****

**Manual de Usuario del Editor de Modelos**

**Web Picture**

V 1.0

**Instrucciones de uso**

Este manual muestra de forma sencilla lo que se debe hacer para realizar modelos sintácticamente correctos en web picture, utilizando el framework EMF de eclipse como soporte. Para mayor información de este framework visite:

<http://projects.eclipse.org/projects/modeling.emf>

**Palabras Clave:**

* WP: Abreviación de la palabra "Web Picture", refiriendose al editor.
* EMF: Se refiere al framework de modelado de eclipse "Eclipse Modeling Framework".
* ecore: Término que hace referencia al metamodelo creado en el framework de eclipse.
* EClass: Se refiere a una clase del metamodelo ecore utilizado en EMF.
* Descriptor: Se refiere a el archivo que define las herramientas disponibles en la paleta y la visualización de las instancias del metamodelo.
* XMI: Se refiere a al archivo que contiene el modelo según la especificación XMI de EMF .

**Convenciones:**

* Cuando vea un cuadro parecido al siguiente.

***Texto importante a tener en cuenta.***

Indica que contiene información que usted debería tener muy en cuenta en caso de que necesite conocimiento adicional que este manual no le pueda brindar o bien, visitar otra sección de este.

* Cuando vea un cuadro parecido al siguiente.



**Texto de advertencia.**

Indica que contiene información crítica que si usted no tiene en cuenta, puede afectar de manera negativa las funcionalidades del programa.

Para cada funcionalidad del programa, se describirá brevemente lo que hace y después se ilustrará dicha funcionalidad a través de un ejemplo. Lo anterior se mostrará a partir de el editor "Simple Classes", ubicado en la carpeta examples/uml/simple classes

Si desea aprender rápidamente la herramienta, siga los pasos de la sección de ejemplo de cada ítem del índice en orden descendente.

***Este manual NO describe el proceso de creación de Ecore en eclipse necesario para definir los modelos. En caso de necesitarlo, por favor dirigirse a:***

***http://www.vogella.com/articles/EclipseEMF/article.html***

**Contenidos.**

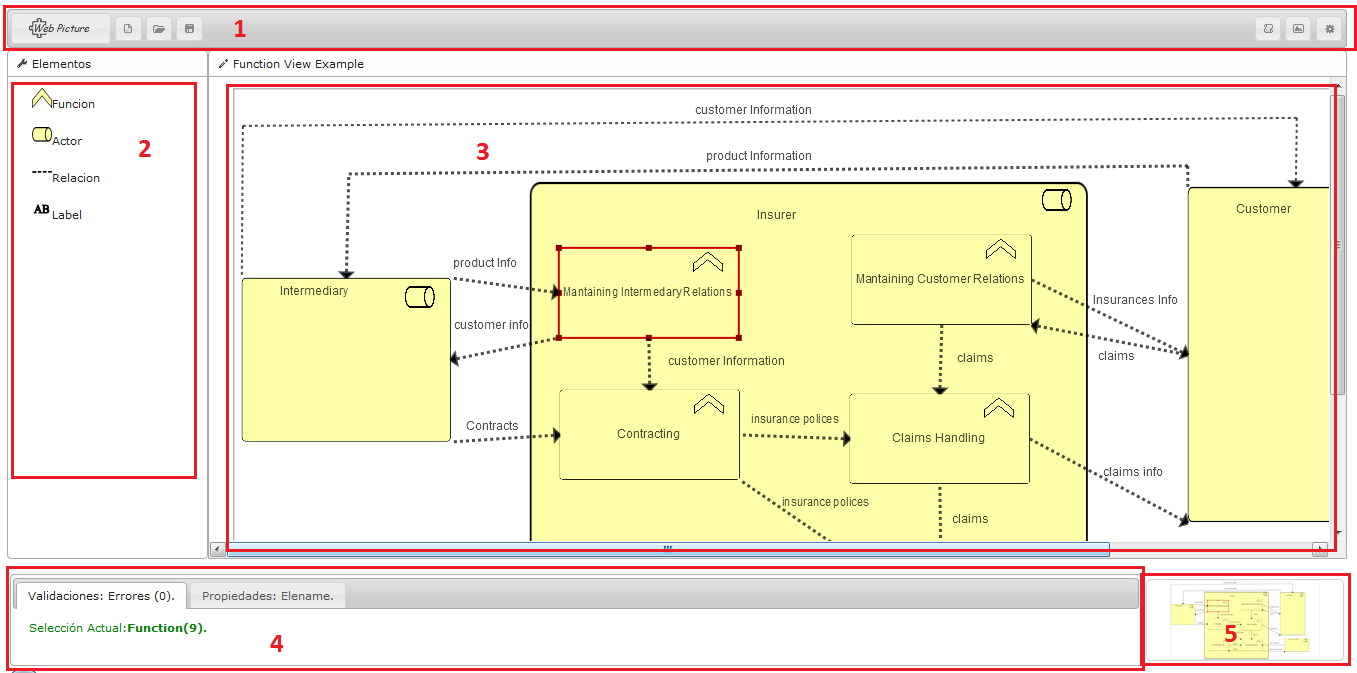
1. **Funciones Básicas de Web Picture.**
   1. Descripción de las secciones del editor.
   2. Cargar un Editor.
   3. Crear un nuevo Modelo.
   4. Abrir un Modelo.
   5. Guardar un Modelo.
   6. Exportar Modelo.
   7. Nombre y Tamaño del modelo.
   8. Edición del modelo.
      1. Agregar una instancia
      2. Doblar un conector.
      3. Cambiar un valor de atributo.
      4. Eliminar una instancia.
      5. Realizar Contenencia.
      6. Realizar Conexión.
2. **Definir el descriptor de Visualización.** 
   1. Representación gráfica.
   2. Validaciones Disponibles.

