
Tutorial 2: Cómo incorporar diseño de interfaz de usuario en aplicaciones OpenXava con MOSKitt

Tabla de contenidos

Introducción	1
Tarea 1: Creación de un nuevo modelo de Sketcher con MOSKitt.	1
Tarea 2: Creación de un nuevo módulo OpenXava para el mantenimiento de Facturas (Invoice)	3
Tarea 2.1: Modelado del módulo OpenXava para el mantenimiento de Facturas (Invoice)	4
Tarea 3: Lanzar la aplicación OpenXava	15

Introducción

En este tutorial partimos de un Modelo UML2 ya construido para nuestra aplicación. Para obtener este modelo el usuario puede optar por una de las siguientes opciones:

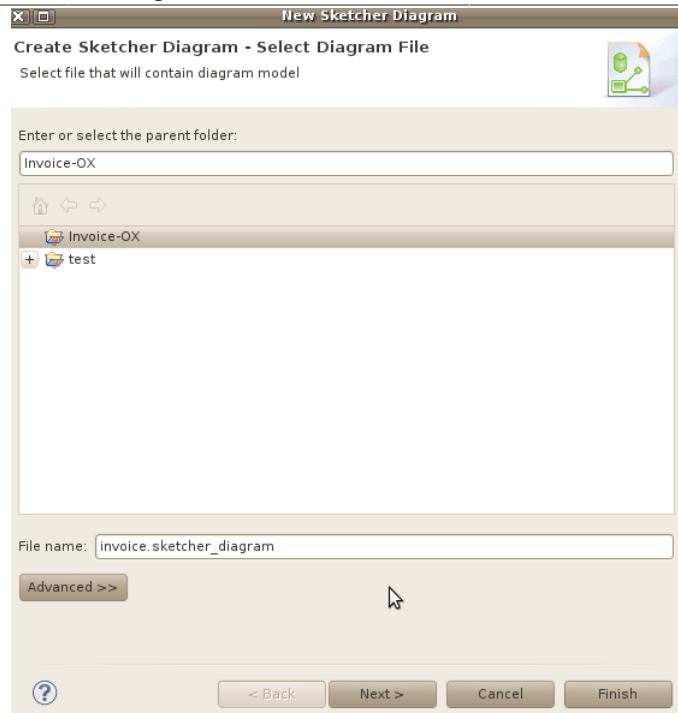
- Construir con MOSKitt el modelo UML2 que se indica en el *Tutorial1: Cómo desarrollar una aplicación OpenXava a partir de un modelo UML2 con MOSKitt*.
- Partir del modelo `invoice_domain` que puede localizar en el directorio `oxportal/workspace/InvoicingModels` proporcionado por MOSKitt en el entorno `oxportal-4.2.X.zip` localizado en la web de MOSKitt.

A partir de este modelo vamos a definir el modelo de interfaz de usuario que debe seguir uno de los Módulos OpenXava que queremos construir, el módulo `invoice`. El resto de módulos se construirán por defecto a partir del modelo UML.

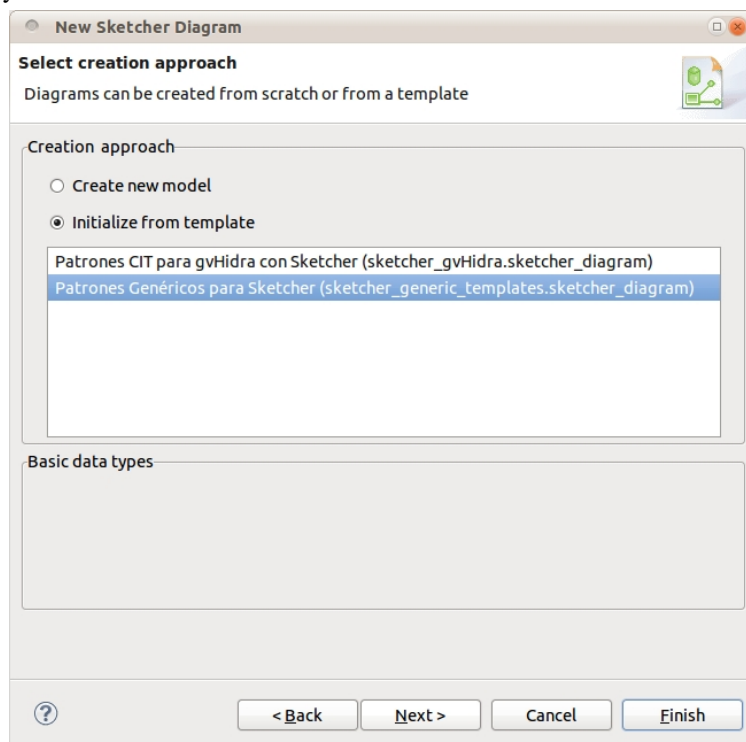
Tarea 1: Creación de un nuevo modelo de Sketcher con MOSKitt.

1. Crear un nuevo modelo de Sketcher con la opción `File/New.../MOSKitt Sketcher Diagram` la cual abrirá el asistente para la creación de modelos Sketcher para que le podamos asignar un nombre al modelo.

Tutorial 2: Cómo incorporar diseño de interfaz de usuario en aplicaciones OpenXava con MOSKitt

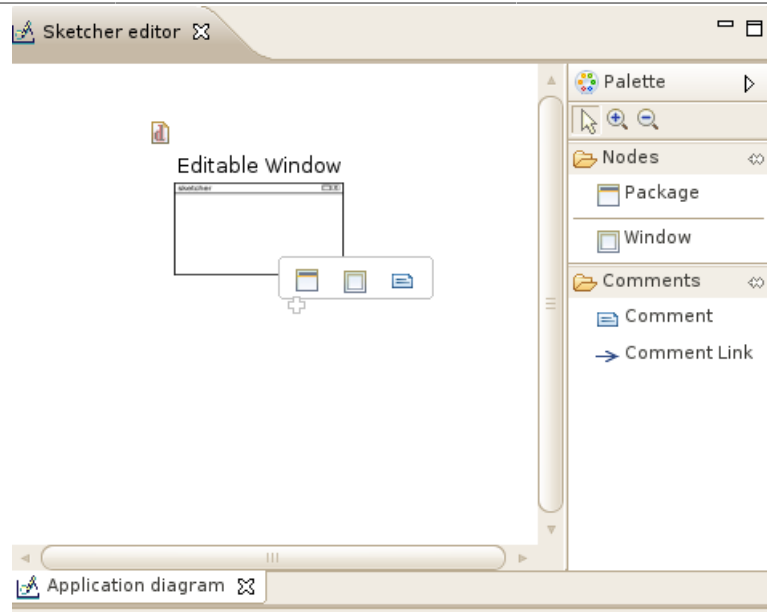


2. Seleccionar la opción Initialize from template y en concreto, de la lista que aparece en el asistente, la opción *Patrones Genéricos para Sketcher* (*sketcher_generic_templates.sketcher_diagram*) tal y como se muestra en la figura que hay a continuación:



Esto pone a disposición del usuario componentes web básicos ya modelados y listos para usar por los analistas y que facilitan la especificación en el Sketcher.

