



Anexo

Herramienta de Formularios Dinámicos (K2B Dynamic Form)

Anexo – Herramienta de Formularios Dinámicos (K2B Dynamic Form)

El módulo **K2B Dynamic Form** es un producto que permite crear interfaces de usuario flexibles en tiempo de ejecución. El mismo soporta diferentes **tipos de datos** tales como lógicos, caracteres, numéricos, fechas, fechas con hora, contraseñas, colores, correos electrónicos, direcciones web, archivos, entre otros.

Adicionalmente se puede definir diferentes **controles para la entrada y visualización** de dichos tipos de datos. A modo de ejemplo, un campo de tipo carácter lo podemos ingresar utilizando cuadros de texto, área de texto, combo box, combo box dinámicos, campos que se autocompletan (suggest).

Es relevante destacar la posibilidad de definir **reglas** en tiempo de ejecución. Las reglas pueden ser clasificadas en **reglas de visualización** o **reglas de negocio**. Las reglas de visualización serán evaluadas exclusivamente en el cliente. Las reglas de negocio son evaluadas en el servidor y de ser posible también en el cliente. A modo de ejemplo presentaremos una regla de visualización. Las reglas de visualización nos permiten mostrar, ocultar, habilitar y deshabilitar campos; dichas reglas van a ser evaluadas en función de valores de otros campos. Si llevamos dichas reglas teóricas a un caso práctico podemos tomar como ejemplo el formulario de **AFIDI**. En el mismo tenemos un campo que dice si el "producto esta acompañado de substratos o soporte", en caso de que el mismo este marcado debemos incorporar la "descripción del substrato o soporte".

A través de este componente se generan, en tiempo de **implantación de cada operación de comercio exterior**, los formularios que serán necesarios para solicitar y presentar la información a los usuarios del sistema. Este módulo será ejecutado sobre la herramienta de **workflow**, permitiendo configurar en tiempo de implantación cada una de las pantallas.

El módulo de formularios dinámicos incorpora el concepto de **dominio**, cada dominio nos permite definir un **concepto semántico**, a modo de ejemplo podemos pensar en el dominio País. Cada dominio debe definir el tipo de datos, los componentes de entrada/visualización, reglas para carga dinámica, reglas visualización y validación de los campos, etc. En VUCE se predefinirán los dominios definidos por la OMA para poder utilizarlos durante el proceso de creación de formularios.

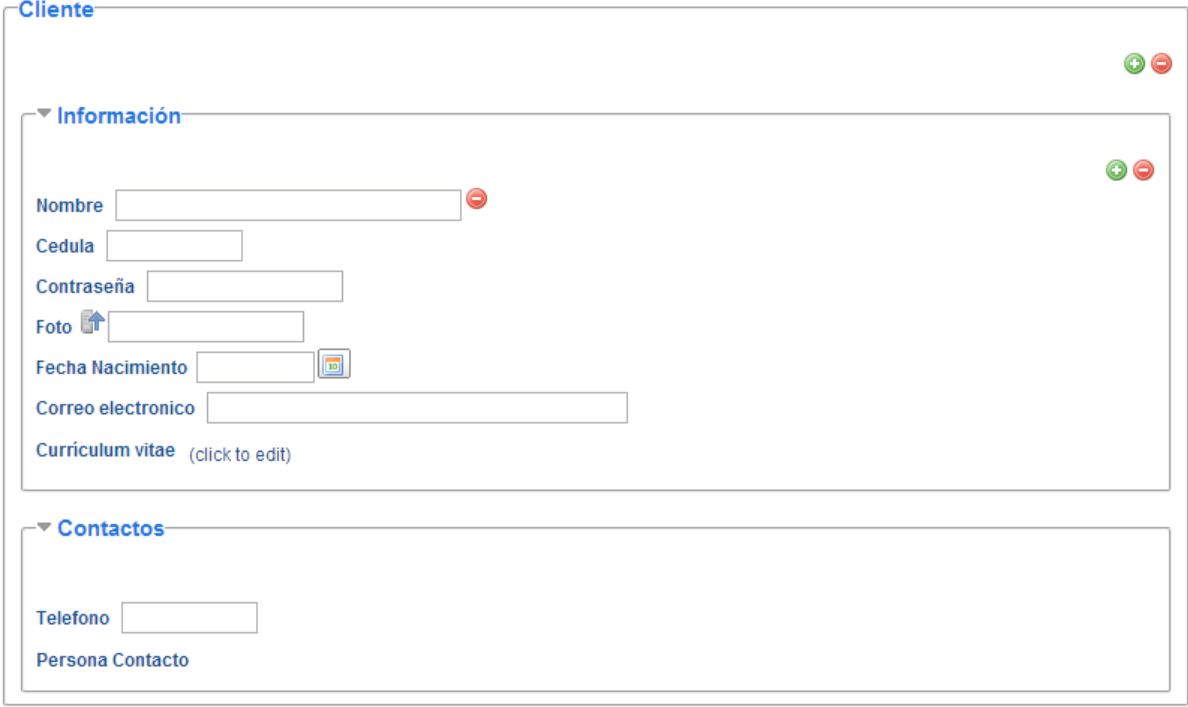
La herramienta **GXScheduler** será utilizada sobre los formularios dinámicos, la misma permite gestionar un calendario de eventos cronológicos. Los eventos pueden ser vistos por día, semana o mes, así como también se pueden agregar, actualizar y eliminar directamente.

