

# iOpenTunes

Curso	Programación de OpenERP/Odoo
Departamento	Organización de Empresas
Profesor	Jose Zambudio Bernabeu
Descripción	2º Ejercicio Realización de un módulo Odoo/OpenERP

## Contenido

- 1 Introducción
- 2 Ejercicios

## 1. Introducción

iOpenTunes es un módulo Odoo/OpenERP orientado a la gestión y venta de obras musicales, ya sean por canción o albums. Para ello iOpenTunes podrá gestionar los autores (cantantes), sus canciones, albums y el número de descargas de estas dos últimas.

## 2. Ejercicios

- i) Realizar un módulo Odoo/OpenERP llamado **iopentunes**.
- ii) La estructura de nuestro módulo será la realizada hasta ahora. (Ver OpenAcademy)
- iii) Añadir el siguiente icono para el módulo anteriormente creado:



- iv) Modelos

### iv)i) **Author**

- Varios autores pueden ser los dueños de una canción.
- Varios autores pueden ser los dueños de un álbum.

Author.py		
<b>name</b>	fields.char	<ul style="list-style-type: none"> <li>● Tamaño: 64</li> <li>● Es necesario rellenar este campo.</li> </ul>
<b>active</b>	boolean	<ul style="list-style-type: none"> <li>● Por defecto: True</li> </ul>
<b>album_ids</b>	many2many	<ul style="list-style-type: none"> <li>● Relación con el model "album"</li> <li>● Relación bidireccional</li> </ul>
<b>song_ids</b>	many2many	<ul style="list-style-type: none"> <li>● Relación con el model "song"</li> <li>● Relación bidireccional</li> </ul>

- iv)i)i. Ya que dos autores no pueden tener el mismo nombre artístico, debemos añadir una restricción para esto.

#### iv)ii) Song

- Una canción podrá tener varios autores.
- Una canción sólo puede estar en un album. En verdad una canción podría estar en varios álbumes, pero vamos a restringir a un único álbum para utilizar todas las relaciones.
- Una canción tendrá siempre la imagen del álbum relacionado.
- Una canción tendrá un único género musical. Damos la posibilidad de no especificar ningún género para la canción.

Song.py		
<b>name</b>	fields.char	<ul style="list-style-type: none"> <li>● Con nombre de campo "Name"</li> <li>● Tamaño: 64</li> <li>● Es necesario rellenar este campo.</li> <li>● Este campo será traducible.</li> </ul>
<b>active</b>	boolean	<ul style="list-style-type: none"> <li>● Con nombre de campo "Active"</li> <li>● Por defecto: True</li> </ul>
<b>author_ids</b>	many2many	<ul style="list-style-type: none"> <li>● Con nombre de campo "Authors"</li> <li>● Relación con el model "author"</li> <li>● Relación bidireccional</li> <li>● Debemos restringir el borrado de autores si este tiene canciones registradas.</li> <li>● Es necesario rellenar este campo.</li> </ul>
<b>album_id</b>	many2one	<ul style="list-style-type: none"> <li>● Con nombre de campo "Album"</li> <li>● Relación con el model "album"</li> </ul>
<b>image</b>	related	<ul style="list-style-type: none"> <li>● Con nombre de campo "Image"</li> <li>● Relación con el model "album"</li> <li>● No debe guardarse en base de datos.</li> </ul>
<b>image_medium</b>	related	<ul style="list-style-type: none"> <li>● Con nombre de campo "Image"</li> <li>● Relación con el model "album"</li> <li>● SI debe guardarse en base de datos.</li> </ul>
<b>image_small</b>	related	<ul style="list-style-type: none"> <li>● Con nombre de campo "Image"</li> <li>● Relación con el model "album"</li> <li>● No debe guardarse en base de datos.</li> </ul>
<b>duration</b>	float	<ul style="list-style-type: none"> <li>● Con nombre de campo "Duration"</li> <li>● Debemos indicar al usuario la siguiente ayuda: "Song duration Minutes : Seconds"</li> </ul>
<b>downloads</b>	integer	<ul style="list-style-type: none"> <li>● Con nombre de campo "Downloads"</li> </ul>
<b>genre_id</b>	many2one	<ul style="list-style-type: none"> <li>● Con nombre de campo "Genre"</li> <li>● Relación con el model "genre"</li> </ul>

**Song.py**

<b>price</b>	float	<ul style="list-style-type: none"> <li>• Con nombre de campo "Price"</li> <li>• 3 enteros 2 decimales</li> </ul>
--------------	-------	--

iv)ii)(i) Debemos ordenar los registros de este modelo por orden de descarga descendente.

