     _name = "sie.edicion"
26     _description = "Ediciones"
27     _columns = {
28         'curso':fields.many2one('product.template','Curso', ondelete='cascade'),
29         # 'edicion':fields.char('Edicion',size=20,required=True),
30         'edicion':fields.many2one('product.template', 'Edicion', ondelete='set null'),
31         'fecha_inicio':fields.date('Fecha_inicio', size=20),
32         'fecha_fin':fields.date('Fecha_fin', size=64),
33         'centro':fields.many2one('res.company', 'Centro',ondelete='cascade',select=True),
34         'estado':fields.boolean('{Finalizado?}'),
35     }
36
37     _sql_constraints = [
38         ('curso_unique', 'unique(curso,edicion)', 'Curso con edicion ya seleccionada!'),
39     ]
40
41     edicion()
42 #---CLASE PROFESOR ---
```

5.- EJERCICIO 5

Cada centro de formación vende otros productos.

| MANUAL | PRECIO |
|----------|---------|
| ODOO | 25,00 € |
| SAP | 38,00 € |
| DYNAMICS | 75,00 € |

Para este apartado simplemente navegaremos, y pinchamos en la pestaña “Almacén”, “Productos”, “Crear” y a continuación, crearemos los nuevos productos.



Una vez terminado lo anterior, creamos los productos, añadiendo la compañía, precio, tipo de producto, etc.

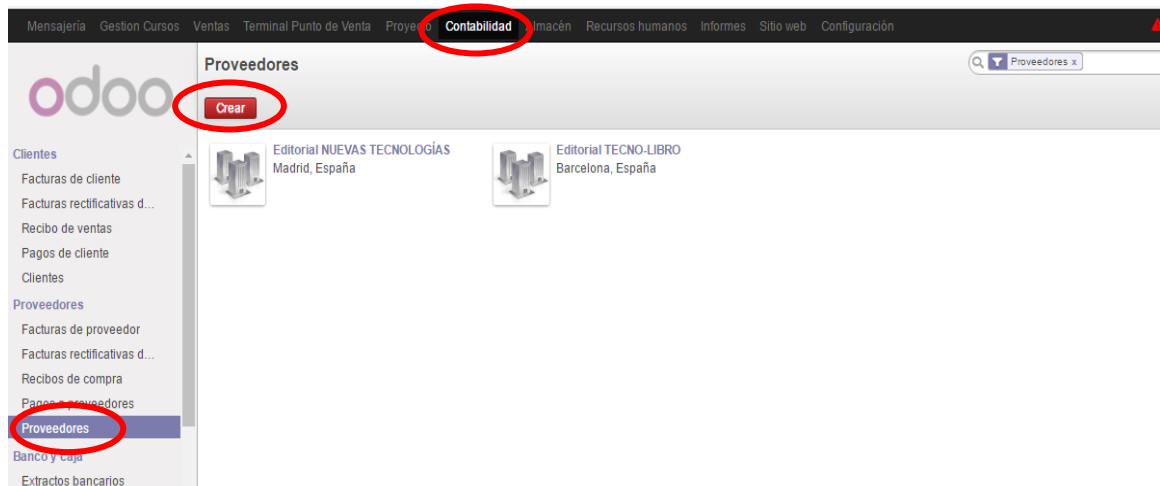
6.- EJERCICIO 6

Se desean registrar también los datos de los proveedores de cada producto.

Editorial TECNO-LIBRO Calle Diagonal 45, Barcelona.

Editorial NUEVAS TECNOLOGÍAS Calle Gran Vía 34, Madrid.

Para este apartado navegamos a la pestaña “Contabilidad”, “Proveedores”, y una vez ahí crearemos los proveedores citados anteriormente.



Una vez introducidos los datos le daremos a guardar y listo.



7.- EJERCICIO 7

Realizar compras de estos productos para tener las existencias que se detalla a continuación.

Manual de ODDO versión 8. Existencia 25 en cada Centro. Proveedor: Editorial TECNO-LIBRO.

Manual Dynamics. Existencia 50 en cada Centro. Proveedor: Editorial NUEVAS TECNOLOGIAS.

Manual SAP. Existencia 15 en cada Centro. Proveedor: Editorial TECNO-LIBRO.

En este apartado, nos desplazaremos hasta la pestaña “Terminal Punto de Venta”, una vez ahí, pinchamos sobre productos, y dentro de cada uno de los citados anteriormente, detallaremos su proveedor, la cantidad, y el centro de destino.

