

PRACTICA 11-a

OBJETIVO: Diferentes tipos de ACCIONES

PASO 1: Acción de ventana (ir.actions.act_window)

El tipo de acción más utilizado, utilizado para mostrar datos del modelo.

Lista de campo:

- **res_model**: el modelo que necesita mostrar datos en la vista
- **views**: una lista de (view_id, view_type), view_type representa el tipo de vista, como formulario, árbol, gráfico ..., view_id es una identificación de base de datos opcional o False, si no Especifique la identificación, el cliente usará automáticamente fields_view_get () para obtener la vista predeterminada del tipo correspondiente, y el primero de la lista de parámetros de tipo se usará para mostrar de manera predeterminada
- **res_id (opcional)**: cuando el tipo de vista predeterminado es formulario, se puede usar para especificar los datos cargados
- **search_view_id (opcional) - (id, nombre)**, id es la vista de búsqueda almacenada en la base de datos, la vista de búsqueda predeterminada del modelo se leerá de forma predeterminada
- **target (opcional)**: defina que la vista se abrirá en la vista actual (actual), use el modo de pantalla completa (pantalla completa), use el cuadro emergente (nuevo), use la principal en lugar de la actual Para limpiar el pan rallado.
- **context (opcional)**: datos de entorno adicionales que deben pasarse a la vista
- **dominio (opcional)**: consulta agregada automáticamente para ver la búsqueda
- **limit (opcional)**: la cantidad de datos que se muestran en el cliente, el valor predeterminado es 80
- **auto_search (opcional)**: si la búsqueda se ejecuta inmediatamente después de cargar la vista predeterminada, el valor predeterminado es Verdadero

PASO 2: La acción de la ventana guardada

La acción de la ventana guardada en la base de datos tiene algunos campos diferentes que el cliente debe ignorar. En la mayoría de los casos, se utiliza para formar la lista de vistas.

1. **view_mode**: una lista separada por comas de tipos de vistas, se mostrarán todos los tipos de vistas
2. **view_ids**: una serie de campos del objeto de vista, que se utiliza para definir el contenido predeterminado de la vista
3. **view_id**: agrega la vista especificada a la vista para evitar que se incluya en view_ids

Los parámetros anteriores se utilizan generalmente al definir acciones con archivos de datos.

```
<record model="ir.actions.act_windows" id="test_action">
  <field name="name"> Acción a testear</field>
  <field name="res_model"> biblioteca.modelo</field>
  <field name="view_mode"> graph</field>
  <field name="view_id" ref="mi_venta_especifica_view"/>
</record>
```

Pasos para combinar vistas en el lado del servidor

1. Obtenga el (id, type) de view_ids a su vez
2. Si view_id está definido y su tipo no está incluido en él, agréguelo al principio
3. Para todos los view_modes que no están especificados, agregue un (False)

PASO 3: Acción de enlace (ir.actions.act_url)

Puede abrir una página de sitio web a través del enlace de odoo, que se puede personalizar a través de dos campos:

1. **url**: el enlace que se abre cuando se activa la acción.
2. **objetivo** - nuevo: abrir en una nueva ventana, auto: reemplazar el contenido de la página actual, predeterminado nuevo.

```
{
  "type": "ir.actions.act_url",
  "url": "http://odoo.com",
  "target": "self",
```

```
}
```

- a) Se puede definir unas acciones directas desde el menú y sino definir unas acciones URL definidas en una función def

```
<record id="accion_buscar" model="ir.actions.act_url">
  <field name="name">Buscador GOOGLE</field>
  <field name="url">https://www.google.es/</field>
  <field name="target">self</field>
</record>

<record id="accion_ayuda_odoo" model="ir.actions.act_url">
  <field name="name">Buscador GOOGLE</field>
  <field name="url">https://www.odoo.com/documentation/15.0/es/</field>
  <field name="target">self</field>
</record>
```

- b) Estas acciones se pueden llamar directamente desde el menú

```
<menuitem id="menu_biblioteca_informacion" name="Informacion" parent="menu_biblioteca_raiz" sequence="30" />
<menuitem id="menu_biblioteca_librosInf" name="buscar en google" parent="menu_biblioteca_informacion" sequence="31" action="accion_buscar"/>
<!--Crear una sección -->
<menuitem id="menu_biblioteca_libros_Sec" name="-----" parent="menu_biblioteca_informacion" sequence="310" />
<menuitem id="menu_biblioteca_Ayuda" name="Pagina ODOO 15" parent="menu_biblioteca_libros_Sec" sequence="32" action="accion_ayuda_odoo"/>
```

PASO 4: Acción del servidor (ir.actions.server)

Puede activar código complejo del lado del servidor mediante el posicionamiento de acciones. Se requieren dos campos relacionados con el cliente:

- **id**: el id de la acción del servidor almacenada en la base de datos.
- **contexto** (opcional): el contexto para ejecutar acciones del lado del servidor.

