

PRACTICA 12

OBJETIVO: GENERAR INFORMES

DESCRIPCIÓN:

Los informes son una parte esencial en la administración y gestión de cualquier empresa los informes se corresponden con las facturas, los listados de clientes, las descripciones de artículos,...

Las aplicaciones comerciales, permiten generar estos informes para el correcto funcionamiento de la empresa.

Los informes pueden tener soporte en dos tipos de formato en formato digital, un formato fichero entendible por un ordenador como puede ser un fichero Word, PDF, o en formato papel, en un trozo de papel físico.

Normalmente, los informes se generan en formato digital lo que permite, a su vez, imprimirlo cuando sea necesario.

En Odoo, un informe no es más que una vista, por lo que se generará de forma parecida a las vistas ya explicadas, estos es, mediante ficheros XML.

Sobre estos ficheros XML se utilizan plantillas QWeb para colocar los diferentes datos extraídos de los modelos y mediante etiquetas HTML y CSS se les añade texto adicional, forma y estilo al informe final

El formato digital escogido por Odoo para los informes es, normalmente, PDF, y la transformación de la vista de XML al formato PDF es realizada por wkhtmltopdf wkhtmltopdf

es una herramienta open source (LGPLv 3 que permite renderizar HTML en PDF o en varios formatos de imagen utilizando el motor de renderizado Qt WebKit

Para generar el informe se debe vincular a una acción, esto es, habrá en nuestro módulo de Odoo algún botón que, al hacer clic sobre él, dispare una acción que mediante el uso de una plantilla permita generarlo

Como la generación de informes es una tarea que se realiza de manera habitual, existe ya definida una acción concreta para crear informes, la cual facilita mucho el trabajo al programador, esta acción puede utilizarse directamente o, si se desea, puede crearse una acción general y personalizarse

De igual modo, también está disponible un modelo de plantilla para definir el cuerpo del informe, aunque puede crearse desde cero una nueva plantilla para informes personalizados, que permita, por ejemplo, expandir la funcionalidad, dando acceso a más modelos,

En resumen, los pasos para crear un informe, utilizando las facilidades que brinda Odoo, son

- Se utiliza el elemento predefinido en Odoo llamado report
- Se define una nueva plantilla, mediante el uso de una plantilla ya predefinida en Odoo, que generará el contenido y la forma del informe

PASO 1: REPORT

Cada informe que se desee generar debe ser definido por una acción de informe:

ir.actions.report

Pero para simplificar la tarea de generar informes, como ya se ha comentado previamente, existe un acceso sencillo a través de un elemento llamado report en lugar de tener que configurar manualmente toda la acción y su entorno

El elemento report tiene los siguientes atributos:

- **string** establece el texto que aparecerá en el botón para poder generar el informe
- **id** establece un identificador externo asociado al report
- **model** (obligatorio) el nombre del modelo que contiene los datos que se usarán para generar el informe.
- **name** define el id externo de la plantilla que se usará para generar el informe, con la forma module_name.template_id por ejemplo biblioteca. report_ficha_libro
- **report_type** (por defecto qweb pdf con el valor qweb pdf se generan informes en PDF y con el valor qweb html informes en formato HTML
- **print_report_name** contiene una expresión de Python que define el nombre del fichero PDF
- **attachment_use** si se establece con valor True el informe se almacenará como un archivo adjunto del registro, utilizando el nombre generado por la expresión attachment Este campo permite generar un informe una única vez, lo que puede ser útil si motivos legales, como por ejemplo, la firma de un contrato, un comprobante de pago,
- **attachment** contiene una expresión de Python que define el nombre del informe El registro, junto con los datos del mismo, es accesible a través de la variable object

- **paperformat** id externo del formato de papel específico que se desea usar para el informe Si no se especifica, se utiliza el formato predeterminado de papel de la empresa

a) Ejemplo

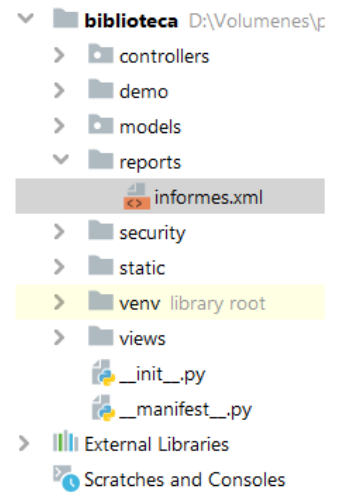
Continuando con el ejemplo de los temas anteriores, vamos a añadirle funcionalidad para generar informes.