Nombre del producto: **Manual DYNAMICS**

☒ Puede ser vendido
☒ Puede ser comprado

0 Compras, 0 Ventas, 0.0 Disponible, Movimientos

Reglas de reabastecimiento, Rutas

Información | Abastecimientos | Inventario | Ventas | Variantes | Contabilidad

Precio de coste: 0,00

Información de la cadena de suministro

Rutas: ☐ Buy, ☐ Bajo pedido

Proveedores

| Proveedor | Tiempo de entrega | Cantidad mínima | Compañía |
|------------------------------|-------------------|-----------------|--------------------|
| Editorial NUEVAS TECNOLOGÍAS | 1 | 50,00 | Centro Ciudad Real |
| Editorial NUEVAS TECNOLOGÍAS | 1 | 50,00 | Centro Miguelturra |
| Editorial NUEVAS TECNOLOGÍAS | 1 | 50,00 | Centro Valdepeñas |

Añadir un elemento

Este procedimiento lo repetiremos con los tres productos citados en el inicio del apartado.

8.- EJERCICIO 8

Cada edición de un curso tiene un profesor. Y un profesor puede impartir varios cursos.

| PROFESORES. |
|--|
| María Gutiérrez. Toledo 34. CIUDAD REAL. NIF: 86463212W CENTRO CIUDAD REAL (CURSO ODOO. EDICION1, CURSO SAP. EDICION1) |
| Lucía Paredes. Calle Espada 65. MIGUELTURRA NIF: 38375214J (CENTRO MIGUELTURRA CURSO ODOO. EDICION2) |
| Mariano Lucerna. Calle Esperanza 45. VALDEPEÑAS. NIF: 44510279N (CENTRO VALDEPEÑAS, CURSO DYNAMICS. EDICION1) |

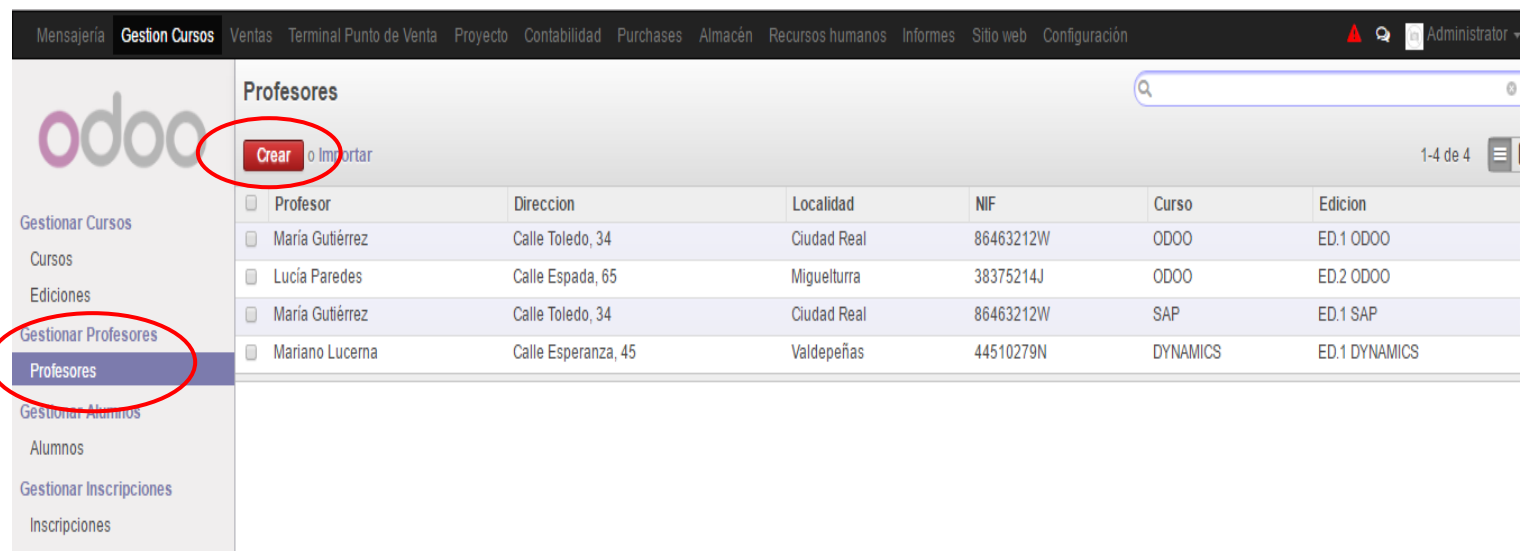
Como ya hicimos en el apartado 2, en este caso seguiremos el mismo procedimiento, añadiendo al código del archivo module.py, el siguiente fragmento de código, para así obtener en nuestro modulo la pestaña profesores. Además, como en casos anteriores, también modificaremos el archivo curso_view.xml, para mantener nuestro sistema organizado.