**iv)(iii) Album**

- Un álbum tendrá uno o muchos formatos.
- Un álbum tendrá uno o muchos autores.

**Album.py**

<b>name</b>	fields.char	<ul style="list-style-type: none"> <li>• Con nombre de campo "Name"</li> <li>• Tamaño: 256</li> <li>• Es necesario rellenar este campo.</li> <li>• Este campo será traducible.</li> </ul>
<b>active</b>	boolean	<ul style="list-style-type: none"> <li>• Con nombre de campo "Active"</li> <li>• Por defecto: True</li> </ul>
<b>image</b>	binary	<ul style="list-style-type: none"> <li>• Con nombre de campo "Image"</li> <li>• Relación con el model "album"</li> </ul>
<b>image_medium</b>	function	<ul style="list-style-type: none"> <li>• Con nombre de campo "Image"</li> <li>• Nombre de método de obtención de la imagen: <code>_get_image</code></li> <li>• Nombre de método de guardado de la imagen: <code>_set_image</code></li> <li>• Multi: image</li> <li>• Trigger: "image" del modelo Album.</li> <li>• Tamaño por defecto de tools.</li> </ul>
<b>image_small</b>	function	<ul style="list-style-type: none"> <li>• Con nombre de campo "Image"</li> <li>• Nombre de método de obtención de la imagen: <code>_get_image</code></li> <li>• Nombre de método de guardado de la imagen: <code>_set_image</code></li> <li>• Multi: image</li> <li>• Trigger: "image" del modelo Album.</li> <li>• Tamaño por defecto de tools.</li> </ul>
<b>description</b>	text	<ul style="list-style-type: none"> <li>• Con nombre de campo "Description"</li> <li>• Este campo será traducible.</li> </ul>
<b>date_published</b>	date	<ul style="list-style-type: none"> <li>• Con nombre de campo "Published date"</li> <li>• Es necesario rellenar este campo.</li> </ul>
<b>format_ids</b>	many2many	<ul style="list-style-type: none"> <li>• Con nombre de campo "Formats"</li> <li>• Relación con el model "album_format"</li> <li>• Es necesario rellenar este campo.</li> </ul>

Album.py		
<b>author_ids</b>	many2many	<ul style="list-style-type: none"> <li>• Con nombre de campo "Authors"</li> <li>• Relación con el model Author</li> <li>• Relación bidireccional</li> <li>• Es necesario rellenar este campo.</li> <li>• Debemos restringir el borrado de autores si este tiene álbumes registradas.</li> </ul>
<b>song_ids</b>	one2many	<ul style="list-style-type: none"> <li>• Con nombre de campo "Songs"</li> <li>• Relación con el model Song</li> <li>• El campo que hace la inversa es "album_id"</li> <li>• Es necesario rellenar este campo.</li> </ul>
<b>price</b>	float	<ul style="list-style-type: none"> <li>• Con nombre de campo "Price"</li> <li>• 3 enteros 2 decimales</li> <li>• El precio introducido debe ser mayor que 0.</li> </ul>
<b>downloads</b>	integer	<ul style="list-style-type: none"> <li>• Con nombre de campo "Downloads"</li> </ul>

#### iv)iv) Genre

- Los géneros musicales mantienen una estructura jerárquica (ver imagen de árbol de géneros). Para ello debemos indicar que el nombre de nuestro field parent es "parent\_id". El orden que indicamos dentro de dicha jerarquía será por nombre.

