

PRACTICA 11

OBJETIVO: Manejar las acciones o comportamientos

DESCRIPCIÓN:

Hemos creado nuestro nuevo **modelo** y sus correspondientes **derechos de acceso**, es hora de interactuar con la interfaz de usuario.

PASO 1: Archivos de datos (XML)

El formato CSV es conveniente cuando los datos a cargar tienen un formato simple.

Cuando el formato es más complejo (por ejemplo, cargar la estructura de una vista o una plantilla de correo electrónico), usamos el formato XML. Por ejemplo, este [campo de ayuda](#) contiene etiquetas HTML. Si bien sería posible cargar dichos datos a través de un archivo CSV, es más conveniente usar un archivo XML.

Los archivos XML deben agregarse a las mismas carpetas que los archivos CSV y definirse de manera similar en el archivo `_manifest.py`. El contenido de los archivos de datos también se carga secuencialmente cuando se instala o actualiza un módulo, por lo tanto, todas las observaciones hechas para los archivos CSV son válidas para los archivos XML. Cuando los datos están vinculados a las vistas, los agregamos a la `views` carpeta.

En este capítulo cargaremos nuestra primera acción y menús a través de un archivo XML. Las acciones y los menús son registros estándar en la base de datos.

NOTA

Cuando el rendimiento es importante, se prefiere el formato CSV al formato XML. Este es el caso de Odoo, donde cargar un archivo CSV es más rápido que cargar un archivo XML.

- En Odoo, la interfaz de usuario (acciones, menús y vistas) se define en gran medida mediante la creación y composición de registros definidos en un archivo XML.
- Un patrón común es Menú > Acción > Ver. Para acceder a los registros, el usuario navega a través de varios niveles de menú; el nivel más profundo es una acción que desencadena la apertura de una lista de registros.

PASO 2: Comportamiento

Referencia : la documentación relacionada con este tema se encuentra en Acciones .

[Actions — documentación de Odoo - 15.0](#)

INFO `rd-demo odoo.modules.loading: loading estate/views/estate_property_views.xml`

- Las acciones se pueden desencadenar de tres maneras:
 - Haciendo clic en los elementos del menú (vinculados a acciones específicas)
 - Haciendo clic en los botones de las vistas (si están conectados a acciones)
 - Como acciones contextuales sobre el objeto

Ejemplo: de Bienes Raíces, nos gustaría vincular un menú al `estate.property` modelo, para que podamos crear un nuevo registro. La acción se puede ver como el vínculo entre el menú y el modelo.

Una acción básica para nuestro `test.model` es:

```
<record id="test_model_action" model="ir.actions.act_window">
  <field name="name">Test action</field>
  <field name="res_model">test.model</field>
  <field name="view_mode">tree,form</field>
</record>
```

- `id` es un **identificador externo**. Se puede utilizar para hacer referencia al registro (sin conocer su identificador en la base de datos).
- `model` tiene un valor fijo de `ir.actions.act_window` (**Acciones de ventana** (`ir.actions.act_window`)).
- `name` es el nombre de la acción.
- `res_model` es el modelo al que se aplica la acción.
- `view_mode` son las vistas que estarán disponibles; en este caso son las vistas de lista (árbol) y formulario. Veremos más **adelante** que puede haber otros modos de visualización.

Se pueden encontrar ejemplos en todas partes en Odoo, pero este es un buen ejemplo de una acción simple. Preste atención a la estructura del archivo de datos XML ya que la necesitará en el siguiente ejercicio.

Ejercicio

Agrega una acción.

Cree el `estate_property_views.xml` archivo en la carpeta adecuada y defínalo en el `_manifest_.py` archivo.

Cree una acción para el modelo `estate.property`.