3. Al finalizar el asistente, tendremos creado un nuevo modelo de sketcher. En el editor aparecerá un nuevo diagrama con un único elemento denominado *Editable Window* sobre el que podemos empezar a especificar el primer módulo de nuestra aplicación.



4. Situados sobre el canvas indicar en la pestaña de Propiedades, en la propiedad Id el nombre de la aplicación. se debe especificar en la propiedad 'Id' del elemento raíz (canvas) del modelo de Sketcher (es el canvas que contiene al elemento inicial que aparece en la figura anterior).

Para más información sobre las plantillas proporcionadas por MOSKitt consultar el Manual del Codgen-OpenXava en la web de MOSKitt.

Tarea 2: Creación de un nuevo módulo OpenXava para el mantenimiento de Facturas (Invoice)

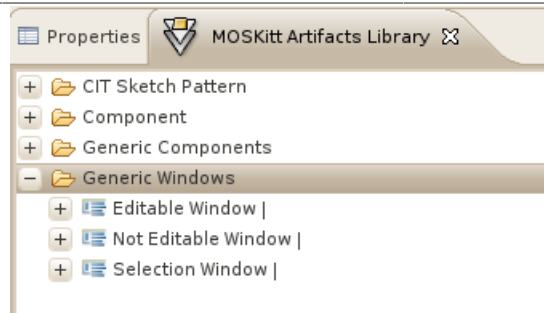
En el modelo Sketcher un módulo OpenXava está representado por un elemento de tipo Window. Para facilitar el modelado de los módulos OpenXava, MOSKitt incorpora ya dos tipos de módulos predefinidos representados por las plantillas:

- Editable Window.
- Not Editable Window.

Para más información sobre las plantillas que proporciona MOSKitt para el modelado de módulos OpenXava consultar el apartado "Plantillas genéricas para el modelado de aplicaciones web" localizado en las sección final de anexos del Manual de Codgen-OpenXava.

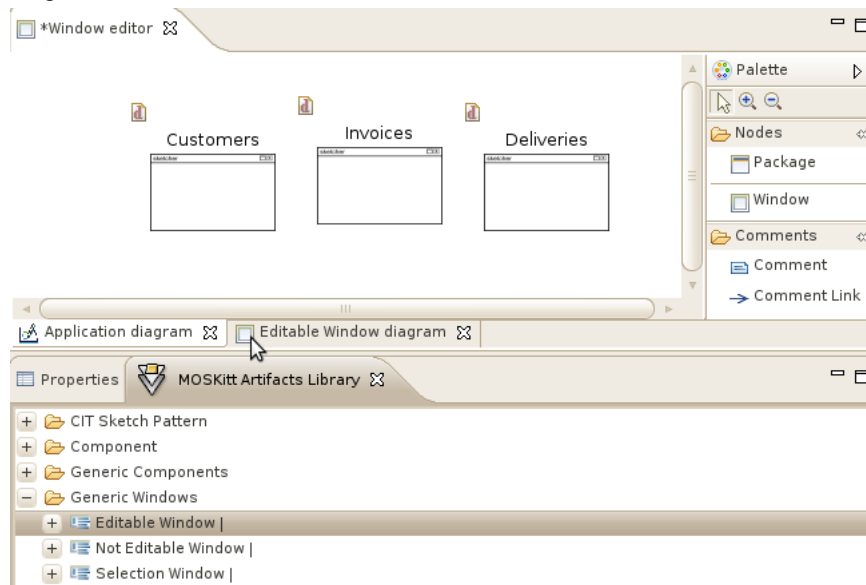
1. Para crear un nuevo módulo OpenXava abrir la vista MOSKitt Artifacts Library y arrastrar la plantilla que más nos interese al canvas inicial del Sketcher, en el cual aparecerá la imagen de una ventana para representarlo.

A continuación se muestra una imagen en la que se muestran las plantillas que proporciona MOSKitt para modelar módulos OpenXava:



¡Ojo! En el grupo `Generic Windows`, la plantilla `Selection Window` representa a una ventana de selección y, aunque en Sketcher es un elemento `Window`, no está representando un módulo OpenXava.

Si vamos arrastrando windows desde la carpeta `Generic Windows` de la vista `MOSKitt Artifacts Library` para especificar cada uno de los módulos OpenXava, al final obtenemos el siguiente diagrama como resultado:



Lo que nos aparece en la figura anterior es una vista del diagrama principal (inicialmente denominado `Application Diagram`), en el cual disponemos de una representación por cada módulo de la aplicación.

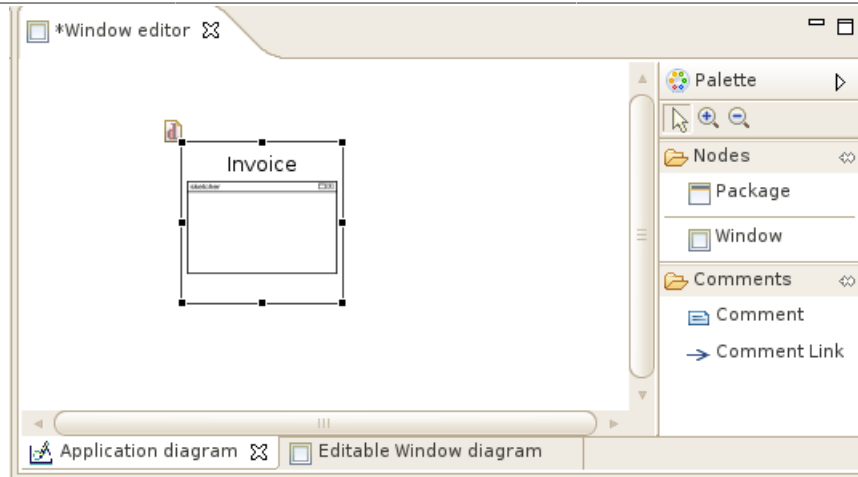
Tarea 2.1: Modelado del módulo OpenXava para el mantenimiento de Facturas (Invoice)

Vamos a modelar un primer módulo para el mantenimiento de Facturas (`Invoice`) en nuestra aplicación de Facturación. El módulo que vamos a modelar con Sketcher está basado en el que se puede consultar en la web de OpenXava en la dirección <http://www.openxava.org/web/guest/invoice>.