```

42  #---CLASE PROFESOR ---
43  class profesor(osv.osv):
44      _name = "sie.profesor"
45      _description = "Profesores"
46      _columns = {
47          'nombre':fields.many2one('hr.employee', 'Profesor', ondelete='set null'),
48          'direccion':fields.char('Direccion',size=30,required=True),
49          'localidad':fields.char('Localidad',size=20,required=True),
50          'nif':fields.char('NIF',size=20,required=True),
51          'curso':fields.many2one('product.template','Curso', ondelete='cascade', select=True),
52          'edicion':fields.many2one('product.template','Edicion', ondelete='cascade', select=True),
53      }
54
55      _sql_constraints = [
56          ('edicion_unico','unique(curso,edicion)', 'La edicion solo la imparte un unico profesor'),
57      ]
58  profesor()

```

Con este fragmento de código cumpliremos lo pedido en este apartado, para añadir los profesores de la tabla anterior, junto con sus datos y cursos a impartir. Por tanto, el resultado en el sistema será el siguiente:



The screenshot shows the Odoo 14 web interface. At the top, there's a navigation bar with various modules like 'Mensajería', 'Gestion Cursos', 'Ventas', etc. The main header area displays the 'Profesores' module name and a search bar. Below this, there's a table of professors. The 'Crear' button is circled in red. In the left sidebar, under 'Gestionar Profesores', the 'Profesores' option is also circled in red. The table contains the following data:

| Profesor | Direccion | Localidad | NIF | Curso | Edicion |
|--|---------------------|-------------|-----------|----------|---------------|
| <input type="checkbox"/> María Gutiérrez | Calle Toledo, 34 | Ciudad Real | 86463212W | ODOO | ED.1 ODOO |
| <input type="checkbox"/> Lucía Paredes | Calle Espada, 65 | Miguelturra | 38375214J | ODOO | ED.2 ODOO |
| <input type="checkbox"/> María Gutiérrez | Calle Toledo, 34 | Ciudad Real | 86463212W | SAP | ED.1 SAP |
| <input type="checkbox"/> Mariano Lucerna | Calle Esperanza, 45 | Valdepeñas | 44510279N | DYNAMICS | ED.1 DYNAMICS |

Las creaciones de los profesores de la imagen anterior se hicieron de la siguiente forma:

| | |
|----------------------------|------------------------|
| Guardar o Descartar | |
| Profesor | María <u>Gutiérrez</u> |
| Dirección | Calle Toledo, 34 |
| Localidad | Ciudad Real |
| NIF | 86463212W |
| Curso | ODOO |
| Edición | ED.1 ODOO |

9.- EJERCICIO 9

Cada alumno se puede matricular a varias ediciones de cursos y cada edición de un curso puede tener varios alumnos matriculados.

| ALUMNOS |
|--|
| Marta López, Calle Calatrava 25, Ciudad Real (CURSO ODOO. EDICION1) NIF: 21036111N |
| Mónica Díaz, Calle Toledo, 50, Ciudad Real (CURSO ODOO. EDICION1) NIF: 49277077Z |
| Arturo Ponce, Calle Quijote 25, Valdepeñas (CURSO DYNAMICS. EDICION1) NIF: 74839497G |
| Ana Ruiz, Calle Ancha 65, Miguelturna (CURSO ODOO. EDICION2) NIF: 23540652Z |

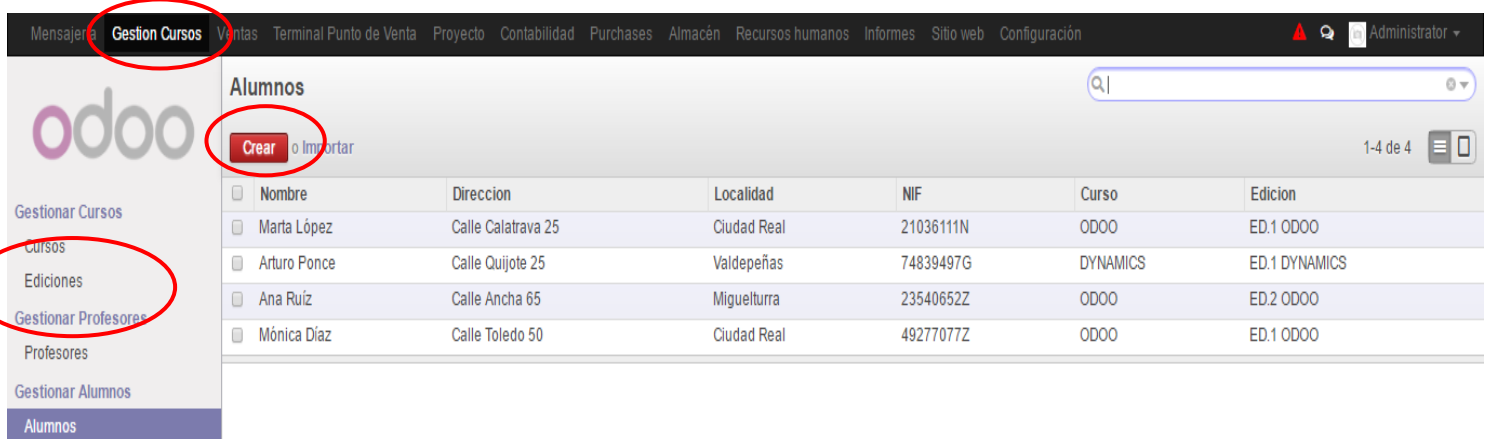
Como ya hicimos en el apartado 2, en este caso seguiremos el mismo procedimiento, añadiendo al código del archivo module.py, el siguiente fragmento de código, para así obtener en nuestro modulo la pestaña alumnos. Además, como en casos anteriores, también modificaremos el archivo curso_view.xml, para mantener nuestro sistema organizado.