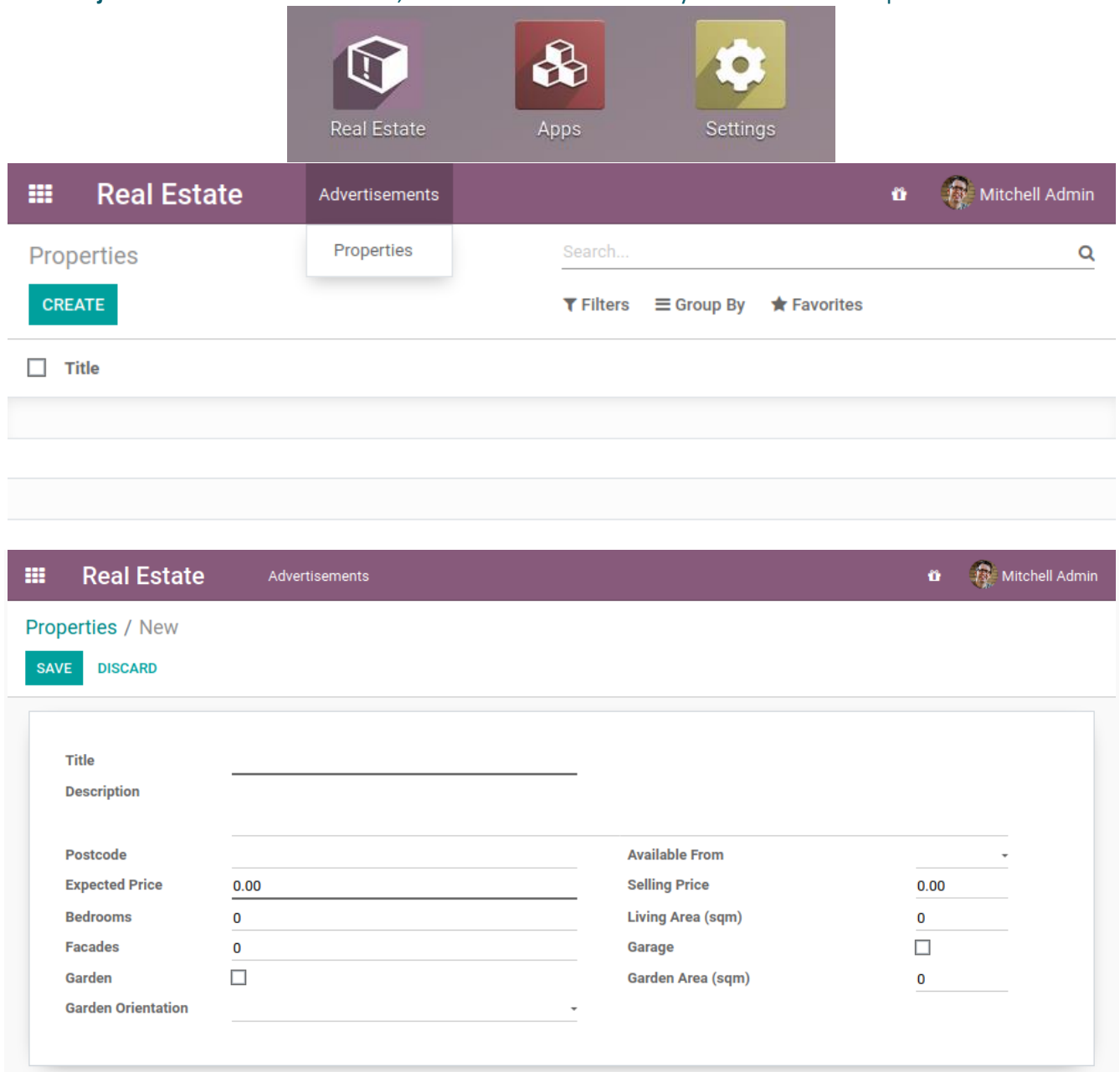
Reinicie el servidor y debería ver el archivo cargado en el registro.

PASO 3: Menús

Referencia : la documentación relacionada con este tema se encuentra en [Accesos directos](#).

NOTA

Objetivo : al final de esta sección, se deben crear tres menús y se muestra la vista predeterminada:



Para reducir la complejidad al declarar un menú (`ir.ui.menu`) y conectarlo a la acción correspondiente, podemos usar el `<menuitem>` atajo .