1.1 Conceptos incluidos en K2B Dynamic Form

1. Formulario

Un formulario es una pantalla para ingreso, modificación o despliegue de datos dentro del sistema. Dentro del alcance del proyecto VUCE utilizamos la palabra formulario como la definición de como debe ser construida la pantalla. En la misma debemos definir que elemento forman parte de dicho formulario y como se deben mostrar.

A continuación presentamos un ejemplo de como es la interfaz grafica para construir un formulario:



Cliente

▼ Información

Nombre

Cedula

Contraseña

Foto

Fecha Nacimiento

Correo electronico

Curriculum vitae (click to edit)

▼ Contactos

Telefono

Persona Contacto

Guardar Formulario

Como podemos ver en la imagen anterior, el modo de construcción de formulario nos permite agregar elementos al formulario, eliminar elementos del formulario y/o guardar el formulario.

2. Instancia de Formulario

Una instancia de formulario es la materialización de un determinado formulario (un caso en concreto). Cada instancia de formulario va a tener asociado un conjunto de valores para los elementos del formulario. A modo de ejemplo nosotros podemos definir el formulario X, dicho formulario puede ser ejecutado muchas veces, cada ejecución me va a generar una nueva instancia de formulario.

La siguiente imagen muestra una instancia del formulario para ingreso de datos de cliente en modo actualización de datos.


Ciente


▼ **Información**

Nombre

Cedula

Contraseña

Foto 

Fecha Nacimiento 

Correo electronico

Curriculum vitae

Contactos

Telefono	Persona Contacto	
29999999	Contacto 1	✘
28888888	Contacto 2	✘

Insertar

Guardar Formulario

3. Modo de una Instancia de Formulario

Una misma instancia de formulario puede ser accedida en diferentes modos, los modos soportados para las instancias de formulario son: visualización, creación, actualización y borrado.

Cuando mostramos un formulario en **modo creación** o en **modo actualización**, el usuario puede modificar todos los campos que fueron marcados como editables. En estos modos el usuario dispone de un botón que le habilita guardar los valores de los elementos en esta instancia del formulario.

Cuando mostramos un formulario en **modo visualización** o en **modo borrado**, todas las instancias de los elementos del formulario son mostrados de solo lectura. La diferencia entre **modo visualización** o en **modo borrado** es que en el segundo el usuario dispone de un botón que le permite borrar la instancia del formulario.

La siguiente imagen muestra una instancia del formulario para ingreso de datos de cliente en modo visualización de datos.

Cliente

▼ **Información**

Nombre Juan Perez

Cedula 1.111.111-9

Contraseña aaaaaaaaaaaaaa

Foto

Fecha Nacimiento 12/12/12

Correo electronico

Currículum vitae

Contactos

Telefono	Persona Contacto
29999999	Contacto 1
28888888	Contacto 2

4. Elemento:

El elemento es parte esencial de la definición de un formulario, podemos pensar que un formulario es una colección ordenada de elementos. Los elementos pueden ser simples o agrupaciones de elementos (esta relación es recursiva).

Para los elementos simples definiremos:

- **Nombre** – El nombre del elemento es lo que mostraremos como etiqueta del mismo en el formulario.
- **Descripción** – La descripción sirve para documentar la finalidad del elemento, la misma no es mostrada en ningún momento dentro del formulario. La mayor utilidad es simplificar la parametrización del sistema, esto favorece la reutilización de un elemento en diferentes formularios.
- **Tipo de datos** – Los posibles tipos de datos son: lógicos, caracteres, numéricos, fechas, fechas con hora, contraseñas, colores, correos electrónicos, direcciones web, archivos.

- **Propiedades de los tipo de datos** – Cada tipo de datos tiene su propio conjunto de propiedades. Por ejemplo el tipo de datos numérico tiene como propiedades el largo total y la cantidad de decimales.
- **Componente de visualización/edición** – Un tipo de datos puede ser mostrado/editado de diferentes maneras, en esta propiedad elegimos cual es el componente para visualización/edición. A modo de ejemplo el tipo de datos numérico puede ser ingresado en una caja de texto con formato o con un control de deslizante (Slider).
- **Tipo de Herencia** – Como ya mencionamos el mismo elemento puede ser reusado en mas de un formulario. El tipo de herencia nos especifica como debemos comportarnos en caso de estar mostrando un elemento que fue ingresado en un formulario anterior dentro de la misma OCE (Operación de Comercio Exterior).
- **Dominio** – En muchos casos nosotros no queremos configurar toda esta información por cada elemento, para estos casos podemos definir un dominio y basar los elementos en el mismo. Cuando nosotros definimos un elemento basado en un dominio, dicho elemento hereda todas las propiedades del dominio.