```

59
60 class alumnos(osv.osv):
61     _name = "sie.alumno"
62     _description = "Alumnos"
63     _columns = {
64         'nombre':fields.many2one('res.partner', 'Nombre', required=True),
65         'direccion':fields.char('Direccion',size=30,required=True),
66         'localidad':fields.char('Localidad',size=20,required=True),
67         'nif':fields.char('NIF',size=20,required=True),
68         'curso':fields.many2one('product.template','Curso', ondelete='cascade', select=True),
69         'edicion':fields.many2one('product.template','Edicion', ondelete='cascade', select=True),
70     }
71
72 alumnos()
73

```

Con este fragmento de código cumpliremos lo pedido en este apartado, para añadir los profesores de la tabla anterior, junto con sus datos y cursos a impartir. Por tanto, el resultado en el sistema será el siguiente:

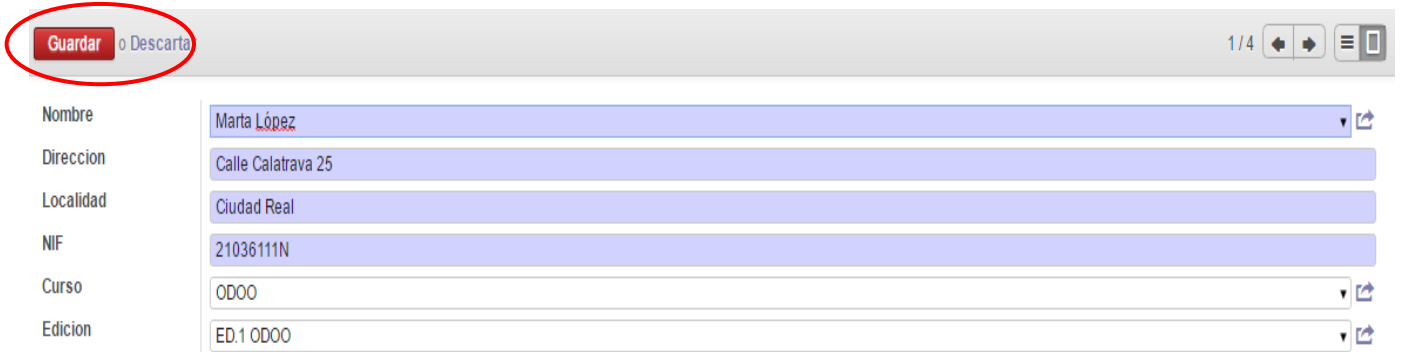


The screenshot shows the Odoo 10 web interface. At the top, the navigation bar includes 'Mensajería', 'Gestion Cursos' (circled in red), 'Ventas', 'Terminal Punto de Venta', 'Proyecto', 'Contabilidad', 'Purchases', 'Almacén', 'Recursos humanos', 'Informes', 'Sitio web', and 'Configuración'. The user is logged in as 'Administrator'. The main content area is titled 'Alumnos' and features a search bar and a 'Crear' button (circled in red). Below the title is a table with the following data:

| Nombre | Direccion | Localidad | NIF | Curso | Edicion |
|--------------|--------------------|-------------|-----------|----------|---------------|
| Marta López | Calle Calatrava 25 | Ciudad Real | 21036111N | ODOO | ED.1 ODOO |
| Arturo Ponce | Calle Quijote 25 | Valdepeñas | 74839497G | DYNAMICS | ED.1 DYNAMICS |
| Ana Ruíz | Calle Ancha 65 | Miguelturra | 23540652Z | ODOO | ED.2 ODOO |
| Mónica Díaz | Calle Toledo 50 | Ciudad Real | 49277077Z | ODOO | ED.1 ODOO |

The left sidebar shows the 'Gestionar Alumnos' menu item circled in red. The main content area also shows a 'Gestionar Alumnos' menu item circled in red.

Para crear los alumnos de la imagen anterior, como hemos mostrado pinchamos en crear, y procedemos a su creación:



The screenshot shows a web form with the following fields:

| Field | Value |
|-----------|--------------------|
| Nombre | Marta López |
| Dirección | Calle Calatrava 25 |
| Localidad | Ciudad Real |
| NIF | 21036111N |
| Curso | OD00 |
| Edición | ED.1 OD00 |

The 'Guardar' button is circled in red.

10.- EJERCICIO 10

Un alumno se matriculará en una edición de un curso a través de un pedido de ventas. Cuando se confirme el pedido de ventas el alumno pasará a estar de manera efectiva matriculado. Un alumno, matriculado en un curso, puede estar en tres estados: pendiente, inscrito, aprobado/suspenso.