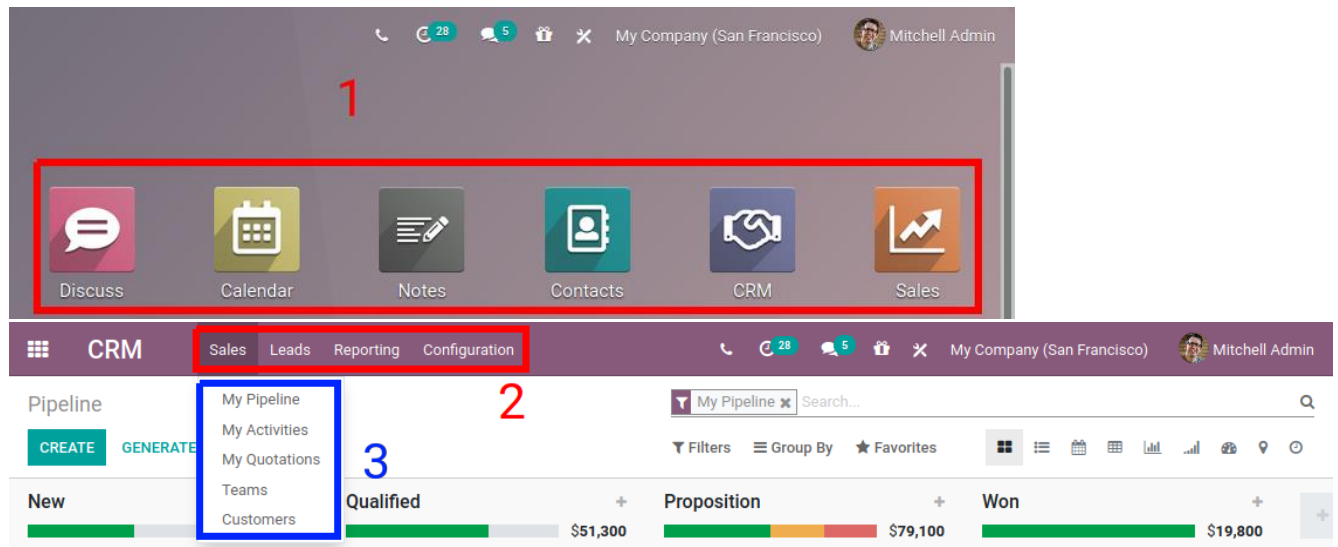
Un menú básico para nuestro `test_model_action` es:

```
<menuitem id="test_model_menu_action" action="test_model_action"/>
```

El menú `test_model_menu_action` está vinculado a la acción `test_model_action` y la acción está vinculada al modelo `test.model`. Como se mencionó anteriormente, la acción puede verse como el vínculo entre el menú y el modelo.

Los menús siempre siguen una arquitectura, y en la práctica existen tres niveles de menús:

1. El menú raíz, que se muestra en el selector de aplicaciones (el selector de aplicaciones de la comunidad de Odoo es un menú desplegable)
2. El menú de primer nivel, que se muestra en la barra superior
3. Los menús de acción



La forma más sencilla de definir la estructura es crearla en el archivo XML. Una estructura básica para nuestro `test_model_action`:

```
<menuitem id="test_menu_root" name="Test">
  <menuitem id="test_first_level_menu" name="First Level">
    <menuitem id="test_model_menu_action" action="test_model_action"/>
  </menuitem>
</menuitem>
```

El nombre del tercer menú se toma del nombre del `action`.

Ejercicio

Añadir menús.