En concreto, añadiremos una ficha en formato PDF y en formato HTML, la utilizaremos como una previsualización, para cada libro contenido en nuestro modelo biblioteca libro.

Para ello utilizaremos la facilidad para generar el informe que aporta Odoo a través del elemento report

Lo primero que hacemos es añadir una nueva carpeta dentro de la estructura de nuestro módulo biblioteca, llamada reports con el fin de tener todos nuestros ficheros ordenados Esta carpeta contendrá los diferentes ficheros que permiten generar los informes.

Dentro de esta carpeta creamos el fichero informes.xml y a continuación lo editamos.



b) Fichero informes.xml

Añadimos al fichero informes.xml, el siguiente contenido:

```
<odoo>
  <data>
    <report string="Imprimir la ficha del libro"
      id="biblioteca_libro_report"
      model="biblioteca.libro"
      report_type="qweb-pdf"
      name="biblioteca.report_ficha_libro"
      print_report_name="(object.name)"/>
  </data>
</odoo>
```

Este código hemos añadido la acción que permite al hacer clic en “Imprimir la ficha del libro”:

string="Imprimir la ficha del libro"

Generar un fichero PDF:

report_type="qweb-pdf"

Llamado con el nombre del libro:

print_report_name="(object.name)"

Los datos los sacamos del modelo:

Model="biblioteca.libro"

Y se usará la plantilla:

name="biblioteca.report_ficha_libro"

c)

También añadimos también al fichero informes.xml, a continuación de la entrada report anterior y entre las etiquetas el siguiente contenido:

```
<report string="Previsualizar la ficha del libro"
  id="biblioteca_libro_report_preview"
  model="biblioteca.libro"
  report_type="qweb-html"
  name="biblioteca.report_ficha_libro"
  print_report_name="(object.name)"
/>
```

Con este código hemos añadido la acción que permite generar el mismo informe pero en formato HTML al hacer clic en “Previsualizar la ficha del libro”: string="Previsualizar la ficha del libro"

Generar un fichero HTML: report_type="qweb-html"

Llamado con el nombre del libro: print_report_name="(object.name)"

Los datos los sacamos del modelo: model="biblioteca.libro" Y se usará la misma plantilla:

name="biblioteca.report_ficha_libro"

Como puedes observar, en el segundo informe vamos a utilizar la misma plantilla que en el primero, `biblioteca.report_ficha_libro`, al fin y al cabo lo que queremos es mostrar la misma disposición en un formato PDF y en un formato HTML.

c) Fichero

Para hacer uso del fichero que hemos creado, debemos modificar el fichero `__manifest__.py` ubicado en la carpeta que contiene el módulo de Odoo. Añadimos a la lista `data`, el elemento `'reports/informes.xml'`, que es la ruta al fichero que define los informes. Quedando de la siguiente forma:

```
'data': [  
    'security/grupos.xml',  
    'security/ir.model.access.csv',  
    'views/vistas.xml',  
    'views/views.xml',  
    'views/templates.xml',  
    'demo/demo.xml',  
    'reports/informes.xml'  
],
```

A continuación, hay que crear esta plantilla `report_ficha_libro`.

PASO 2: PLANTILLAS DE INFORMES

Normalmente, las plantillas actúan como pseudo-documentos, como trozos que al procesarse generan las diferentes partes o secciones de los documentos o informes.

La ejecución del parámetro `scaffold` del ejecutable de Odoo crea en el directorio `views` un fichero llamado `templates.xml` para contener las diferentes plantillas que usa un módulo de Odoo.