Genre.py		
<b>name</b>	fields.char	<ul style="list-style-type: none"> <li>• Con nombre de campo "Name"</li> <li>• Tamaño: 64</li> <li>• Es necesario rellenar este campo.</li> <li>• Este campo será traducible.</li> </ul>
<b>parent_id</b>	many2one	<ul style="list-style-type: none"> <li>• Con nombre de campo "Parent"</li> <li>• Relación con el model Genre</li> <li>• Atributo select a True</li> <li>• Indicar a OpenERP que en caso de borrado del padre, borre todos los hijos.</li> </ul>
<b>child_ids</b>	one2many	<ul style="list-style-type: none"> <li>• Con nombre de campo "Childs Genres"</li> <li>• Relación con el model Genre</li> <li>• El field que hace la inversa es "parent_id"</li> </ul>
<b>parent_left</b>	integer	<ul style="list-style-type: none"> <li>• Con nombre de campo "Left Parent"</li> <li>• Atributo select a 1</li> </ul>
<b>parent_right</b>	integer	<ul style="list-style-type: none"> <li>• Con nombre de campo "Left Parent"</li> <li>• Atributo select a 1</li> </ul>

iv)iv)i. Este modelo debe ordenarse por el field parent\_left.

iv)iv)ii) Se debe sobre-escribir el método `name_get` el cual obtiene por defecto el valor del field `name` de los registros, devolviendo el siguiente formato:

iv)iv)ii)i. En caso de tener un padre: Nombre del padre + / + nombre del registro

iv)iv)ii)i.i) Ejemplo: **Folk / Afrikan**

iv)iv)ii)i.ii) Ejemplo: **Folk / Asian**

iv)iv)ii)ii) En caso de no tener padre: Nombre de registro.

iv)iv)ii)ii)i) Ejemplo: **Folk**

iv)iv)iii) Se debe controlar la existencia de una posible recursión entre los géneros creados. Para no ralentizar el proceso de creación de géneros, vamos a comprobar únicamente 100 niveles.

#### iv)v) Album Format

album_format.py		
<b>name</b>	fields.char	<ul style="list-style-type: none"><li>● Con nombre de campo "Name"</li><li>● Tamaño: 64</li><li>● Es necesario rellenar este campo.</li></ul>
<b>sequence</b>	integer	<ul style="list-style-type: none"><li>● Con nombre de campo "Sequence"</li><li>● Por defecto valor 1</li></ul>

v) **Menuitems** - Se pide realizar la siguiente estructura.

**1. iOpenTunes**

id: iopentunes\_menu\_root

**1.1. iOpenTunes**

id: iopentunes\_menu\_folder

**1.1.1. Authors**

id: iopentunes\_author\_menu

action: iopentunes\_author\_action

**1.2. Media**

id: iopentunes\_media\_menu\_folder

**1.2.1. Albums**

id: iopentunes\_albums\_menu

action: iopentunes\_albums\_action

**1.2.2. Songs**

id: iopentunes\_songs\_menu

action: iopentunes\_songs\_action

**1.3. Configuration**

id: iopentunes\_config\_menu\_folder

**1.3.1. Genders**

id: iopentunes\_genders\_menu

action: iopentunes\_genres\_action

**1.3.2. Formats**

id: iopentunes\_album\_formats\_menu

action: iopentunes\_album\_formats\_action

vi) **Actions:**

vi)i) iopentunes\_author\_action

Tipo vista: form

Vistas: tree, form

Vista inicial: view\_iopentunes\_authors\_tree

Añadir help.

vi)ii) iopentunes\_albums\_action

Tipo vista: form

Vistas: tree,form

Vista inicial: view\_iopentunes\_albums\_tree

Añadir help.

vi)iii) iopentunes\_songs\_action

Tipo vista: form

Vistas: tree,form

**Context: {'group\_by': 'album\_id'}**

Vista inicial: view\_iopentunes\_songs\_tree

Añadir help.

vi)iv) iopentunes\_genres\_action

Tipo vista: **tree**

Vistas: tree,form

Vista inicial: view\_iopentunes\_genres\_tree

Añadir help.

vi)v) iopentunes\_album\_formats\_action

Tipo vista: form

Vistas: **tree**

Vista inicial: view\_iopentunes\_album\_formats\_tree

Añadir help.