Cree el `estate_menus.xml` archivo en la carpeta adecuada y defínalo en el `__manifest__.py` archivo. Recuerde la carga secuencial de los archivos de datos ;-)

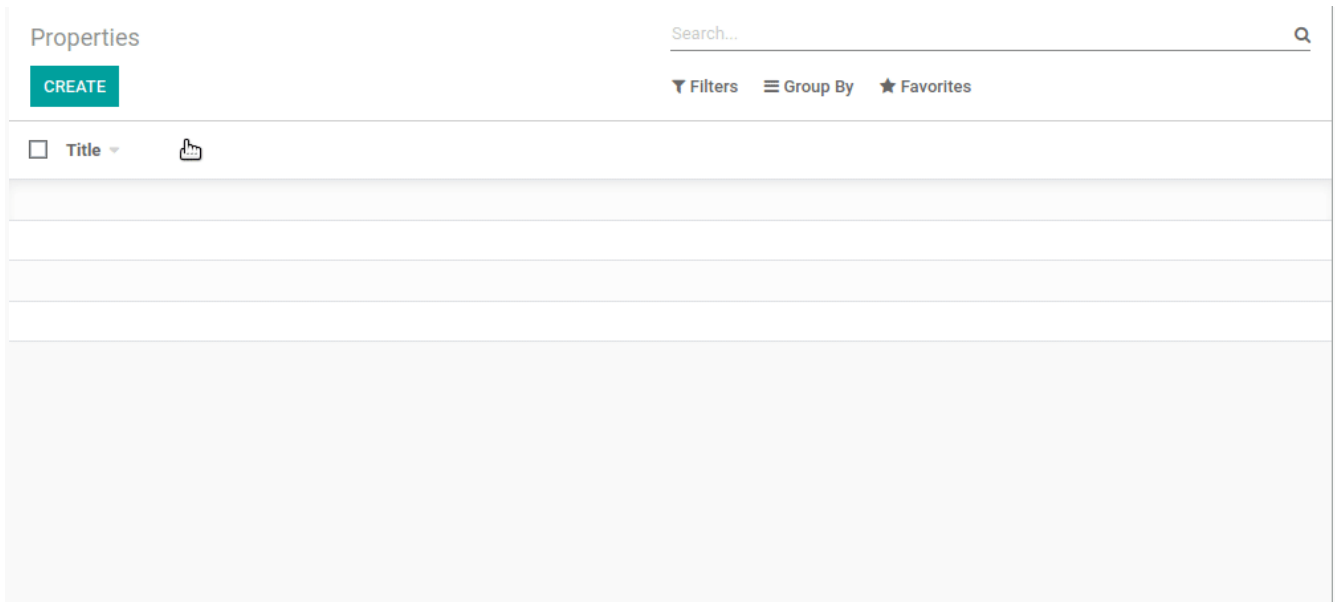
Cree los tres niveles de menús para la `estate.property` acción creada en el ejercicio anterior. Consulte el **objetivo** de esta sección para obtener el resultado esperado.

Reinicie el servidor y **actualice el navegador**. Ahora debería ver los menús, ¡e incluso podrá crear su primer anuncio de propiedad inmobiliaria!

PASO 4: Campos, atributos y vista

NOTA:

Objetivo : al final de esta sección, el precio de venta debe ser de solo lectura y la cantidad de habitaciones y la fecha de disponibilidad deben tener valores predeterminados. Además, los valores de precio de venta y fecha de disponibilidad no se copiarán cuando se duplique el registro.



Los campos reservados `active` y `state` se agregan al `estate.property` modelo.

Hasta ahora, solo hemos utilizado la vista genérica para nuestros anuncios de propiedades inmobiliarias, pero en la mayoría de los casos queremos ajustar la vista. Hay muchos ajustes posibles en Odoo, pero generalmente el primer paso es asegurarse de que:

- algunos campos tienen un valor predeterminado
- algunos campos son de solo lectura
- algunos campos no se copian al duplicar el registro

En nuestro caso de negocio zapatería, nos gustaría lo siguiente:

- El precio de venta debe ser de solo lectura (se completará automáticamente más adelante)
- La fecha de disponibilidad y el precio de venta no deben copiarse al duplicar un registro
- El número predeterminado de habitaciones debe ser 2
- La fecha de disponibilidad predeterminada debe ser en 3 meses

a) Algunos atributos nuevos

Antes de continuar con el diseño de la vista, retrocedamos a la definición de nuestro modelo. Vimos que algunos atributos, como `required=True`, afectan el esquema de la tabla en la base de datos. Otros atributos afectarán la vista o proporcionarán valores predeterminados.

Ejercicio

Agregue nuevos atributos a los campos.