Como mencionamos previamente los elementos pueden ser a su vez agrupaciones de otros elementos, en este caso el tipo debe ser grupo. Cuando estamos frente a un elemento de tipo grupo tenemos dos opciones:

- a) **Agrupación lógica de elementos.** La misma tiene un título y se representara como un conjunto de elementos con un recuadro alrededor. Este tipo de agrupación de elementos va a tener la opción de ser colapsable o no.
- b) **Grilla de elementos.** La misma sirve cuando queremos permitir ingresar o mostrar un conjunto de registros. Las grillas tienen propiedades que dicen si permiten inserción, modificación y/o edición de los diferentes registros de las grillas.

En la siguiente imagen podemos ver resaltamos "Información" pues es una **Agrupación lógica de elementos** y "Contactos" pues es una **"Grilla de elementos"**

Ciente

▼ **Información**

Nombre Juan Perez

Cedula 1.111.111-9

Contraseña aaaaaaaaaaaaaa

Foto

Fecha Nacimiento 12/12/12

Correo electronico

Curriculum vitae

Contactos

Telefono	Persona Contacto
29999999	Contacto 1
28888888	Contacto 2

5. Dominio:

Los dominios son la representación en el sistema de un **concepto semántico** de existente en la realidad. En los dominios vamos a poder modelar las propiedades que debe tener dicho concepto en el sistema. El objetivo de modelar los conceptos de esta manera es la reutilización de los mismos. Esto lleva a una uniformización de los formularios.

En los dominios podemos modelar las siguientes propiedades:

- **Tipo de datos** – Los posibles tipos de datos son: lógicos, caracteres, numéricos, fechas, fechas con hora, contraseñas, colores, correos electrónicos, direcciones web, archivos.
- **Propiedades de los tipo de datos** – Cada tipo de datos tiene su propio conjunto de propiedades. A modo de ejemplo el tipo de datos para representar caracteres tiene el largo total como propiedad.
- **Componente de visualización/edición** – Un tipo de datos puede ser mostrado y editado de diferente manera, en esta propiedad elegimos cual es el componente para visualización y edición. A modo de ejemplo el tipo de datos para representar caracteres puede ser ingresado por cajas de texto, combo box, etc.
- **Propiedades de los componentes de visualización/edición** – Cada componente de visualización tiene asignado un conjunto de propiedades las cuales podemos configurar. A modo de ejemplo el control de tipo **caja de texto con autocompletado** debe tener asociado un programa de carga, cantidad de letras mínimas antes de lanzar el pedido al programa de carga, cantidad máxima de resultados a mostrar, si se debe forzar o no la selección.
- **Valor por defecto** – Valor por defecto para los elementos del dominio.

En el próximo capítulo definiremos detalladamente todos los tipos de datos, todos los controles de visualización/edición.; para cada uno de ellos definiremos sus propiedades.

6. Versionado:

Cuando nosotros deseamos modificar un formulario que esta siendo usado en un proceso de workflow, no podemos hacerlo sobre el mismo. Las instancias de workflow que estén ejecutando deben continuar con la definición de formulario que comenzaron. Para resolver este problema los formularios y elementos tienen la capacidad de versionarlos (modificar los formularios futuros sin alterar los activos o históricos).

1.2 Tipos de datos

- Lógico:** El tipo de dato lógico o booleano es en computación aquel que puede representar valores de lógica binaria, esto es 2 valores (verdadero o falso).

Propiedades del tipo de datos lógico:

Control	Tipo de visualización para el tipo de datos lógico. Su valor por defecto es Casilla de verificación (Checkbox)
Etiqueta	Permite especificar si la etiqueta va a la izquierda o la derecha del checkbox.
Valor por defecto	False

- Caracter:** El tipo de caracter representa cualquier cadena de texto.

Propiedades del tipo de datos caracter:

Control	Tipo de visualización para el tipo de datos caracter. Su valor por defecto es Caja de texto. Los controles validos para este tipo de datos son: <ul style="list-style-type: none"> • Caja de texto • Área de texto • Editor embebido • Etiqueta • Combo box • Combo box dinámico • Caja de texto con autocompletado
Largo	Cantidad máxima de caracteres aceptados para este campo.
Requerido	Si el campo es requerido o no.
Valor por defecto	Texto vacío.

- 3. Fecha:** El tipo de fecha representa a cualquier fecha dentro del sistema, el mismo no incluye información sobre la hora, la misma la podemos pensar conceptualmente como una terna año, mes y día.

Propiedades del tipo de datos fecha:

Control	<p>Tipo de visualización para el tipo de datos fecha.</p> <p>Su control por defecto es un control para selección de fechas, el mismo permite seleccionar las fechas mediante un calendario o digitarlas directamente en el área de texto formateada.</p> <p>Los controles validos para este tipo de datos son:</p> <ul style="list-style-type: none"> • Selección de fechas tipo calendario • Selección de fechas no visual (caja de texto con formato)
Valor por defecto	Fecha vacía.

- 4. Fecha con hora:** El tipo de fecha con hora representa a cualquier campo de tipo fecha con hora dentro del sistema. La información de la hora guarda hora, minutos y segundos.