El contenido de este fichero es idéntico se define mediante XML, de forma parecida a como se definen las vistas.

Las plantillas de informes son en esencia plantillas (templates).

Como se ha comentado previamente, en Odoo hay disponible una plantilla por defecto, llamada `external_layout`, que agrega el encabezado y el pie de página predeterminados en un informe. El cuerpo del informe, el cuerpo PDF, debe estar contenido dentro de unas etiquetas

. Además, la plantilla que se usará para generar el informe, se selecciona mediante su `id` y debe coincidir con el atributo `name` especificado en la declaración del informe. Por ejemplo, si queremos usar una plantilla que hemos creado llamada:

```
<template id="report_fecha_libro">
```

En la declaración del informe debemos indicarlo mediante el atributo `name`:

```
<report  
...  
name="biblioteca.report_ficha_libro"
```

Por defecto, el elemento `report` es sencillo y solo aporta registros de un único modelo a la plantilla del informe.

Si se desea acceder a otros registros de otros modelos dentro de la misma plantilla se necesita crear un informe personalizado mediante la etiqueta `record`.

Los elementos disponibles, a través del contexto de renderización, en la plantilla del informe son los siguientes:

- **docs** todos los registros que se desean imprimir en el informe.
- **doc_ids** lista de los `ids` de los registros de `docs` o, dicho de otra forma, todos los `ids` de los registros que se desean imprimir en el informe.
- **doc_model** el modelo de los registros de `docs` o, dicho de otra forma, el modelo que usan los registros que se desean imprimir en el informe.

Vamos a crear la plantilla con el `id` externo `report_ficha_libro` que utilizan nuestros informes, utiliza este `id` para poderse llamar desde las entradas donde añadimos el atributo `name="biblioteca.report_ficha_libro"`, observa como se antepone el nombre del módulo al de la plantilla para que funcione correctamente. Vamos al fichero `views/templates.xml` y lo editamos, añadiendo las siguientes líneas que definen la plantilla:

```
<template id="report_ficha_libro">
</template>
```

Ahora incluiremos, entre las etiquetas , el cuerpo de la plantilla que generará el documento.
Lo primero es utilizar una plantilla de Odoo para evitar tener que crear el cuerpo externo desde cero.

La plantilla se llama web.html_container, y crea un cuerpo básico (html, head, body...) de una pagina HTML, así que la llamamos:

```
<template id="report_libro_card">
    <t t-call="web.html_container">
    </t>
</template>
```

A continuación, le indicamos qué es lo que tiene que hacer con cada registro del modelo libro.biblioteca que le enviemos a la plantilla, que se corresponderá con una ficha, aunque solo le enviaremos uno, esto lo hacemos mediante un bucle: , docs representa un array con los registros enviados.

Lo primero será definir para cada ficha, para cada documento docs, una disposición independiente, pero en vez de hacer nosotros dicha ficha volvemos a utilizar otra plantilla de Odoo ya creada para ello llamada web.external_layout.

d) Estructura de la plantilla

La estructura de la plantilla report_ficha_libro debe tener por ahora esta forma:

```
<template id="report_ficha_libro">
    <t t-call="web.html_container">
        <t t-foreach="docs" t-as="o">
            <t t-call="web.external_layout">
            </t>
        </t>
    </t>
</template>
```

Y como queremos que cada ficha de libro salga en una página distinta añadimos <div class="page"></div> que hará que cuando se genere el PDF fuerce esta disposición.