Puesto que nos interesa disponer de las operaciones básicas de mantenimiento (inserción, edición y borrado), haremos uso de la plantilla `Editable Window` (como es el primer módulo podemos aprovechar la `Editable Window` que nos proporciona Sketcher al crear un modelo nuevo).

Para editar la ventana debemos hacer doble click sobre la figura que representa al módulo en el diagrama principal (con el nombre por defecto de `Application diagram`):

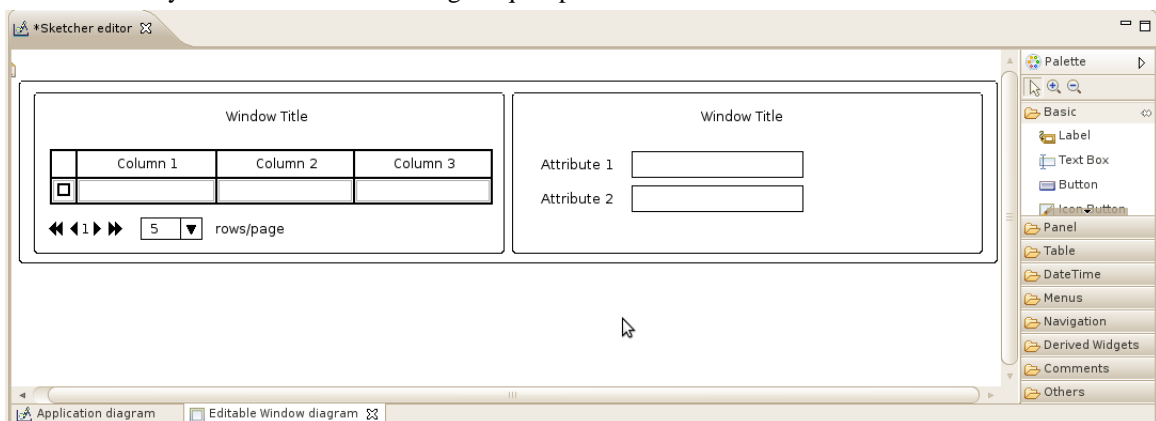
Tutorial 2: Cómo incorporar diseño de interfaz de usuario en aplicaciones OpenXava con MOSKitt



De esta ventana vamos a especificar la siguiente información:

- Propiedad *Id* de la Window: está localizada en la pestaña de propiedades, subpestaña Basic de la Window. En esta propiedad indicaremos el nombre del módulo, en este caso, Invoice.

Al hacer doble click se abrirá una nueva pestaña en el editor para poder modificar el contenido de la ventana tal y como se muestra en la figura que aparece a continuación:



Como podemos ver en la figura anterior, no empezamos de cero, sino que partimos ya de un modelo inicial que nos proporciona la plantilla Editable Window. Ésta plantilla contiene dos elementos principales de tipo Panel:

- Un Panel que representa la vista en formato Lista (o List) del módulo.
- Un Panel que representa la vista en formato Detalle (o Detail) del módulo.

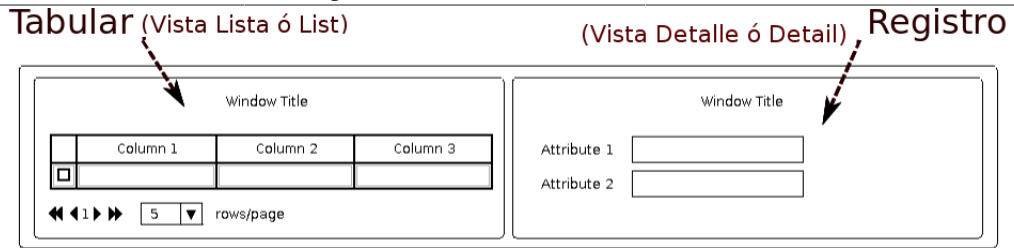
Para que el generador de código reconozca en estos paneles a las correspondientes vistas OpenXava, estos elementos Panel están etiquetados respectivamente con los tags:

- Tabular: Tag utilizado para marcar la vista en formato Lista (List).
- Registro: Tag utilizado para marcar la vista en formato Detalle (Detail).

Los tags asociados a un elemento del Sketcher aparecen en la subpestaña Tags de en la Vista Properties.

A continuación se muestra qué elemento gráfico en el Sketcher está etiquetado con los tabs Tabular y Registro:

Tutorial 2: Cómo incorporar diseño de interfaz de usuario en aplicaciones OpenXava con MOSKitt

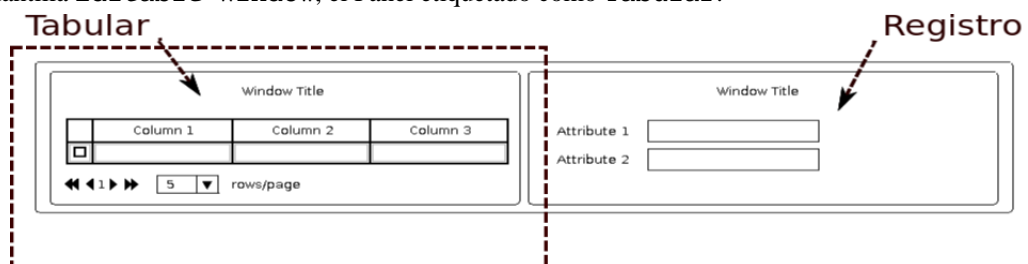


¡Ojo! Para evitar errores de interpretación vamos a aclarar la nomenclatura utilizada por MOSKitt para referirse a conceptos diferentes:

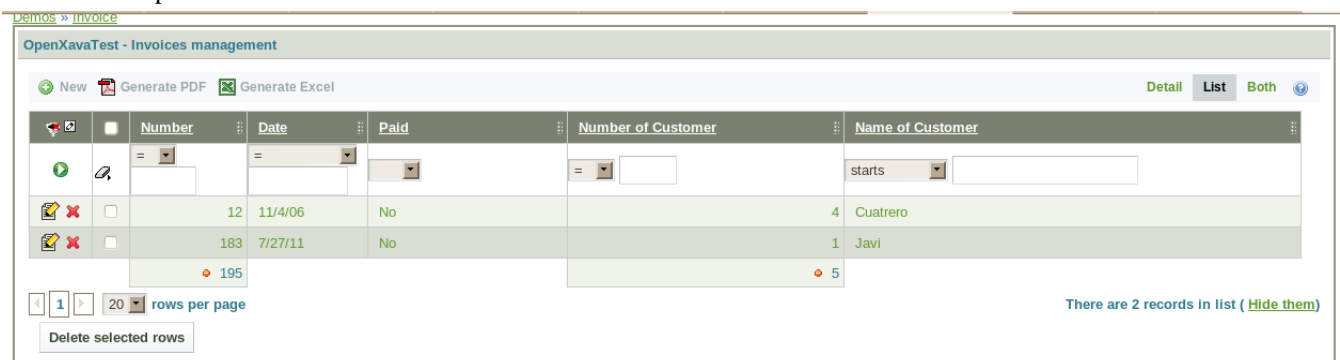
- La etiqueta (tag) `Tabular` se utiliza para representar a la vista `Lista` (`List`) de un módulo OpenXava.
- La etiqueta (tag) `Registry` se utiliza para representar a la vista `Detalle` (`Detail`) de un módulo OpenXava.
- La etiqueta (tag) `Detail` se utiliza para representar a un bloque `Detalle` en una jerarquía Maestro/Detalle contenida en un módulo OpenXava (no confundir con la vista `Detalle` correspondiente a la etiqueta `Registry`).

Tarea 2.1.1: Modelar la vista Lista/List del mantenimiento de Facturas (Invoice)

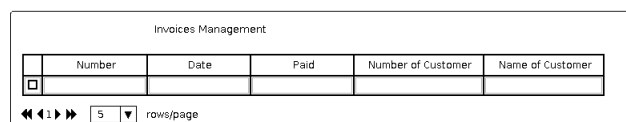
El siguiente paso es modelar la vista en formato `Lista` (`List`) del módulo de mantenimiento de Facturas (`Invoice`). Esta vista se corresponde con uno de los componentes que forman parte de la plantilla `Editable Window`, el Panel etiquetado como `Tabular`:



Siguiendo el ejemplo de la página web de OpenXava, la ventana que vamos a modelar tiene el siguiente aspecto:



Haciendo uso de los elementos de la paleta que nos proporciona MOSKitt-Sketcher completamos el panel anterior para que tenga el siguiente aspecto:

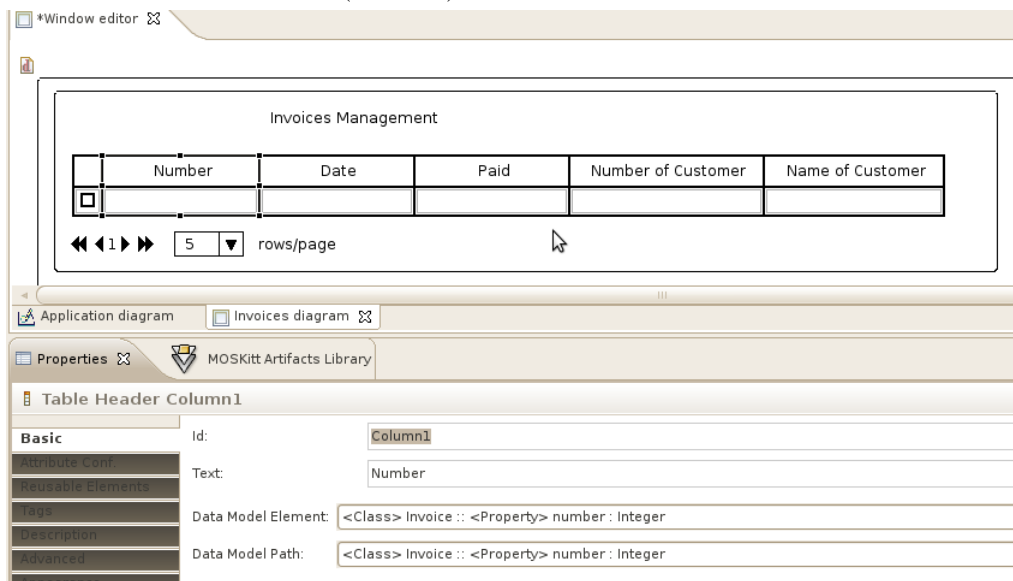


Básicamente, lo único que hemos tenido que hacer es:

- Añadir las nuevas columnas a la tabla que nos proporciona la plantilla hasta disponer de las 5 necesarias para el ejemplo (son elementos `Table` `Column` localizados en la paleta del Sketcher, en la pestaña `Table`).
- Indicar el título de la cabecera de cada una de las columnas de la tabla. Para hacer esto le hemos dado valor a la propiedad `Text` localizada en la subpestaña `Basic` de la pestaña de `Properties` del elemento `Table` `Column` correspondiente.
- Enlazar los widgets (elementos `Table` `Column` en este caso) con las correspondientes propiedades de UML2 completando las propiedades: *Data Model Element* y *Data Model Path*.
- Enlazar las columnas etiquetadas como `Number`, `Date` y `Paid` con las propiedades `Number`, `Date` y `Paid` de la clase **Invoice**.
- Enlazar las columnas etiquetadas como `Number of Customer` y `Name of Customer` con las propiedades `Number` y `Name` de la clase **Customer** a través de la asociación **Invoice_Customer** que aparece cuando desplegamos la clase **Invoice** en el wizard `Object Selection` (¡jojo!, esto es muy importante).