Propiedades del tipo de datos fecha con hora:

Control	<p>Tipo de visualización para el tipo de datos fecha con hora.</p> <p>Su control por defecto un control para selección de fecha y hora.</p> <p>El control para la selección de las fechas con hora permite seleccionar las fechas (con el mismo control que para las fechas sin hora) y agregar la hora, minutos y segundos mediante combo boxes.</p>
Valor por defecto	Fecha y hora vacías.

- 5. Imagen:** El tipo de imagen nos permitirá mostrar imágenes dentro de un formulario dinámico, un ejemplo de esto puede ser mostrar la foto de una determinada persona.

Propiedades del tipo de datos imagen:

Control	<p>Tipo de visualización para el tipo de datos imagen.</p> <p>Este tipo de datos soporta un único control, el mismo muestra la imagen o permite seleccionarla si estamos en modo edición.</p>
Alto	Alto para mostrar la imagen.
Ancho	Ancho para mostrar la imagen.
Texto alternativo	Texto alternativo a mostrar cuando no tenemos disponible la imagen.

- 6. Grupo:** El tipo grupo nos permite agrupar otros elementos. Las agrupaciones pueden ser lógicas o para formar grillas.

Propiedades del tipo de datos grupo:

Control	Tipo de visualización para el tipo de datos grupo. Su valor por defecto es grupo (agrupación lógica de elementos). Los controles validos para este tipo de datos son: <ul style="list-style-type: none"> • Grupo (agrupación lógica de elementos) • Grilla de elementos.
Colpasable	Indica si la agrupación lógica de elementos es colapsable.
Colpasado	Indica si la agrupación lógica de elementos es debe comenzar colapsada o no.

- 7. Enumerado:** El tipo de datos enumerado representa a un valor seleccionable de una lista de posibles valores.

Propiedades del tipo de datos enumerado:

Control	Tipo de visualización para el tipo de datos enumerado. Su valor por defecto es Combo box. Los controles validos para este tipo de datos son: <ul style="list-style-type: none"> • Combo box • Combo box dinámico • Radio buttons
Valores estáticos	Lista de los posibles valores del enumerado.
Programa de carga	Programa para cargar los posibles valores del enumerado.

- 8. Numérico:** El tipo de datos numérico representa a valores numéricos ya sean enteros o decimales.

Propiedades del tipo de datos numérico:

Control	Tipo de visualización para el tipo de datos numérico. Su valor por defecto es Caja de texto que solo acepta números. Los controles validos para este tipo de datos son: <ul style="list-style-type: none"> • Caja de texto que solo acepta números • Control de deslizante (Slider) para la selección del valor numérico.
Largo	Cantidad máxima de dígitos aceptados para este campo.
Decimales	Cantidad máxima de decimales (números luego de la coma) aceptados para este campo. El valor por defecto de este valor es cero.

9. Color: El tipo de datos color representa a colores utilizados en el sistema.

Propiedades del tipo de datos color:

Control	Tipo de visualización para el tipo de datos color. Este tipo de datos soporta un único control, el mismo permite seleccionar gráficamente un color válido.
---------	---

10. Correo electrónico: El tipo de datos correo electrónico representa correos electrónicos válidos. El mismo es un tipo particular de cadena de caracteres, podemos verlo como una especialización del tipo carácter, por esta razón tiene todas las propiedades del mismo, por esta razón únicamente detallaremos las propiedades que varían del tipo de datos carácter.

Propiedades del tipo de datos correo electrónico:

Control	Tipo de visualización para el tipo de datos correo electrónico. Este tipo de datos soporta un único control, el mismo es una caja de texto con una expresión regular para aceptar únicamente correos electrónicos válidos.
---------	---

11. Contraseña: El tipo de datos contraseña representa contraseñas en el sistema. El mismo es un tipo particular de cadena de caracteres, podemos verlo como una especialización del tipo carácter, por esta razón tiene todas las propiedades del mismo, por esta razón únicamente detallaremos las propiedades que varían del tipo de datos carácter.

Propiedades del tipo de datos contraseña:

Control	Tipo de visualización para el tipo de contraseña. Este tipo de datos soporta un único control, el mismo es una caja de texto que enmascara los caracteres ingresados (no los muestra, en su lugar muestra asteriscos).
---------	---

12. Archivo: El tipo de datos archivos nos sirve para implementar descarga o subida de archivos.

Propiedades del tipo de datos archivo:

Control	Tipo de visualización para el tipo archivo. Este tipo de datos soporta un único control, el mismo permite bajar el archivo mediante un link, o elegir un archivo local para subirlo al servidor.
Alto	Alto para mostrar el campo.
Ancho	Ancho para mostrar el campo.
Envío automático	Flag que me indica si el archivo se debe enviar automáticamente o esperar a ser enviado con todo el formulario.
Extensiones permitidas	Extensiones permitidas para la subida de archivos.
Directorio	Directorio en el servidor donde se guardan los archivos.

13. Direcciones web: El tipo de datos direcciones web nos da la posibilidad de ingresar direcciones web.

Propiedades del tipo de datos dirección web:

Control	Tipo de visualización para el tipo de datos direcciones web. Este tipo de datos soporta un único control, el mismo es una caja de texto con una expresión regular para validar direcciones web.
---------	--

1.3 Componente de visualización/edición

- **Autocompletado:** El componente de visualización/edición con autocompletado es una caja de texto con autocompletado, lo que usualmente llamamos suggest.