Las acciones almacenadas en la base de datos pueden realizar algunas acciones especiales según el estado, y algunos campos se comparten entre estados.

- **model_id** - modelo asociado con la acción, disponible en contextos de evaluación.
- **condición** (opcional): use contextos de evaluación del lado del servidor para ejecutar el código Python, si es Falso, bloquee la ejecución de la acción, el valor predeterminado es Verdadero.

El tipo de acción se puede expandir a voluntad, el tipo de acción predeterminado:

- **code**: código Python que se ejecuta cuando se llama a la acción:

```
<record model="ir.actions.server" id="print_instance">
  <field name="name">Acción Server</field>
  <field name="model_id" ref="model_res_partner"/>
  <field name="code">
    Raise Warning(object.name)
  </field>
</record>
```

```
<field name="code">
  If object.same_condition():
    action = {
      "type": "ir.actions.act_window",
      "view_mode": "form",
      "res_model": object._name,
      "res_id": object.id,
    }
</field>
```

- **object_create**: use el gancho para crear un nuevo registro (a través del método de creación o copia) **use_create**
 1. **new**: crea un registro basado en el **model_id** especificado.
 2. **new_other**: crea un registro basado en el **crud_model_id** especificado.
 3. **copy_current**: copia el registro al que hace referencia la acción.
 4. **copy_other**: Copia un registro obtenido a través de **ref_object**.
- **fields_lines**: campos que deben modificarse al crear o copiar registros, One2many tendrá los siguientes campos:
 1. **col1** - **ir.model.fields** contenidos en use_create que necesitan ser reasignados
 2. **value**: el valor correspondiente al campo, analizado según el tipo
 3. **type** - valor: es el valor del campo de valor y la ecuación de valor: el campo de valor se analizará como python
- **crud_model_id**: cuando use_create es new_other, representa la identificación del modelo utilizada para crear un nuevo registro
- **ref_object**: cuando use_create es copy_other, se usa para especificar el registro al que se hace referencia al crear el registro
- **link_new_record** - Ya sea para usar link_field_id para asociar el nuevo registro con el registro actual en many2one, el valor predeterminado es False
- **link_field_id**: especifica el campo de la asociación many2one entre el registro actual y el nuevo registro
- **object_write** - similar a object_create, pero solo modifica el registro actual sin crear un nuevo registro use_create
 1. actualización actual al registro actual.
 2. otro-modificar y actualizar al nuevo registro especificado por crud_model_id o ref_object.
 3. expresión: modificar y actualizar los registros filtrados por crud_model_id y write_expression
write_expression-Devuelve una expresión de Python de identificación de objeto o registro
fields_lines, crud_model_id, ref_object son consistentes con object_create.
- **multi**: ejecutará las acciones definidas por la relación child_ids many2many una por una. Si una acción devuelve la acción en sí, la última acción se devuelve al cliente como la siguiente acción de la acción múltiple anterior.

El disparador envía una señal al flujo de trabajo.

- **wkf_transition_id-used** para activar la identificación que tiene una relación Many2one con workflow.transition
- **use_relational_model**: si es base (predeterminado), activa la señal de mantenimiento del registro actual; si es relacional, activa el campo del registro actual filtrado por wkf_model_id y wkf_field_id
- **client_action**: devuelve la acción definida por **action_id**

PASO 5: Contexto

Algunas claves están disponibles en el contexto y las acciones del servidor:

- **modelo**: el modelo asociado con la acción a través de **model_id**
- **object, obj** - solo disponible cuando **active_model** y **active_id** están disponibles, y se dan los registros filtrados por active_id
- **pool** - registro actual de la base de datos
- **datetime, dateutil, time** - módulo de Python
- **cr** - cursor de consulta actual
- **usuario** - registro de usuario actual
- **contexto** - contexto de ejecución

PASO 6: Acción de informe (ir.actions.report.xml)

Esta acción es el disparador para imprimir el informe.

<https://odoo-master.readthedocs.io/en/master/reference/actions.html>

- **name** (obligatorio): se utiliza al buscar en una lista
- **model** (obligatorio): el modelo de fuente de datos reflejado en el informe

- **tipo_informe** (obligatorio) - **qweb-pdf** | **qweb-html**
- **report_name** - nombre del informe, el nombre del archivo pdf utilizado para la salida
- **groups_id**: puede leer o usar el grupo de usuarios del informe actual, campo Many2many
- **paperformat_id**: el formato de papel utilizado en el informe, el formato de la empresa se utiliza de forma predeterminada, el campo Many2one
- **attach_use**: cuando el valor es verdadero, el informe se generará solo en la primera solicitud y luego se imprimirá directamente desde el informe guardado, que se puede utilizar para el informe que no cambiará después de la generación
- **adjunto**: use una expresión de Python para definir el nombre del informe, se puede acceder al registro con el objeto variable