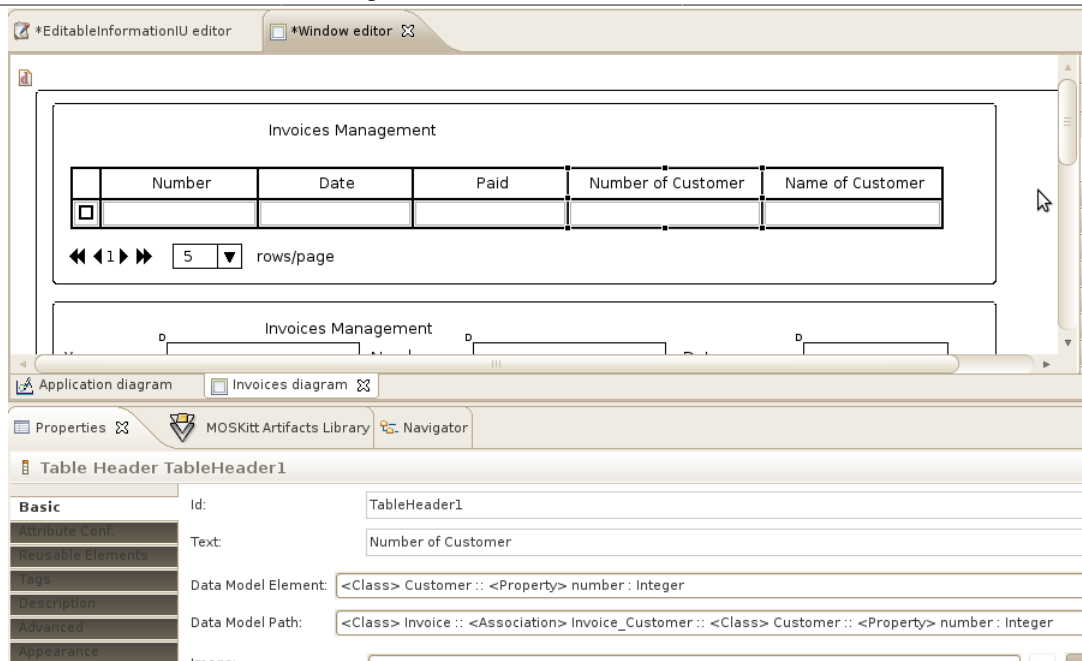
Para más información sobre cómo relacionar los modelos de Sketcher y UML2 consultar el apartado que aparece más adelante en este punto del manual y que trata sobre "Cómo relacionar elementos del Sketcher con elementos de UML2".

En la siguiente figura podemos ver cómo se han completado estas propiedades para la columna referente al Número de la Factura (`Number`):



En la siguiente figura podemos ver cómo se han completado estas propiedades para la columna referente al Número del Cliente de la factura (`Number of Customer`). Observar la diferencia con respecto a la propiedad *Data Model Element Path* de la figura anterior:

Tutorial 2: Cómo incorporar diseño de interfaz de usuario en aplicaciones OpenXava con MOSKitt

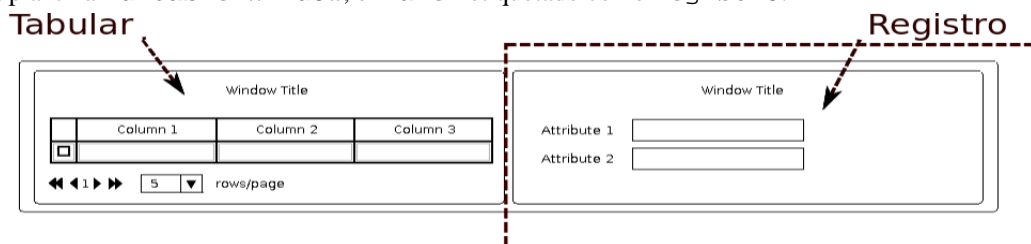


Para más información sobre cómo relacionar los modelos de Sketcher y UML2 consultar el apartado que aparece más adelante en este manual y que trata sobre Cómo relacionar elementos del Sketcher con elementos de UML2.

Para más información sobre cómo utilizar el editor MOSKitt-Sketcher consultar el Manual de Usuario en la ayuda de MOSKitt.

Tarea 2.1.2: Modelar la vista Detalle/Detail del mantenimiento de Facturas (Invoice)

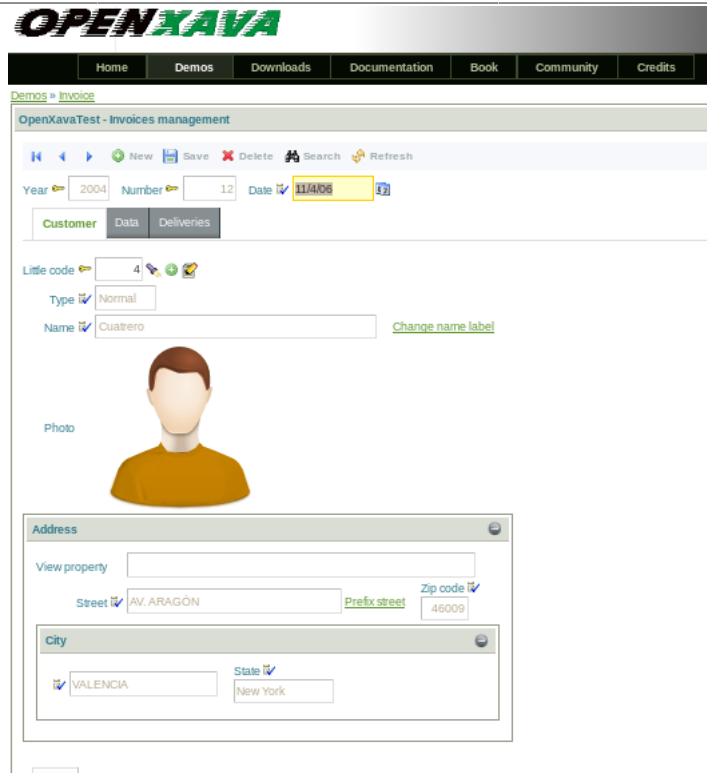
El siguiente paso es modelar la vista en formato Detalle (Detail) del módulo de mantenimiento de facturas (Invoice). Esta vista se corresponde con uno de los componentes que forman parte de la plantilla Editable Window, el Panel etiquetado como Registro:



Siguiendo el ejemplo de la página web de OpenXava, la ventana que vamos a modelar es más compleja que la anterior ya que contiene varios paneles anidados, así es que vamos a ir por partes.

El primer panel que nos interesa modelar tiene el siguiente aspecto:

Tutorial 2: Cómo incorporar diseño de interfaz de usuario en aplicaciones OpenXava con MOSKitt



En él aparece una cabecera con datos de la factura y tres pestañas. En la primera de ellas, etiquetada como `Customer`, se muestra sólo información del cliente de la factura.