A continuación, entre las etiquetas <div class="page"></div> añadimos los datos que queremos que se muestren

Como utilizamos un bucle para recorrer todos los registros de esta forma <t t-foreach="docs" t-as="o"></t>, se accede a cada uno de ellos como o, así para mostrar el campo lenguaje, definido en el modelo biblioteca.libro como language, sería:

```
<span t-field="o.language" />
```

Como no sabemos si los campos del modelo biblioteca.libro tienen valores o no, ya que no son obligatorio, podemos hacer una comprobación previa antes de mostrarlos:

```
<t t-if="o.language">
    <span t-field="o.language" />
</t>
```

Por último, le añadimos un estilo CSS para que tenga un aspecto mejor con un párrafo HTML y un texto indicando qué es lo que se muestra:

```
<p><span style="color: black; font-family: Verdana; font-weight: bold;">Idioma:</span>.</p>
```

La portada del libro, que se almacena con el nombre del campo imagen del modelo biblioteca.libro, la vamos a mostrar con la etiqueta de HTML, por lo que debemos colocarla en el atributo src de dicha etiqueta. Sin embargo, en lugar de proporcionar una URL o la ruta a un fichero que contiene la imagen, no disponemos de tal cosa, le decimos que los datos están en línea, a continuación, que es una imagen png y lo hacemos mediante la sentencia data:image/png;base64 (para más detalles, ver la RFC 2397)

seguida de los datos binarios de la imagen, que previamente hemos formateado como una cadena de caracteres mediante la llamada a la función `to_text()` de Python (para más información accede a https://www.kite.com/python/docs/ansible.cli.to_text).

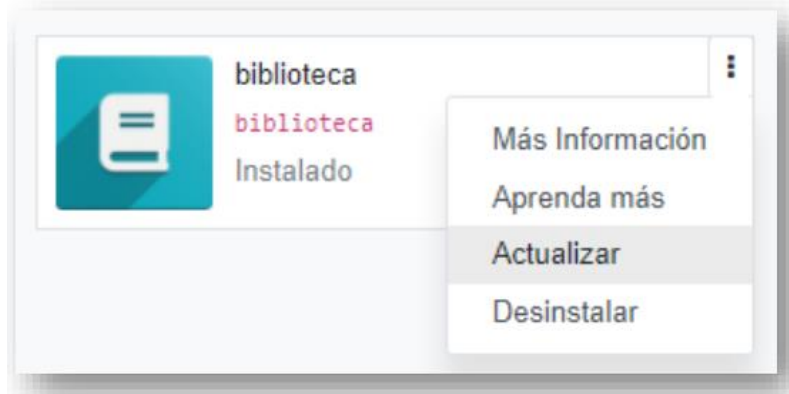
Al final, con estilo CSS incluido, la etiqueta `` queda de la siguiente manera:

```

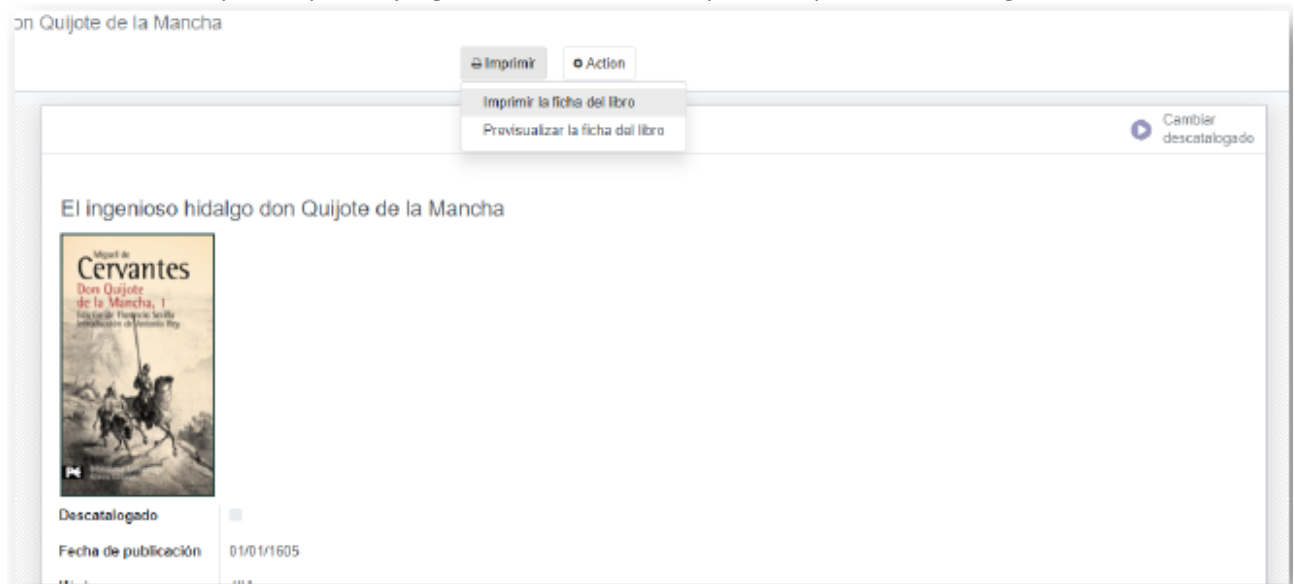
```