## vii) Vistas - Se pide crear las siguientes vistas:

Album_view.xml	ID	Fields
<b>tree</b>	view_iopentunes_albums_tree	<ul style="list-style-type: none"> <li>• Deben aparecer los fields:                             <ul style="list-style-type: none"> <li>• downloads</li> <li>• name</li> <li>• date_published</li> <li>• format_ids widget="many2many_tags"</li> <li>• author_ids widget="many2many_tags"</li> <li>• song_ids widget="one2many_list"</li> <li>• price</li> </ul> </li> </ul>
<b>form</b>	view_iopentunes_albums_form	<ul style="list-style-type: none"> <li>• Deben aparecer los fields:                             <ul style="list-style-type: none"> <li>• image_medium widget="image" class="oe_avatar oe_left"</li> <li>• name</li> <li>• downloads</li> <li>• format_ids</li> <li>• author_ids</li> <li>• song_ids</li> <li>• description</li> <li>• price</li> </ul> </li> </ul>

Album_format_view.xml	ID	Fields
<b>tree</b>	view_iopentunes_album_formats_tree <ul style="list-style-type: none"> <li>• debe tener el atributo editable="bottom"</li> </ul>	<ul style="list-style-type: none"> <li>• Deben aparecer los fields:                             <ul style="list-style-type: none"> <li>• name</li> <li>• sequence widget="handle"</li> </ul> </li> </ul>

Author_view.xml	ID	Fields
<b>tree</b>	view_iopentunes_authors_tree	<ul style="list-style-type: none"> <li>• Deben aparecer los fields:                             <ul style="list-style-type: none"> <li>• name</li> <li>• album_ids widget="many2many_tags"</li> <li>• song_ids widget="one2many_list"</li> </ul> </li> </ul>
<b>form</b>	view_iopentunes_authors_form	<ul style="list-style-type: none"> <li>• Deben aparecer los fields:                             <ul style="list-style-type: none"> <li>• name</li> <li>• active invisible="1"</li> <li>• album_ids</li> <li>• song_ids</li> </ul> </li> </ul>

Genre_view.xml	ID	Fields
<b>tree</b>	view_iopentunes_genres_tree <ul style="list-style-type: none"> <li>• Especificar que el field_parent es "child_ids"</li> <li>• Añadir atributo toolbar=1</li> </ul>	<ul style="list-style-type: none"> <li>• Deben aparecer los fields:                             <ul style="list-style-type: none"> <li>• name</li> <li>• album_ids widget="many2many_tags"</li> <li>• song_ids widget="one2many_list"</li> </ul> </li> </ul>

## PROGRAMACIÓN DE OPENERP/ODOO

Genre_view.xml	ID	Fields
form	view_iopentunes_genres_form	<ul style="list-style-type: none"> <li>• Deben aparecer los fields:                             <ul style="list-style-type: none"> <li>• name</li> <li>• parent_id</li> </ul> </li> </ul>

Song_view.xml	ID	Fields
tree	view_iopentunes_songs_tree	<ul style="list-style-type: none"> <li>• Deben aparecer los fields:                             <ul style="list-style-type: none"> <li>• downloads</li> <li>• name</li> <li>• author_ids widget="many2many_tags"</li> <li>• album_id</li> <li>• duration</li> <li>• gender_id</li> <li>• price</li> </ul> </li> </ul>
form	view_iopentunes_songs_form	<ul style="list-style-type: none"> <li>• Deben aparecer los fields:                             <ul style="list-style-type: none"> <li>• image_medium widget="image" class="oe_avatar oe_left"</li> <li>• downloads</li> <li>• name</li> <li>• active invisible="1"</li> <li>• author_ids widget="many2many_tags"</li> <li>• album_id</li> <li>• duration</li> <li>• gender_id</li> <li>• price</li> </ul> </li> </ul>

- \* Ya que veo que se me hecha el tiempo encima os subo esta primera versión del ejercicio. En caso de ampliarlo o modificar algún error os lo haré saber mediante una noticia en poliformat y una noticia en el grupo de google.
- \* Ya sabéis que si tenéis dudas podéis consultarlas en grupo por el grupo de google. Si alguien todavía no está incluido mandarme un email a [jose.zambudio@diagram.es](mailto:jose.zambudio@diagram.es) y os incluiré en este.