En este ejercicio, hemos creado una nueva clase dentro del módulo, llamada inscripción. Para ello hemos implementado el siguiente código Python:

```
74 class inscripcion(osv.osv):
75     _name = "sie.inscripcion"
76     _description = "Inscripciones"
77     _columns = {
78
79         'alumnos': fields.many2one('res.partner', 'Alumnos', ondelete='no action', select=True),
80         'edicion': fields.many2one('sie.edicion', 'Edicion', ondelete='cascade', select=True),
81         'state': fields.selection([('pendiente_inscripcion', 'Pendiente Inscripcion'),
82                                   ('inscrito', 'Inscrito'),
83                                   ('enviarfactura', 'Factura'),
84                                   ('aprobado', 'Aprobado'),
85                                   ('suspenso', 'Suspenso')
86                                   ],
87                                   'State', readonly=True,
88                                   help='Estado de la inscripción.')
89     }
90     _defaults = {
91         'state': lambda *a : 'pendiente_inscripcion'
92     }
93     #----Metodos Estados de Transicion----#
94     def pendiente_inscripcion(self, cr, uid, ids, context=None) :
95         self.write(cr, uid, ids, { 'state' : 'pendiente_inscripcion'}, context=context)
96         return True
97
98     def inscrito(self, cr, uid, ids, context=None) :
99         self.write(cr, uid, ids, { 'state' : 'inscrito'}, context=context)
100        return True
101
102     def enviarfactura(self, cr, uid, ids, context=None) :
103         self.write(cr, uid, ids, { 'state' : 'enviarfactura'}, context=context)
104         return True
105
106     def aprobado(self, cr, uid, ids, context=None) :
107         self.write(cr, uid, ids, { 'state' : 'aprobado'}, context=context)
108         return True
109
110     def suspenso(self, cr, uid, ids, context=None) :
111         self.write(cr, uid, ids, { 'state' : 'suspenso'}, context=context)
112         return True
113
114     inscripcion()
115
```

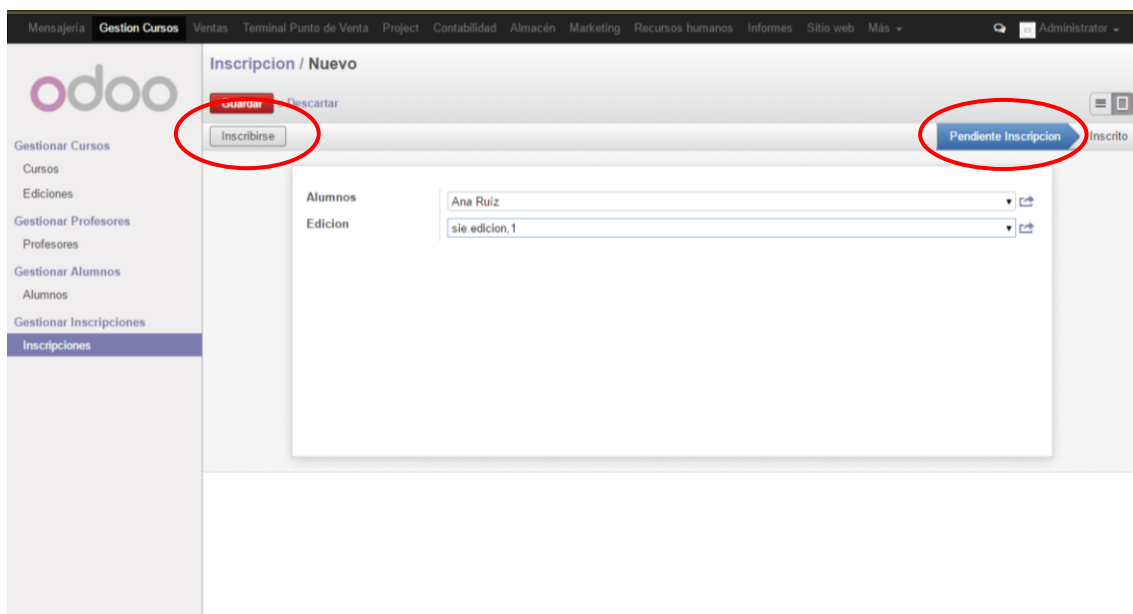
Una vez implementado el código anterior, hemos modificado como en apartados anteriores, el código .XML, añadiendo los siguientes fragmentos:

```

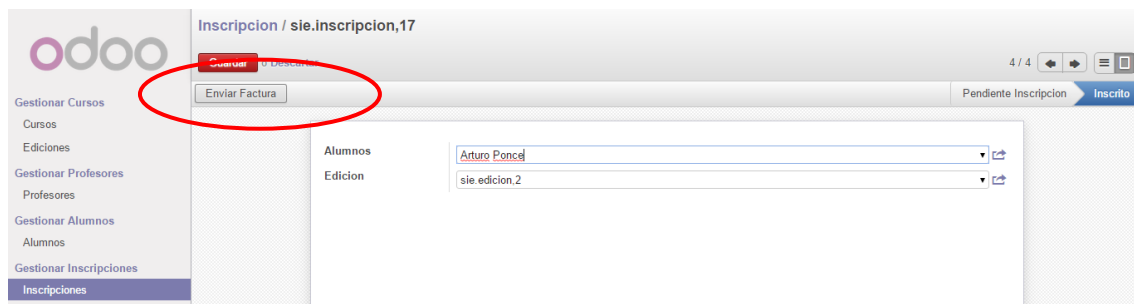
7      <record model="ir.ui.view" id="view_sie_inscripcion_form">
8          <field name="name">Formulario de inscripcion</field>
9          <field name="model">sie.inscripcion</field>
10         <field name="type">form</field>
11         <field name="arch" type="xml">
12             <form version="8.0" string="Inscripcion">
13                 <header>
14                     <button name="inscrito" type="object" string="Inscribirse" states="pendiente_inscripcion"/>
15                     <button name="aprobado" type="object" string="Aprobado" states="enviarfactura" />
16                     <button name="suspense" type="object" string="Suspense" states="enviarfactura" />
17                     <button name="enviarfactura" type="object" string="Enviar Factura" states="inscrito" />
18                 </header>
19                 <field name="state" statusbar_visible="pendiente_inscripcion,inscrito" widget="statusbar"/>
20             </form>
21         </field>
22     </record>
23
24     <record model="ir.ui.view" id="view_sie_inscripcion">
25         <field name="name">Lista de inscripciones</field>
26         <field name="model">sie.inscripcion</field>
27         <field name="type">tree</field>
28         <field name="arch" type="xml">
29             <tree string="sie.inscripcion">
30                 <field name="alumnos"/>
31                 <field name="edicion"/>
32                 <field name="state"/>
33             </tree>
34         </field>
35     </record>

```

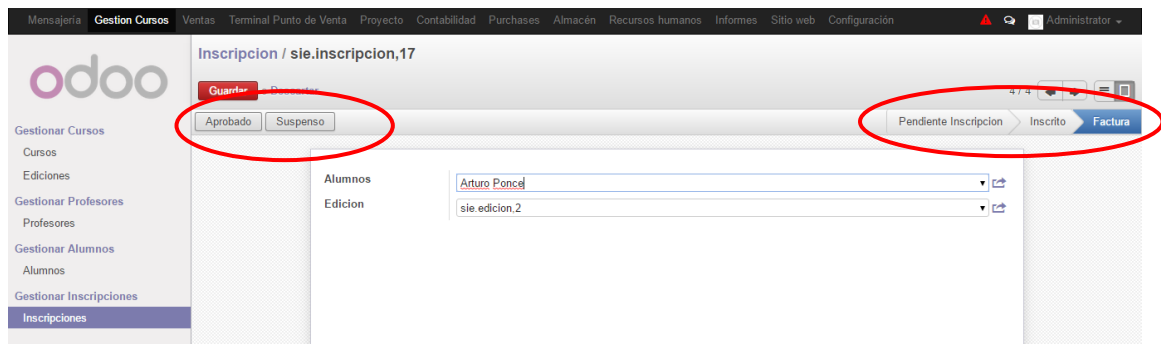
Una vez implementado el código, nos vamos a la interfaz, y seleccionamos uno de los alumnos que hemos creado previamente y una edición. Si guardamos la inscripción sin inscribir al alumno esta quedará como pendiente. Para inscribirlo en la edición del curso, haremos click en el botón que hemos creado en el código de Python y XML.



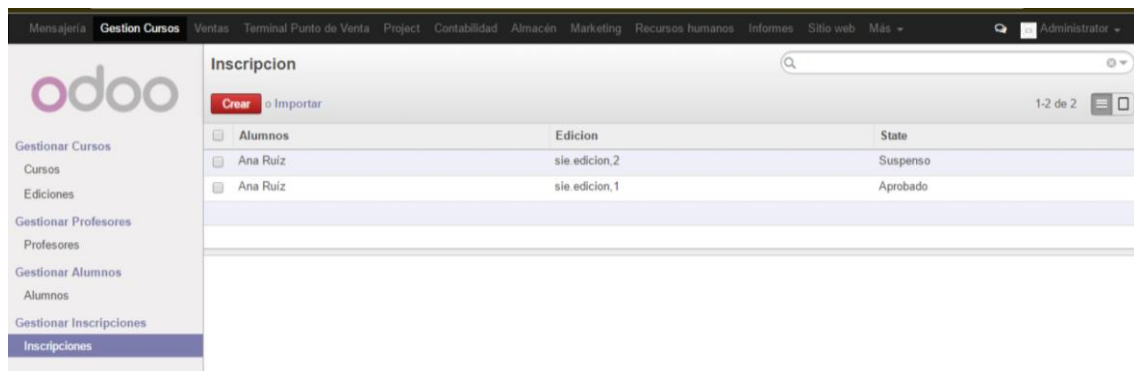
Una vez rellenos los campos podemos inscribir al alumno en la edición. Acto seguido de estar inscrito, nos aparecerá una nueva opción para enviarle la factura al cliente de dicha inscripción.