El resultado final de la plantilla `report_ficha_libro`:

Como hemos hecho cambios en el modulo biblioteca, y el modulo ya estaba instalado, hay que actualizarlo en Odoo para que se incorporen las mejoras que hemos hecho. Para ello en la página Aplicaciones, Actualizamos la lista de aplicaciones, buscamos el módulo, y en el menú del módulo escogemos actualizar:



Nos vamos a la vista formulario de cualquier libro y podremos observar que junto al botón Action, tenemos ahora el botón Imprimir que desplegará un menú con las opciones que hemos configurado:



Haciendo clic en cada una de ellas obtendremos la ficha del libro en PDF o su previsualización en HTML:

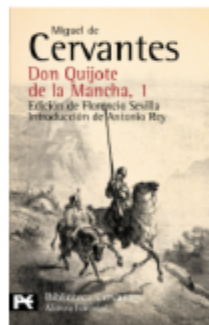
Imprimir

 Your logo

My Company False False

El ingenioso hidalgo don Quijote de la Mancha

Título: El ingenioso hidalgo don Quijote de la Mancha



Autor/es: Miguel de Cervantes Saavedra

Fecha de publicación: 01/01/1605

Idioma: Español

Páginas: 704

Resumen: Don Quijote, hombre valiente y de lo más aventurero, se embarca (arrastrando a Sancho) en locas aventuras, hazañas, b

```

<template id="Report_ficha_libro">
  <t t-call="web.html_container">
    <t t-foreach="docs" t-as="o">
      <t t-call="web.external_layout">
        <div class="page">
          <h2><span t-field="o.name"/></h2>
          <p style="color: black; font-family: Verdana; font-weight: bold;">Titulo:
            <span t-field="o.name"/>
          </p>
          <t t-if="o.imagen">
            
          </t>
          <p></p>
          <t t-if="o.author_ids">
            <p><span style="color: black; font-family: Verdana; font-weight: bold;">Autor/es:</span>
              <span t-field="o.author_ids" />
            </p>
          </t>
          <t t-if="o.date_release">
            <p><span style="color: black; font-family: Verdana; font-weight: bold;">Fecha de publicación:</span>
              <span t-field="o.date_release" />
            </p>
          </t>
          <t t-if="o.language">
            <p><span style="color: black; font-family: Verdana; font-weight: bold;">Idioma:</span>
              <span t-field="o.language" />
            </p>
          </t>
          <t t-if="o.pages">
            <p><span style="color: black; font-family: Verdana; font-weight: bold;">Páginas:</span>
              <span t-field="o.pages" />
            </p>
          </t>
          <t t-if="o.summary">
            <p><span style="color: black; font-family: Verdana; font-weight: bold;">Resumen:</span>
              <span t-field="o.summary" />
            </p>
          </t>
        </div>
      </t>
    </t>
  </t>
</template>

```

Bibliografía

- <https://ubunlog.com/wkhtmltopdf-genera-pdf-imagenes-desde-web/>
- <http://www.geninit.cn/developer/reference/reports.html>
- <https://www.odoo.com/documentation/15.0/developer/reference/backend/actions.html>
- <https://www.odoo.com/documentation/15.0/developer/reference/backend/reports.html>
- https://www.kite.com/python/docs/ansible.cli.to_text