Encuentre los atributos apropiados (ver `Field`) para:

- establecer el precio de venta como de solo lectura
- evitar la copia de la fecha de disponibilidad y los valores del precio de venta

Reinicie el servidor y actualice el navegador. No debería poder establecer ningún precio de venta. Al duplicar un registro, la fecha de disponibilidad debe estar vacía.

b) Valores predeterminados

A cualquier campo se le puede dar un valor predeterminado. En la definición del campo, agregue la opción `default=X` donde `X` es un valor literal de Python (booleano, entero, flotante, cadena) o una función que toma un modelo y devuelve un valor:

```
name = fields.Char(default="Unknown")
```

```
last_seen = fields.Datetime("Last Seen", default=lambda self: fields.Datetime.now())
```

El `name` campo tendrá el valor "Desconocido" de forma predeterminada, mientras que el `last_seen` campo se establecerá como la hora actual.

Ejercicio

Establecer valores predeterminados.

Agregue los atributos predeterminados apropiados para que:

- el número predeterminado de zapatos es 2
- la fecha de disponibilidad predeterminada es en 3 meses

Consejo: esto podría ayudarte: `today()`

Verifique que los valores predeterminados estén configurados como se esperaba.

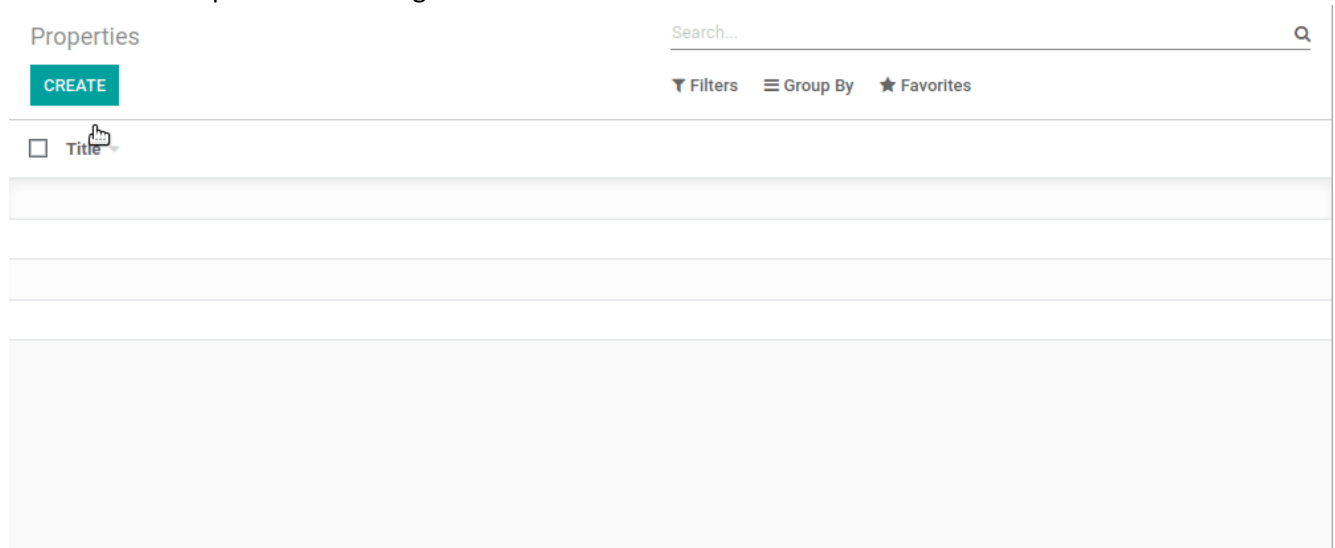
c) Campos reservados

Referencia : la documentación relacionada con este tema se encuentra en Nombres de campos reservados . Algunos nombres de campo están reservados para comportamientos predefinidos. Deben definirse en un modelo cuando se desea el comportamiento relacionado.

Ejercicio: **Añadir campo activo.**

Agregue el `active` campo al `estate.property` modelo.