Haciendo uso de los elementos de la paleta que nos proporciona el Sketcher completamos el panel original para que tenga el siguiente aspecto:

En este caso ya hemos tenido un poco más de trabajo ya que hemos tenido que:

- Reubicar los `TextBox` que la plantilla proporcionaba por defecto para alinearlos en horizontal.
- Añadimos los nuevos elementos `TextBox` necesarios hasta disponer de los 4 que aparecen en la figura anterior correspondientes a los campos `Year`, `Number`, `Paid` y `Date`.

- Modificar la propiedad *Text* de cada una de las *Label* que acompañan a los *TextBox* para poner la etiqueta requerida en cada caso (*Year*, *Number*, *Paid* y *Date*).
- Enlazar todos los campos con las correspondientes propiedades de la clase **Invoice** del diagrama de UML2 (para más información sobre la relación entre los modelos de Sketcher y UML2 consultar el apartado sobre Cómo relacionar elementos del Sketcher con elementos de UML2). Para ello debemos completar las propiedades: *Data Model Element* y *Data Model Path*.
- Añadir un elemento *TabPanel* (panel de pestañas) con 3 *TabItems* (pestañas). Estos elementos se pueden arrastrar desde la subpestaña *Panel* de la paleta del Sketcher.

Para cada uno de los *TabItems* contenidos en esta vista hemos tenido que:

- Modificar la propiedad *Text* para poner la etiqueta requerida en cada caso (*Customer*, *Data* y *Deliveries*).
- Incluir los widgets necesarios.

Vamos a ir recorriendo cada uno de ellos para ver qué hemos tenido que hacer en cada caso para modelarlo:

- **TabItem Customer:** Esta pestaña muestra los datos del cliente de la factura tal y como podemos ver en la figura anterior. Para modelar esta pestaña hemos tenido que:
 - Incluir un componente *Single Selection Group Panel* arrastrándolo desde la carpeta *Generic Components* de la vista *MOSKitt Artifacts Library* (para más información sobre las plantillas que proporciona Sketcher para facilitar el modelado consultar la sección "Plantillas genéricas para el modelado de aplicaciones web").
 - Añadir un elemento *Image* para el campo *photo* de la clase **Customer**, con su elemento *Label* correspondiente.
 - Modificar la propiedad *Text* de cada *Label* que acompaña a los *TextBox/Image* para poner la etiqueta requerida en cada caso (*Little Code*, *Type*, *Name* y *Photo*).
 - Enlazar todos los widgets con las propiedades *number*, *type*, *name* y *photo* de la clase **Customer** del diagrama de UML2 a través de la asociación **Invoice_Customer** que aparece cuando desplegamos la clase **Invoice** en el wizard *Object Selection* (¡jojo!, esto es muy importante). Para ello debemos completar las propiedades: *Data Model Element* y *Data Model Path* de cada widget. Para más información sobre la relación entre los modelos de Sketcher y UML2 consultar el apartado sobre "Cómo relacionar elementos del Sketcher con elementos de UML2".
- **TabItem Data:** Esta pestaña muestra las líneas de la factura en formato tabular tal y como podemos ver en la figura del modelo de sketcher que aparece a continuación. Para modelar esta pestaña hemos tenido que:
 - Incluir un componente *Detail-Tabular Panel* arrastrando desde la carpeta *Generic Components* de la vista *MOSKitt Artifacts Library* (para más información sobre las plantillas que proporciona Sketcher para facilitar el modelado consultar la sección "Plantillas genéricas para el modelado de aplicaciones web").
 - Añadir nuevas columnas a la tabla hasta disponer de las 7 que necesitamos (son elementos *Table Column* localizados en la pestaña *Table* de la paleta del Sketcher).
 - Indicar el título de la cabecera de cada una de las columnas de la tabla dando valor a la propiedad *Text* localizada en la subpestaña *Basic* de la pestaña de *Properties* de cada elemento *Table Column*.
 - Enlazar todos los campos con las correspondientes propiedades de la clase **InvoiceDetail** del diagrama de UML2 a través de la asociación **Invoice_InvoiceDetail** que aparece cuando desplegamos la clase **Invoice** en el wizard *Object Selection* (¡jojo!, esto es muy

Tutorial 2: Cómo incorporar diseño de interfaz de usuario en aplicaciones OpenXava con MOSKitt

importante). Para ello debemos completar las propiedades: *Data Model Element* y *Data Model Path* de cada widget. Para más información sobre la relación entre los modelos de Sketcher y UML2 consultar el apartado sobre "Cómo relacionar elementos del Sketcher con elementos de UML2".

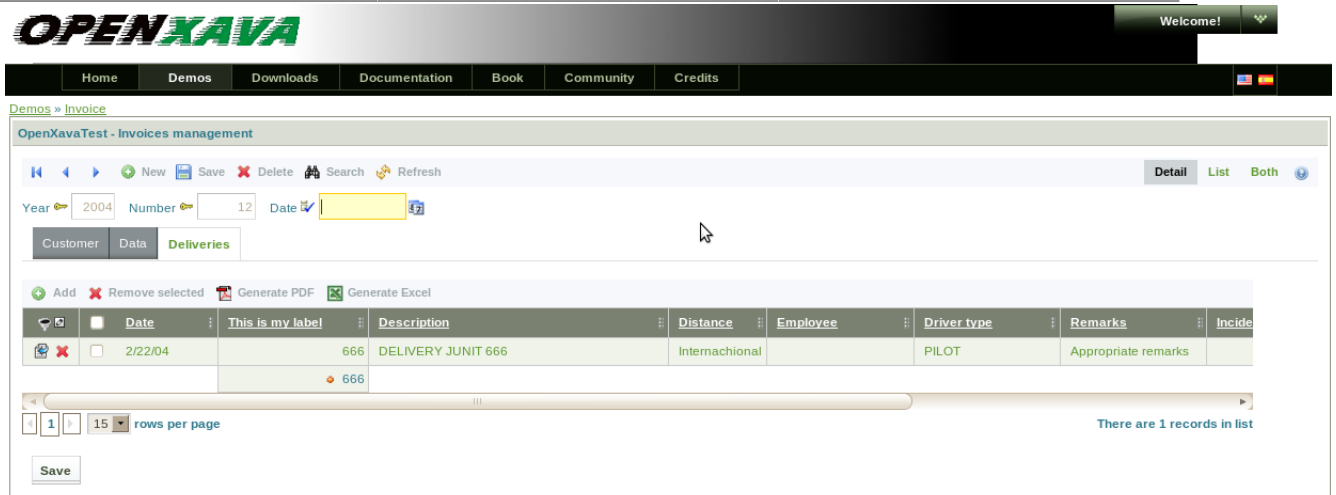
The screenshot shows the OpenXava web application interface. At the top, there's a header with the OpenXava logo and navigation links: Home, Demos, Downloads, Documentation, Book, Community, Credits. Below the header, there's a sub-header for 'OpenXavaTest - Invoices management'. The main content area has tabs for 'Customer', 'Data', and 'Deliveries'. The 'Deliveries' tab is selected, showing a table with columns: Service type, Quantity, Unit price, Delivery date, Remarks, Amount, and Free. The table contains two rows of data. At the bottom, there are pagination controls showing '1' of '10' rows per page and a 'Save' button.