PASO 7: Acciones del cliente (ir.actions.client)

Activar una acción completamente implementada en el cliente

1. **etiqueta**: el identificador de la acción en el lado del cliente, generalmente una cadena especial
2. **params (opcional)**: datos en formato de diccionario de datos de Python que se utilizan para pasar al cliente
3. **objetivo (opcional) - actual**: acción abierta en el área de contenido actual, pantalla completa: abrir en modo de pantalla completa, nuevo: abrir en cuadro emergente

acción del cliente

```
{
    "type": "ir.actions.client",
    "tag": "pos.ui"
}
```

PASO 8: Acciones del cliente vistas.xml

Analizamos la estructura que se encuentra a continuación respecto a las vistas

```
<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" />
        </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 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">tree,form,kanban</field>
<field name="help" type="html">
    <p class="oe_view_nocontent_create">
        Todos mis libros
    </p>
    <p>
        Haz clic en 'Crear' para añadir un libro.
    </p>
</field>
</record>
<record model="ir.ui.view" id="centromedico.historialclinico_form">
    <field name="name">centromedico.historialclinico.form</field>
    <field name="model">centromedico.historialclinico</field>
    <field name="arch" type="xml">
        <form>
            <group col="2" colspan="2">
                <field name="phistorial" style='width: 500px;'>
                    <tree>
                        <field name="name"/>
                    </tree>
                </field>
            </group>
            <group col="2" colspan="2">
                <field name="diagnostico" style='max-width: 1150px; width:
500px;'/>
                <field name="tratamiento" style='max-width: 1150px; width:
500px;'/>
                <field name="evolucion" style='max-width: 1150px; width:
500px;'/>
            </group>
        </form>
    </field>
</record>

<record model="ir.actions.act_window"
id="centromedico.historialclinico_action_window">
    <field name="name">centromedico.historialclinico_window</field>
    <field name="res_model">centromedico.historialclinico</field>
    <field name="view_mode">tree,form</field>
</record>

<menuitem name="Historial clínico" id="centromedico.historialclinico_menu"
parent="" action="centromedico.historialclinico_action_window"
groups="centromedico.grupo_doctores"/>

    <menuitem id="menu_biblioteca_raiz" name="Biblioteca" sequence="10"/>
    <menuitem id="menu_biblioteca_catalogo" name="Catálogo de libros"
parent="menu_biblioteca_raiz" sequence="0"/>

    <menuitem id="menu_biblioteca_libros" name="Consultar"
parent="menu_biblioteca_catalogo" sequence="0"
action="accion_biblioteca_libro_form"/>

    <menuitem id="menu_biblioteca_libros1" name="Consulta Arbolr"
parent="menu_biblioteca_catalogo" sequence="1"
action="accion_biblioteca_libro_form"/>
    <menuitem id="menu_biblioteca_libros11" name="Consulta lista"
parent="menu_biblioteca_catalogo" sequence="2"
action="accion_biblioteca_libro_form1"/>
    <menuitem id="menu_biblioteca_listados" name="listados"
parent="menu_biblioteca_raiz" sequence="2" action="book_form_view"/>

    <menuitem id="menu_biblioteca_libros12" name="Consulta Informe1"

```

```

parent="menu_biblioteca_listados" sequence="11"
action="accion_biblioteca_libro_form"/>
    <menuitem id="menu_biblioteca_libros121" name="Consulta Informe2"
parent="menu_biblioteca_listados" sequence="12"
action="accion_biblioteca_libro_form"/>
    <menuitem id="menu_biblioteca_informacion" name="Informacion"
parent="menu_biblioteca_raiz" sequence="3" />
    <menuitem id="menu_biblioteca_librosINf" name="Consulta Informe1"
parent="menu_biblioteca_informacion" sequence="31"
action="accion_biblioteca_libro_form"/>
</data>
</odoo>

```

PASO 9: Análisis

Analizar la vista a)

```

<record id="view_form_todo_task_ui" model="ir.ui.view">
    <field name="name">view_form_todo_task_ui</field>
    <field name="model">todo.task</field>
    <field name="arch" type="xml">
        <form>
            <header><!-- Buttons and status widget --> </header>
            <sheet><!-- Form content --> </sheet>
            <!-- History and communication: -->
            <div class="oe_chatter">
                <field name="message_follower_ids" widget="mail_followers" />
                <field name="message_ids" widget="mail_thread" />
            </div>
        </form>
    </field>
</record>

```

Referencia Web

<https://groups.google.com/g/openerp-spain-users/c/7IM-tIDPYQA>

<https://programmerclick.com/article/44171606191/>