Reinicie el servidor, cree una nueva propiedad, luego regrese a la vista de lista... ¡La propiedad no aparecerá en la lista! `active` es un ejemplo de un campo reservado con un comportamiento específico: cuando un registro tiene `active=False`, se elimina automáticamente de cualquier búsqueda. Para mostrar la propiedad creada, deberá buscar específicamente registros inactivos.



Ejercicio

Establezca un valor predeterminado para el campo activo.

Establezca el valor predeterminado apropiado para el `active` campo para que no desaparezca más.

Tenga en cuenta que el valor predeterminado `active=False` se asignó a todos los registros existentes.

Ejercicio

Agregar campo de estado.

Agregue un `state` campo al `estate.property` modelo. Hay cinco valores posibles: Nuevo, Oferta recibida, Oferta aceptada, Vendido y Cancelado. Debe ser obligatorio, no debe copiarse y debe tener su valor predeterminado establecido en "Nuevo".

¡Asegúrese de usar el tipo correcto!

Se `state` usará más adelante para varias mejoras de la interfaz de usuario.

Ahora que podemos interactuar con la interfaz de usuario gracias a las vistas predeterminadas, el siguiente paso es obvio: queremos definir **nuestras propias vistas** .

NOTA Se necesita una actualización ya que el cliente web mantiene una memoria caché de los distintos menús y vistas por motivos de rendimiento.

PASO 5: Ejemplo implícito desarrollado en el ejemplo librillo

```
<odoo>
<data>
  <menuitem name="librillo" id="librillo.menu_root"
web_icon="librillo,static/iconos/librillo.png"/>
  <!-- explicit list view definition -->

  <record model="ir.ui.view" id="librillo.list">
    <field name="name">librillo list</field>
    <field name="model">librillo.librillo</field>
    <field name="arch" type="xml">
```

```

        <tree>
            <field name="name"/>
            <field name="value"/>
            <field name="value2"/>
        </tree>
    </field>
</record>

<!-- actions opening views on models -->

<record model="ir.actions.act_window" id="librillo.action_window">
    <field name="name">librillo window</field>
    <field name="res_model">librillo.librillo</field>
    <field name="view_mode">tree,form</field>
</record>

<!-- server action to the one above -->
<record model="ir.actions.server" id="librillo.action_server">
    <field name="name">librillo server</field>
    <field name="model_id" ref="model_librillo_librillo"/>
    <field name="state">code</field>
    <field name="code">
        action = {
            "type": "ir.actions.act_window",
            "view_mode": "tree,form",
            "res_model": model._name,
        }
    </field>
</record>

<!-- Top menu item -->

<menuitem name="librillo" id="librillo.menu_root"/>
<!-- menu categories -->

<menuitem name="Menu 1" id="librillo.menu_1" parent="librillo.menu_root"/>
<menuitem name="Menu 2" id="librillo.menu_2" parent="librillo.menu_root"/>

<!-- actions -->

<menuitem name="List" id="librillo.menu_1_list" parent="librillo.menu_1"
    action="librillo.action_window"/>
<menuitem name="Server to list" id="librillo" parent="librillo.menu_2"
    action="librillo.action_server"/>
</data>
</odoo>

```

PASO 6: Definición de las acciones asociadas a un menú

```

<odoo>
<data>
<record model="ir.ui.view" id="vista_biblioteca_libro_tree">
    <field name="name">vista_biblioteca_libro_tree</field>
    <field name="model">biblioteca.libro</field>
    <field name="arch" type="xml">
        <tree>
            <field name="name" />
            <field name="date_release" />
            <field name="pages" />
            <field name="language" />
        </tree>
    </field>
</record>

```

```

<record model="ir.actions.act_window" id="accion_biblioteca_libro_form">
  <field name="name">Catálogo de Libros</field>
  <field name="type">ir.actions.act_window</field>
  <field name="res_model">biblioteca.libro</field>
  <field name="view_mode">tree,form,kanban</field>
  <field name="help" type="html">
    <p class="oe_view_nocontent_create">
      Libros
    </p>
    <p>
      Haz clic en 'Crear' para añadir un libro.
    </p>
  </field>
</record>