Este formulario se modela en Sketcher tal y como aparece en la siguiente figura:

The diagram shows the Sketcher model of the 'Invoices Management' form. It includes fields for Year, Number, and Date. Below these are tabs for Customer, Data, and Deliveries. The Deliveries tab is active, showing a table with columns: Service Type, Quantity, Unit Price, Delivery Date, Remarks, Amount, and Free. The table has one row of data. Below the table are pagination controls showing '5' rows/page and a 'Save' button.

- **TabItem Deliveries:** Esta pestaña muestra la lista de albaranes asociados con la factura. No permite insertar nuevos albaranes, sólo relacionar albaranes existentes con la factura que tenemos activa en el módulo. Para modelar esta pestaña hemos tenido que:
 - Incluir un componente Multiple Selection Panel localizado en la carpeta Generic Components de la vista MOSKitt Artifacts Library (para más información sobre las plantillas que proporciona Sketcher para facilitar el modelado consultar la sección "Plantillas genéricas para el modelado de aplicaciones web").
 - Completar el componente del mismo modo que hemos procedido hasta ahora con el resto de paneles con los widgets correspondientes.
 - Enlazar todos los campos con las correspondientes propiedades de la clase **Delivery** a través de la relación Invoicing_Delivery del diagrama de UML2 (para más información sobre la relación entre los modelos de Sketcher y UML2 consultar el apartado sobre "Cómo relacionar elementos del Sketcher con elementos de UML2"). Para ello debemos completar las propiedades: *Data Model Element* y *Data Model Path* de cada widget.

Tutorial 2: Cómo incorporar diseño de interfaz de usuario en aplicaciones OpenXava con MOSKitt



Este formulario se modela en el Sketcher tal y como aparece en la siguiente figura:

Para más información sobre cómo utilizar el editor MOSKitt-Sketcher consultar el Manual de Usuario en la ayuda de MOSKitt proporcionado por este módulo.

Tarea 2.1.3: Modelar la vista Both del mantenimiento de Facturas (Invoice)

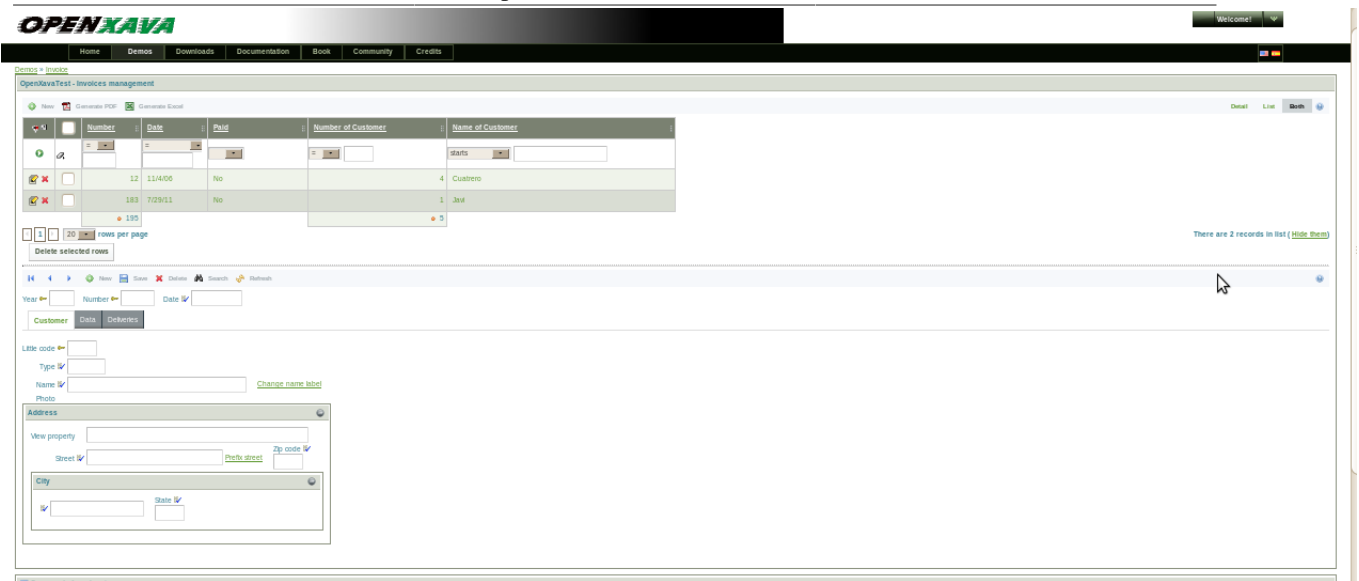
Esta vista no es necesario modelarla ya que lo hemos ido haciendo mientras hemos ido modelando cada una de las dos vistas del módulo: Tabular y Registro.

Tabular (Vista Lista ó List)

Registro (Vista Detalle ó Detail)

Siguiendo el ejemplo de la página web de OpenXava, la ventana que hemos modelado tiene el siguiente aspecto:

Tutorial 2: Cómo incorporar diseño de interfaz de usuario en aplicaciones OpenXava con MOSKitt



Haciendo uso de los elementos de la paleta que nos proporciona el Sketcher hemos ido completando los paneles de Lista y Detalle y al final el componente completo tiene el siguiente aspecto:

The sketcher diagram illustrates the layout of the Invoices management interface. It consists of two main panels: 'Invoices List' and 'Invoices Detail'. The 'Invoices List' panel contains a table with columns: Number, Date, Paid, Number of Customer, and Name of Customer. Below the table are pagination controls (5 rows/page). The 'Invoices Detail' panel contains fields for Year, Number, Paid, Date, and Comment. Below these fields are tabs for 'Customer', 'Data', and 'Deliveries'. The 'Customer' tab is active, showing fields for Little Code, Type, Name, and Photo.