Cuando enviemos la factura al alumno, nos llevará a una nueva ventana, para poder poner el nuevo estado del alumno, que en nuestro caso podría estar aprobado o suspenso.



De tal manera al finalizar la inscripción quedará así.



11.- EJERCICIO 11

A cada alumno matriculado una vez que está inscrito en el curso, se le debe emitir una factura y se debe realizar la gestión hasta el cobro de la misma.

Para realizar esta tarea, nos dirigimos al *módulo ventas -ventas – pedidos de ventas*.

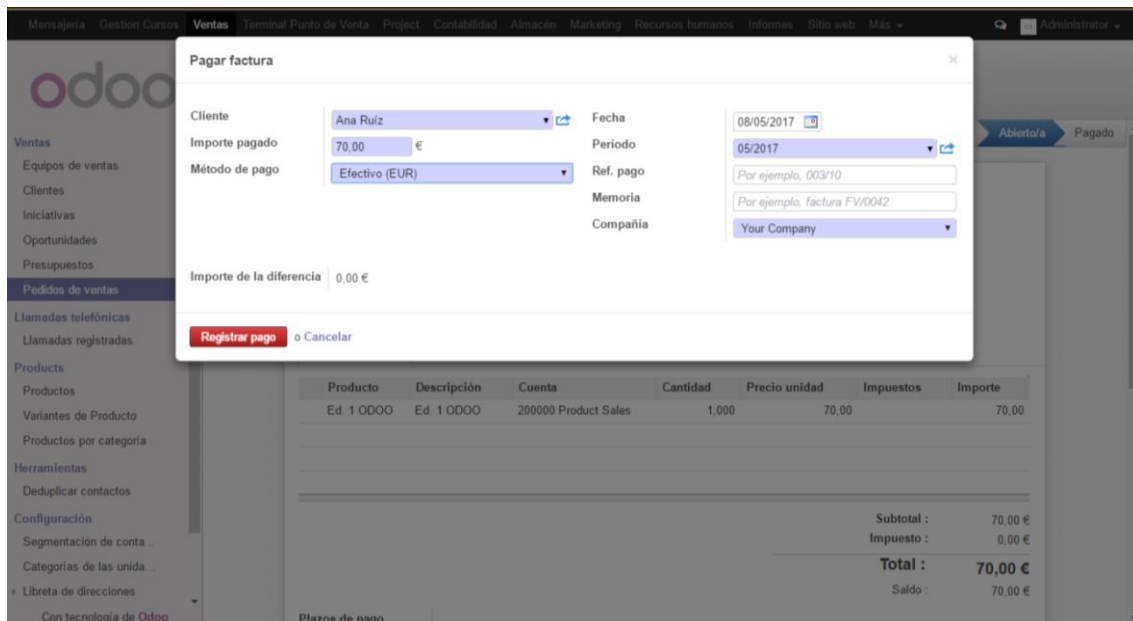
Ahí comenzamos a realizar el pedido.

The screenshot shows the Odoo 'Presupuesto SO006' form. The left sidebar contains a menu with 'Pedidos de ventas' highlighted. The main form area has a header with 'Presupuesto SO006' and a status bar with buttons: 'Presupuesto borrador', 'Presupuesto enviado', 'Pedido de venta', and 'Realizado'. The form fields include 'Cliente' (Ana Ruiz), 'Fecha' (08/05/2017 19:41:29), 'Referencia / Descripción', and 'Almacén' (Almacen Ciudad Real). Below these is a table 'Lineas del pedido' with columns: Producto, Descripción, Cantidad, Precio unidad, Impuestos, and Subtotal. The table contains one row: 'Ed 1 ODOO', 'Ed 1 ODOO', 1,000, 70,00, and 70,00. At the bottom right, the summary shows: 'Base imponible : 70,00 €', 'Impuestos : 0,00 €', and 'Total : (actualizar) 70,00 €'.

Una vez completada la información, hacemos click en **crear factura**

The screenshot shows the Odoo 'Pedido de venta SO006' form. The left sidebar is the same. The main form area has a header with 'Pedido de venta SO006' and a status bar with buttons: 'Presupuesto borrador', 'Presupuesto enviado', 'Pedido de venta', and 'Venta a facturar'. The 'Crear factura' button is circled in red. The form fields include 'Cliente' (Ana Ruiz, Calle Ancha, 13127 Miguelturra, España), 'Fecha' (08/05/2017 19:41:29), 'Referencia / Descripción', and 'Almacén' (Almacen Ciudad Real). Below these is a table 'Lineas del pedido' with columns: Producto, Descripción, Cantidad, Precio unidad, Impuestos, and Subtotal. The table contains one row: 'Ed 1 ODOO', 'Ed 1 ODOO', 1,000, 70,00, and 70,00. At the bottom right, the summary shows: 'Base imponible : 70,00 €', 'Impuestos : 0,00 €', and 'Total : 70,00 €'.