```

```

<record id="vista_biblioteca_libro_kanban" model="ir.ui.view">
  <field name="name">vista_biblioteca_libro_kanban</field>
  <field name="model">biblioteca.libro</field>
  <field name="arch" type="xml">
    <kanban>
      <field name="id"/>
      <field name="name"/>
      <field name="date_release"/>
      <templates>
        <t t-name="kanban-box">
          <div t-attf-class="oe_kanban_global_click">
            <div class="o_kanban_image">
              
            </div>
            <div class="oe_kanban_details">
              <strong class="o_kanban_record_title">
                <field name="name"/>
              </strong>
              <div t-if="record.date_release.value">
                <t t-esc="record.date_release.value"/>
              </div>
            </div>
          </div>
        </t>
      </templates>
    </kanban>
  </field>
</record>

```

```

<record model="ir.actions.act_window" id="accion_biblioteca_libro_form1">
  <field name="name">Catálogo de Libros</field>
  <field name="type">ir.actions.act_window</field>
  <field name="res_model">biblioteca.libro</field>
  <field name="view_mode">list</field>
  <field name="help" type="html">
    <p class="oe_view_content_create">
      Libros2
    </p>
    <p>
      Haz clic en 'Crear' para añadir un libro.
    </p>
  </field>
</record>

```

```

<record id="action_1" model="ir.actions.act_window">
  <field name="name">OPCIÓN MENU 1</field>
  <field name="type">ir.actions.act_window</field>
  <field name="res_model">biblioteca.libro</field>
  <field name="view_mode">form</field>
</record>

<record id="action_2" model="ir.actions.act_window">
  <field name="name">OPCIÓN MENU 2</field>
  <field name="type">ir.actions.act_window</field>
  <field name="res_model">biblioteca.libro</field>
  <field name="view_mode">tree,kanban,form</field>
</record>
<record id="action_3" model="ir.actions.act_window">
  <field name="name">OPCIÓN MENU 3</field>
  <field name="type">ir.actions.act_window</field>
  <field name="res_model">biblioteca.libro</field>
  <field name="view_mode">kanban</field>
<!-- <field name="view_mode">list,form</field> -->
</record>

<record id="action_22" model="ir.actions.act_window">
  <field name="name">OPCIÓN MENU 2 - LIST </field>
  <field name="type">ir.actions.act_window</field>
  <field name="res_model">biblioteca.libro</field>
  <field name="view_mode">list</field>
</record>

<menuitem id="menu_biblioteca_raiz" name="biblioteca" sequence="1"
web_icon="biblioteca,static/description/biblio.png"/>
<menuitem id="menu_biblioteca_catalogo" name="Catálogo de libros" parent="menu_biblioteca_raiz"
sequence="11"/>

<menuitem id="menu_biblioteca_libros" name="Dar de Alta" parent="menu_biblioteca_catalogo" sequence="12"
action="action_1"/>

<menuitem id="menu_biblioteca_libros1" name="Lista" parent="menu_biblioteca_catalogo" sequence="13"
action="action_2"/>
<menuitem id="menu_biblioteca_libros11" name="Consulta kanban" parent="menu_biblioteca_catalogo"
sequence="14" action="action_3"/>
<menuitem id="menu_biblioteca_listados" name="listados" parent="menu_biblioteca_raiz" sequence="20"
action="accion_biblioteca_libro_form"/>

<menuitem id="menu_biblioteca_libros12" name="Consulta Informe1" parent="menu_biblioteca_listados"
sequence="21" action="accion_biblioteca_libro_form1"/>
<menuitem id="menu_biblioteca_libros121" name="Consulta Informe2" parent="menu_biblioteca_listados"
sequence="22" action="action_22"/>
<menuitem id="menu_biblioteca_informacion" name="Informacion" parent="menu_biblioteca_raiz" sequence="30"
/>
<menuitem id="menu_biblioteca_librosINF" name="Consulta Informe1" parent="menu_biblioteca_informacion"
sequence="31" action="accion_biblioteca_libro_form"/>
</data>
</odoo>

```