**3. Requerimientos del sistema.**

# Funciones Básicas de Web Picture.

**1.1. Descripción de las secciones del editor.**

WP se compone de 5 secciones que cumplen un grupo de funciones en especifico. A continuación se describen la funcionalidades que agrupa cada uno de estos.



**Secciones:**

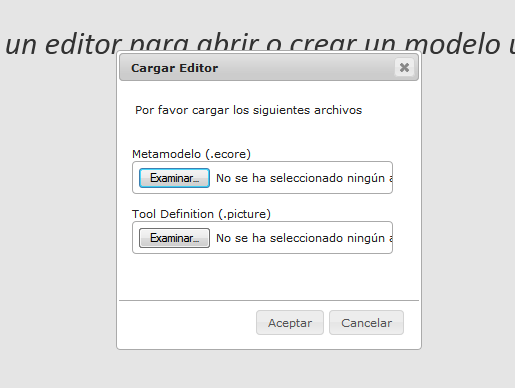
1. **Barra de Herramientas:** Esta sección agrupa las operaciones que WP permite realizar con los modelos creados en el mismo. Las funciones que agrupa, mencionadas de izquierda a derecha, son las de Cargar un editor, Crear nuevo Modelo, Abrir un Modelo, Guardar el modelo Actual, Exportar a XMI, Exportar a Imagen y Opciones.
2. **Paleta de Elementos:** En esta sección se carga la paleta definida en la sección <palette> del descriptor (previamente cargado en la 'Carga del editor'). A partir de éste se determinan las instancias de los elementos que el usuario puede agregar al modelo, a partir del ecore cargado.
3. **Editor del Modelo:** En esta sección se realiza la edición visual del modelo. Agrupa las funciones básicas que proporciona cualquier otro editor de diagramas convencional (cambio de tamaño, posición , conexión de conectores, contenencias).
4. **Información de validaciones y Edición de propiedades:** En esta sección se agrupan las funciones de edición detallada de atributos de las instancias (la pestaña propiedades) y la información de las validaciones del modelo con respecto al ecore (la pestaña validaciones).

***Para saber qué mensajes de validación se pueden observar, es necesario dirigirse al documento 'Documento de grado' ubicado en la carpeta 'docs'.***

1. **Previsualización del modelo:** Esta sección muestra la previsualización completa del modelo actual. Útil cuando los modelos son extremadamente grandes.

**1.2. Cargar un Editor.**

Para cargar un editor, haga clic en el botón  ubicado en la barra de herramientas. En seguida saldrá un dialogo (Mostrado abajo) indicando los archivos que debe cargar. Luego, dé clic en "Aceptar".