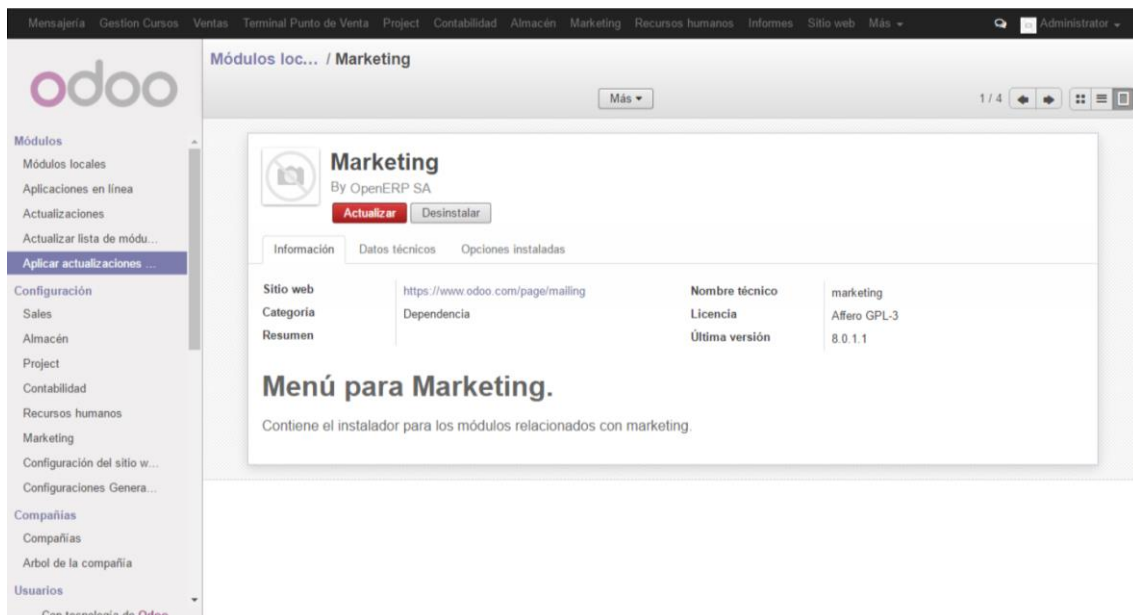
Acto seguido, pagamos la factura y seleccionamos la opción de pago que queramos.



12.- EJERCICIO 12

Montar una campaña masiva de mensajes a todos los alumnos que seleccionen. Para ello añade antes a los alumnos una dirección de correo electrónico.

Lo primero que hemos hecho es instalar el módulo de marketing.



Instalado, procedemos a la creación del correo. Los detalles son los siguientes:

De: admin@example.com

Asunto: Ejercicio12

Para: Customers (Todos los clientes creados previamente)

The screenshot shows the 'Envios masivos' (Mass Emails) form in Odoo. The form is titled 'Envios masivos / Ejercicio12'. It has a sidebar on the left with the following menu items: 'Envio masivo', 'Envios masivos', 'Plantillas de correo', 'Listas de correo', 'Listas de correo', and 'Contactos'. The main form area has a header with 'Guardar' (Save) and 'Descartar' (Discard) buttons. Below the header are buttons for 'Envio de prueba' (Test Send), 'Enviar a todos' (Send to all), 'Borrador' (Draft), 'Probado' (Tested), and 'Enviado' (Sent). The form fields include: 'De' (From) set to 'Administrator <admin@example.com>', 'Asunto' (Subject) set to 'Ejercicio12', and 'Destinatarios' (Recipients) with radio buttons for 'Leads / Opportunities', 'Customers' (selected), and 'Lista de correo' (Email list). Below the recipients, it says '32 records selected' and 'Change selection'. There are tabs for 'Cuerpo del correo' (Email body), 'Opciones' (Options), and 'Seguimiento' (Tracking). The 'Cuerpo del correo' tab is active, showing a 'Diseño de correo' (Email design) button with a blue arrow pointing to it and the text 'Pulse para diseñar su correo.' (Click to design your email.). At the bottom, there is a link to 'Adjuntar un archivo' (Attach a file).

Una vez creado el mail, pasará por los distintos estados; envío de prueba, enviar a todos. De tal manera que cambiará los estados y quedará de la siguiente forma.

The screenshot shows the 'Envios masivos' (Mass Emails) status table in Odoo. The table has a header with 'Crear' (Create) button and a search bar. The table has three columns: 'Draft', 'Tested', and 'Sent'. The 'Draft' column has a count of 0. The 'Tested' column has a count of 0. The 'Sent' column has a count of 1. The 'Sent' column contains a card for 'Ejercicio12' with the date '08/05/2017 10:13:03'. The card shows a progress bar for 'Entregado' (Delivered) at 0%. It also shows 'Abierto 0 %' (Opened 0%) and 'Respondido 0 %' (Responded 0%).

| Draft | Tested | Sent |
|-------|--------|------|
| 0 | 0 | 1 |

Ejercicio12
08/05/2017 10:13:03

Entregado: 0 %

Abierto 0 %

Respondido 0 %