Este componente se utiliza cuando se desea que el usuario a escribir en el cuadro de texto y el sistema le sugiera opciones a medida que el escribe. Básicamente se van refinando los resultados en función de lo que digita el usuario.

Una de las propiedades más interesantes del autocompletado en formularios es que un valor puede ser autocompletado dependiendo de los valores ingresados en campos anteriores. El ejemplo mas claro es la relación País-Ciudad, nosotros no queremos autocompletar con cualquier ciudad, queremos autocompletar con las ciudades del País.

Propiedades del componente de visualización/edición autocomplete:

Programa de carga	<p>Programa que será llamado vía Ajax para obtener dinámicamente los valores que deben ser sugeridos.</p> <p>El programa recibe como parámetros los valores de los cuales depende + la cadena de caracteres que se lleva cargado hasta ahora.</p> <p>La respuesta de este programa es un JSON con el resultado que debe ser desplegado.</p>
Cantidad mínima de letras	<p>Cantidad de letras mínima a digitar antes de que se despliegue el resultado del autocompletado.</p> <p>El valor por defecto de esta propiedad es dos.</p>
Forzar selección	<p>True obliga a que se seleccione sí o sí un valor de la lista de elementos "sugeridos". False permite ingresar elementos nuevos (que no están en la lista).</p> <p>El valor default es true.</p>
Máxima cantidad de resultados	<p>Máxima cantidad de resultados a mostrar para seleccionar.</p> <p>El default es 30.</p>
Valor por defecto	<p>Descripción del valor por defecto.</p>
Lista de campos	<p>Lista de campos de los que depende este campo.</p> <p>En el ejemplo de País-Ciudad, Ciudad depende de País.</p>

- **Combo box:** El componente de visualización/edición combo box es un combo box con valores estáticos.

Por combo box con valores estáticos entendemos que los mismos pueden ser cargados desde una lista de valores en caso de un enumerado o por medio de un programa desde cualquier fuente de datos en tiempo de instanciación del formulario. Dicho en otras palabras tenemos flexibilidad para definir que elementos va a tener el combo box, pero los mismos se cargan cuando carga la pagina que tiene la instancia del el formulario. En otras palabras los elementos se cargan en la respuesta del servidor y no vía Ajax.

Propiedades del componente de visualización/edición combo box:

Programa de carga	<p>Programa (procedimiento o data provider GeneXus) que será llamado al instanciar el formulario para obtener los posibles valores del combo box.</p> <p>El programa de carga no recibe parámetros.</p> <p>La respuesta de este programa es un SDT GeneXus con el resultado que debe ser desplegado.</p>
-------------------	--

- **Combo box dinámico:** El componente de visualización/edición combo box dinámico es un combo box donde sus valores son obtenidos dinámicamente en tiempo de ejecución vía Ajax.

Una de las propiedades más interesantes del combo box dinámico en formularios es que los posibles valores del mismo pueden ser cargados dependiendo de los valores ingresados en campos anteriores. El ejemplo mas claro es la relación País-Ciudad, nosotros no queremos cargar como opción todas las ciudades del mundo, queremos sugerir como opción las ciudades del País.

Propiedades del componente de visualización/edición combo box dinámico:

Programa de carga	<p>Programa que será llamado vía Ajax para obtener dinámicamente los valores posibles del combo box dinámico.</p> <p>El programa recibe como parámetros los valores de los cuales depende.</p> <p>La respuesta de este programa es un JSON con el resultado que debe ser desplegado.</p>
Valor por defecto	Descripción del valor por defecto.
Lista de campos	<p>Lista de campos de los que depende este campo.</p> <p>En el ejemplo de País-Ciudad, Ciudad depende de País.</p>

- **Selección de fechas no visual:** Caja de texto con formato para seleccionar una fecha, este control no tiene propiedades adicionales.
- **Editor embebido:** Este control presenta el campo como una etiqueta, permitiendo por medio de botones habilitar su edición, la principal ventaja de este control es que nos permite confirmar la edición o cancelarla.
- **Etiqueta:** Esto control muestra el valor del campo pero no permite edición, es para utilizar el campo como una etiqueta en el formulario.
- **Combo de selección múltiple:** Este control nos permite seleccionar múltiples elementos de un combo box, se utiliza para seleccionar lista de elementos.

Un posible ejemplo de uso es elegir los hobbies de una persona.

Propiedades del componente de visualización/edición combo de selección múltiple:

Ordenable	Si es ordenable o no la lista del elementos seleccionados.
Usar botones	Si la interfaz para la selección sea por medio de botones agregados.

- **Radio button:** Este control muestra un conjunto de radio button para poder elegir el valor de un elemento de tipo de datos enumerado.
- **Control de deslizante (Slider):** Este control muestra una barra deslizante para elegir el valor de un elemento de tipo de datos numérico.

Propiedades del componente de visualización/edición slider:

Mínimo	Mínimo valor aceptado por el control deslizante.
Máximo	Máximo valor aceptado por el control deslizante.