Tarea 2.1.4: Cómo relacionar los elementos del Sketcher con los del Modelo UML2

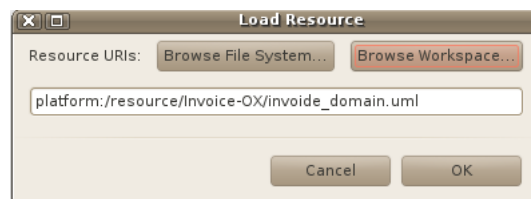
Para completar la especificación de la interfaz necesitamos enlazar muchos de los widgets con elementos del modelo de clases. De este modo el analista podrá indicar:

Tutorial 2: Cómo incorporar diseño de interfaz de usuario en aplicaciones OpenXava con MOSKitt

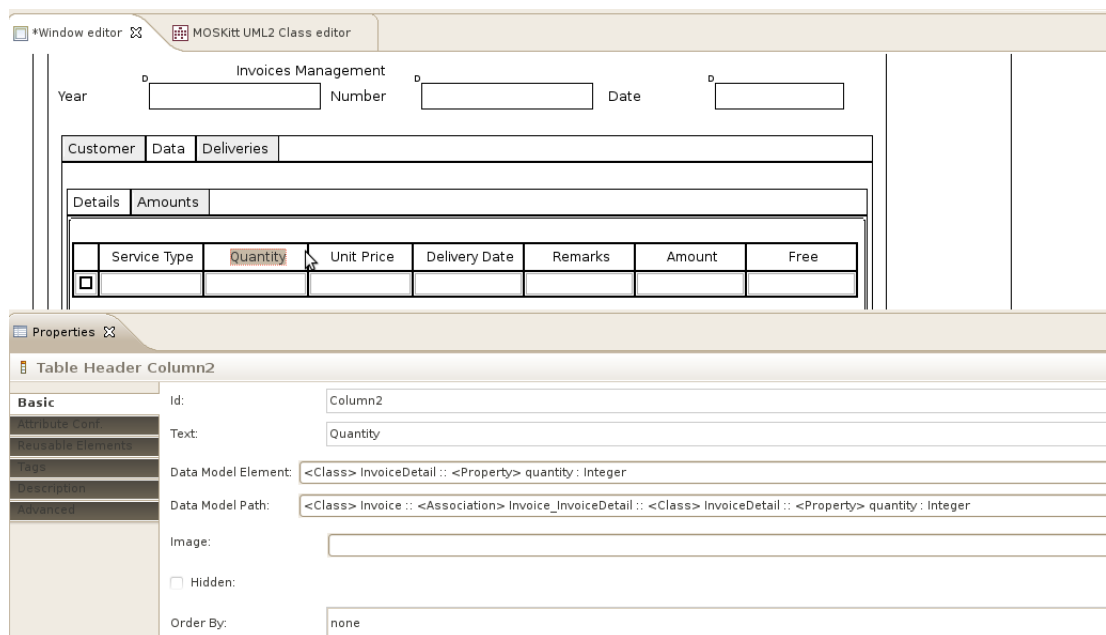
- Qué propiedades de las clases se deben mostrar en los widgets de la interfaz y si se deben poder editar o no.
- Qué métodos de las clases pueden ser invocados desde la interfaz (normalmente pulsando un botón).
- Qué enumerados deben ser ofrecidos en una lista de valores
- etc...

Para tener acceso a los elementos del modelo de UML2 en primer lugar tenemos que cargar este modelo como un nuevo recurso para el Sketcher. Para ello, debemos hacer click con el botón derecho sobre el lienzo del Sketcher y seleccionar la opción Load Resource del menú contextual.

En este momento aparecerá el siguiente asistente para cargar el modelo y, haciendo uso del botón Browse Workspace... debemos buscar el modelo de clases invoice_domain.uml (en nuestro caso de estudio) y finalmente pulsaremos OK.



A partir de este momento todos los elementos del modelo UML2 serán accesibles desde el modelo de Sketcher de la aplicación de facturación y podremos completar las propiedades necesarias de los elementos de cada panel.



En la imagen anterior se muestran las propiedades de un campo de texto. Para cada elemento debemos dar valor a las siguientes propiedades:

- *Id*: Identificador del widget (normalmente por claridad se indica el nombre de la propiedad de UML2 con la que se relaciona).
- *Elemento Modelo*: muestra con qué elemento del modelo de UML2 se relaciona el widget.
- *Ruta Elemento Modelo*: indica cómo se debe navegar en el modelo UML2 para alcanzar la información a mostrar. Vamos a ver su utilidad con un ejemplo:

- Si selecciono la propiedad `Quantity` de la clase `Invoice_Detail` estaremos indicando que en la interfaz se van a mostrar todas las líneas de todas las facturas y, para cada una de ellas su propiedad `Quantity`.
- Sin embargo, si selecciono la misma propiedad, pero navegando desde la clase `Invoice` (a través de la asociación que tengo definida `Invoice_InvoiceDetail`), estaré indicando que lo que voy a mostrar es sólo la propiedad `Quantity` de las líneas de cada factura.

En la siguiente figura podemos observar en el extremo superior izquierdo de los campos de texto que aparece el símbolo **D** para indicar que ya está definida la relación con el modelo UML2 (modelo de Datos). Para saber con qué elemento de UML2 se ha relacionado el widget debemos ir a la pestaña de propiedades y consultar las propiedades *Elemento Modelo* y *Ruta Elemento Modelo* descritas anteriormente.

The screenshot shows a web application titled "Invoices Management". Below the title, there are three text input fields. The first field is preceded by the label "Year" and has a small "D" symbol above it. The second field is preceded by the label "Number" and also has a small "D" symbol above it. The third field is preceded by the label "Date" and has a small "D" symbol above it. Each field is a simple rectangular box.

Para más información sobre todos los elementos disponibles en la paleta del Sketcher y sus propiedades consultar el manual de usuario del Sketcher en la ayuda de MOSKitt.

Tarea 3: Lanzar la aplicación OpenXava

Para ejecutar esta tarea seguir los pasos indicados en el tutorial *Tutorial 3: Cómo ejecutar una aplicación OpenXava con OXPortal*.