- **Área de texto:** Este control muestra un área de texto para ingresar valores de elementos de tipo de datos caracter que puedan ocupar más de una línea (o fila).

Propiedades del componente de visualización/edición slider:

Filas	Cantidad de filas del área de texto.
Columnas	Cantidad de columnas del área de texto.

1.4 Tipos de Reglas

K2B Dynamic Form brinda la posibilidad de definir **reglas** en tiempo de ejecución. Las reglas pueden ser clasificadas en **reglas de visualización** o **reglas de negocio**.

1. **Visualización:** Las reglas de visualización son reglas que tienen como objetivo brindar una mejor experiencia para el usuario. Las mismas nos permiten modificar propiedades de los diferentes componentes que se usan para visualizar los campos.

Las reglas de visualización ejecutan acciones sobre un determinado campo en el cliente web; ejemplos de posibles acciones son: ocultar, mostrar, habilitar, deshabilitar y cargar un valor.

La condición de ejecución de una determinada regla se define por una expresión lógica dependiente de valores de ese campo u otros campos de la instancia del formulario.

2. **Negocio:** Las reglas de negocio tienen como objetivo permitir validar propiedades exigidas a los campos por la realidad.

Las reglas de negocio siempre van a ser ejecutadas en el servidor web, adicionalmente algunas de ellas también serán ejecutadas en el cliente web.

La duplicación de la ejecución de las reglas tiene dos objetivos:

- a) La ejecución del lado del servidor es obligatoria por razones de seguridad, las mejores prácticas sugeridas por **OWASP** dicen que debemos validar todas las entradas del usuario en el servidor.
- b) La ejecución opcional de algunas reglas en el cliente web nos permite hacer interfaces web más amigables.

Las reglas para emitir advertencias o errores son ejemplos de reglas de negocio que serán ejecutadas tanto en el cliente web como en el servidor.

1.5 Dominios de la OMA

El módulo de formularios dinámicos incorpora el concepto de dominio semántico. Durante el proyecto VUCE se definirán todos los elementos del diccionario de la OMA como dominios.

Se puede visualizar el listado diccionario de la OMA en la Sección: III Normas y recomendaciones internacionales.

Fue validado que el diccionario de datos de la OMA es soportado por el módulo de formularios dinámicos, esto posibilita un 100% de compatibilidad con la OMA. Por lo tanto el sistema VUCE seguirá los estándares de la OMA tomando en cuenta las normativas nacionales vigentes.

1.6 Impresión a PDF

El modulo de formularios dinámicos permite generar un PDF a partir del contenido de una instancia de formulario dinámico.

La conversión a PDF se hace mediante un objeto externo GeneXus el cual utiliza la librería [PD4ML](http://pd4ml.com/) (<http://pd4ml.com/>).

Ejemplo:

- Instancia del formulario a imprimir:

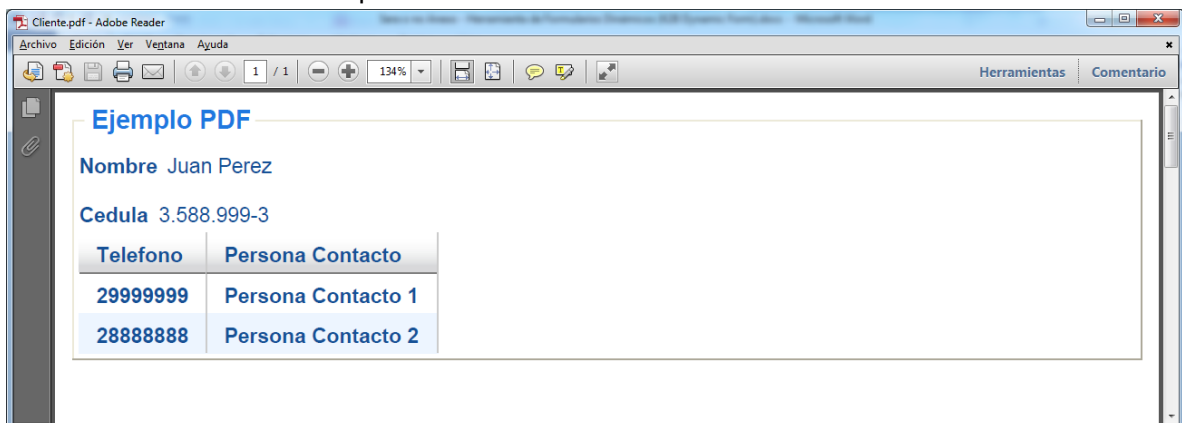
Ejemplo PDF

Nombre Juan Perez
Cedula 3.588.999-3

Contactos

Telefono	Persona Contacto
29999999	Persona Contacto 1
28888888	Persona Contacto 2

- PDF resultado de la impresión:



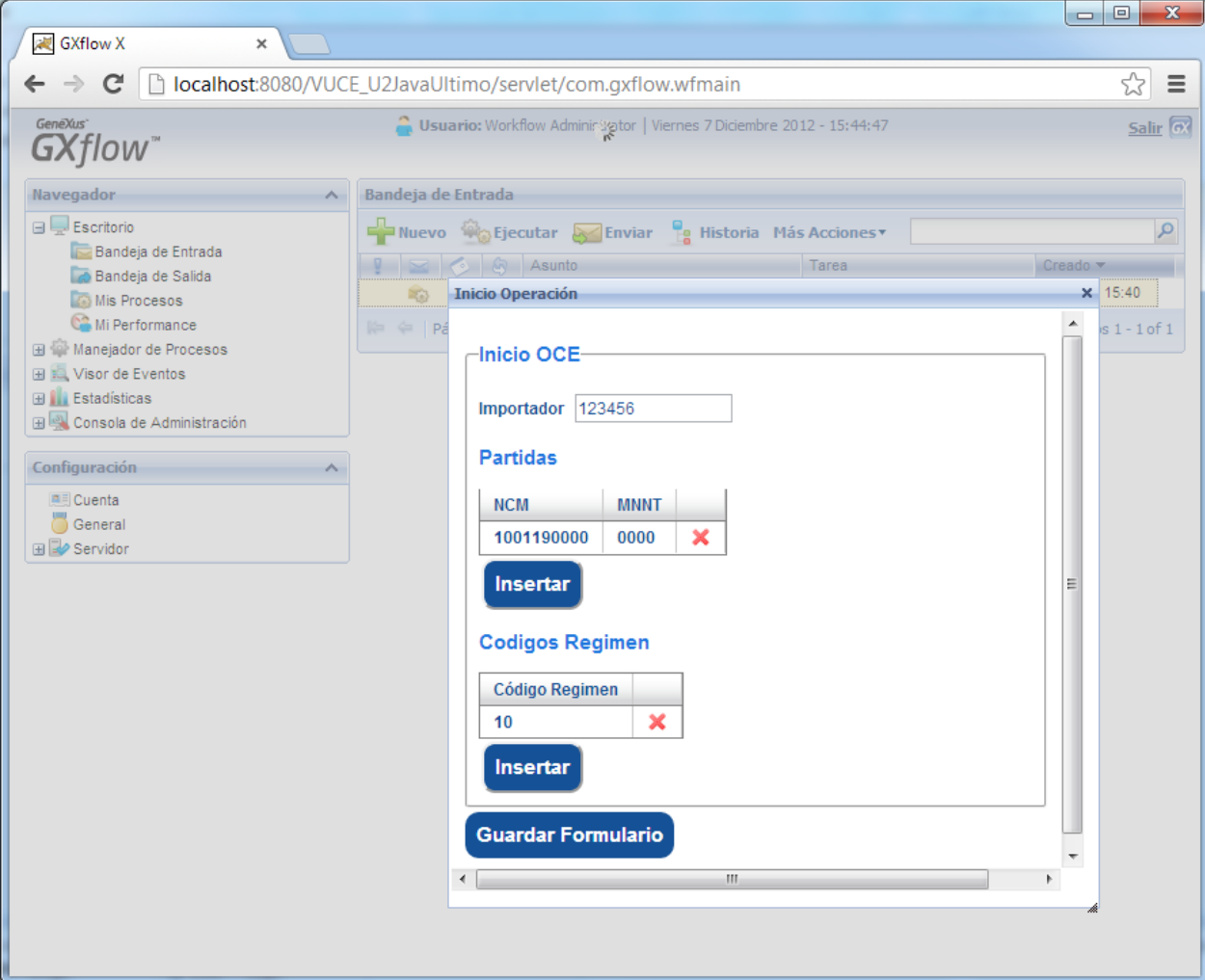
1.7 Exportación a XML

El modulo de formularios dinámicos provee un mecanismo de exportación de formularios a XML, dicho mecanismo es utilizado para obtener una representación estándar e independiente del contexto del formulario. Esta implementación será utilizada dentro de la exportación de OCE. Por más información ir a la Sección: III Interfaces.

1.8 Integración con Workflow

El módulo de formularios dinámicos está altamente integrado con Workflow, en diferentes pasos del proceso se invocan a instancias de formularios dinámicos para ingresar o desplegar datos.

A modo de ejemplo podemos ver el formulario dinámico utilizado para el comienzo de una OCE, la misma es invocada desde una tarea de Workflow:

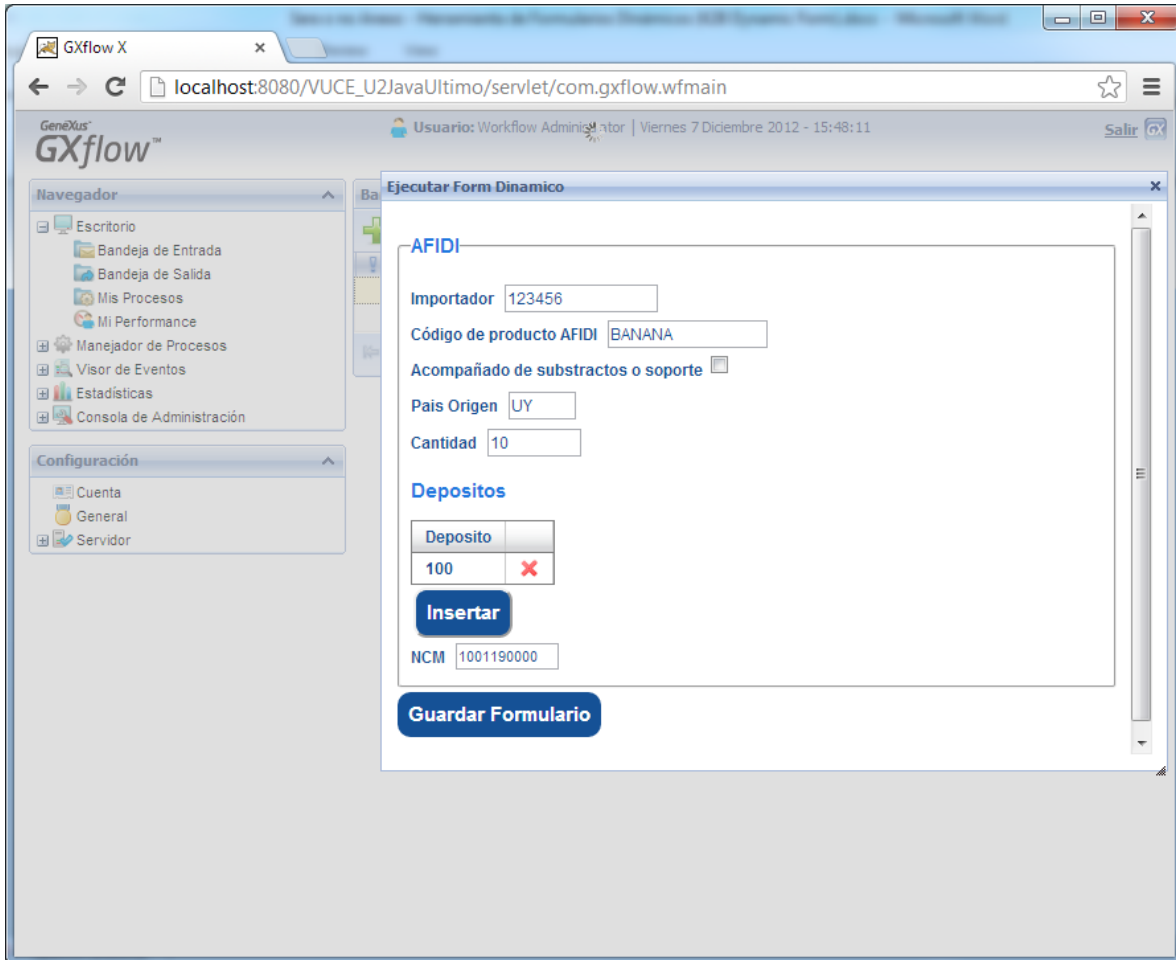


The screenshot shows the GXflow X web application interface. The browser address bar displays `localhost:8080/VUCE_U2JavaUltimo/servlet/com.gxflow.wfmain`. The user is logged in as 'Workflow Administrator' on Friday, December 7, 2012, at 15:44:47. The interface includes a navigation pane on the left with options like 'Escritorio', 'Bandeja de Entrada', and 'Configuración'. The main content area shows an 'Inicio Operación' task with a sub-form titled 'Inicio OCE'. This form contains the following elements:

- Importador:** A text input field containing the value '123456'.
- Partidas:** A table with two columns: 'NCM' and 'MNNT'.

NCM	MNNT
1001190000	0000
- Códigos Regimen:** A text input field containing the value '10'.
- Buttons:** 'Insertar' (blue), 'Guardar Formulario' (blue), and 'Salir' (blue).

Otro ejemplo puede ser la pantalla para el comienzo de trámite de AFIDI:



The screenshot shows a web browser window with the URL `localhost:8080/VUCE_U2JavaUltimo/servlet/com.gxflow.wfmain`. The application is titled "GeneXus GXflow™" and the user is logged in as "Workflow Administrator". The main content area displays a "Ejecutar Form Dinamico" window for "AFIDI". The form contains the following fields and elements:

- Importador:** Text input with value "123456".
- Código de producto AFIDI:** Text input with value "BANANA".
- Acompañado de substractos o soporte:** Checkable field, currently unchecked.
- Pais Origen:** Text input with value "UY".
- Cantidad:** Text input with value "10".
- Depositos:** A table with one row:

Deposito	
100	X
- Insertar:** A blue button.
- NCM:** Text input with value "1001190000".
- Guardar Formulario:** A large blue button at the bottom.

La configuración necesaria para esta integración fue detallada en el modulo núcleo del sistema.

Es importante destacar que si **subimos** un **archivo** en un formulario dinámico, dentro de una tarea de workflow, el mismo queda asociado al proceso y queda disponible en el **content administrator de workflow**.

1.9 Integración con GXScheduler

La herramienta **GXScheduler** será utilizada desde los formularios dinámicos, los formularios dinámicos invocaran al modulo de agenda el cual le brindara las funcionalidades necesarias para gestionar un calendario de eventos cronológicos.

Los eventos pueden ser vistos por día, semana o mes, así como también se pueden agregar, actualizar y eliminar directamente.