****



**Debe asegurarse que los archivos cargados en cada uno de los campos sean los correctos (que el descriptor coincida con el respectivo ecore) . WP no enviará ninguna advertencia de archivos que no coinciden o de otros tipos de archivos cargados.**



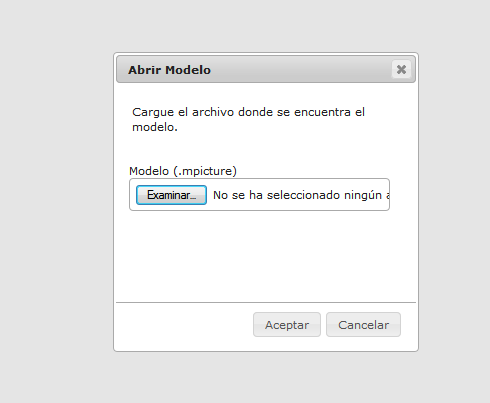
**WP Únicamente soporta ecore simples. Esto es, que tengan únicamente EClass sin extensiones de otras. Adicionalmente, WP no valida ni envía advertencias de ecore o descriptores mal formados. Asegúrese de que estos dos estén bien formados ya que WP puede cargar estos archivos de manera correcta aparentemente, causando a futuro mal funcionamiento.**

**1.3. Crear Nuevo Modelo.**

Para crear un nuevo modelo, haga clic en el botón  ubicado en la barra de herramientas. El modelo que esté editando actualmente se eliminará.

**1.4. Abrir un Modelo.**

Para abrir un modelo, haga clic en el botón  ubicado en la barra de herramientas. Luego, saldrá un dialogo, en el cual debe cargar el archivo mpicture que contiene el modelo. Finalmente, dé clic en aceptar. El modelo que esté editando actualmente se eliminará.

****

Después de pulsar aceptar, debe salir **después de unos segundos**, un aviso indicando que el modelo se ha cargado correctamente.



**Debe asegurarse que el contenido del archivo cargado sea de tipo mpicture. WP no enviará ninguna advertencia de archivos que no coinciden o de otros tipos de archivos cargados.**



**ANTES de cargar el modelo, es necesario haber cargado su editor respectivo. USTED debe asegurarse que el modelo cargado fue creado con el editor que antes debe cargar, ya que WP no hará ninguna validación al respecto.**

**1.5. Guardar un Modelo.**

Para guardar el modelo actual, haga clic en el botón  ubicado en la barra de herramientas. El archivo se guardará en formato .mpicture.



**Dependiendo de la configuración de su navegador, el archivo se guardará en la carpeta de descargas, o bien, saldrá el cuadro de diálogo de guardar.**

**1.6 Exportar un Modelo**

WP puede exportar los modelos a dos formatos diferentes: Imagen png y XMI (modelo de EMF).

***Para exportar a imagen:***

Haga clic en el botón  ubicado en la barra de herramientas. En seguida, se abrirá una nueva ventana de navegador, con la imagen incluida en ella.

***Para exportar a XMI:***

Haga clic en el botón  ubicado en la barra de herramientas. El archivo se guardará en formato .xmi. Para que la generación de este modelo funcione correctamente, es necesario que se tengan en cuenta las siguientes advertencias:



**Si se desea exportar a XMI, es NECESARIO que el ecore correspondiente cumpla las siguientes condiciones:**

* **EClasses deben tener un atributo llamado "id" de tipo EInt.**
* **Debe existir un EClass "Model" que contenga (haga referencias) a todas las demás EClasses del modelo.**

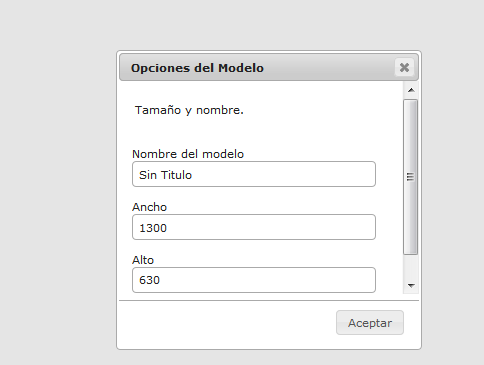
**Si no tiene en cuenta estas recomendaciones, la función de exportación podría arrojar resultados erróneos**.



**Dependiendo de la configuración de su navegador, el archivo se guardará en la carpeta de descargas, o bien, saldrá el cuadro de diálogo de guardar.**

**1.7. Nombre y Tamaño del modelo.**

Para configurar el nombre del modelo y su tamaño, haga clic en el botón  ubicado en la barra de herramientas. En seguida se abrirá un diálogo con los valores actuales de tamaño y nombre. Proceda a realizar sus cambios y luego haga clic en aceptar. Si no desea esos cambios, haga clic en cancelar.



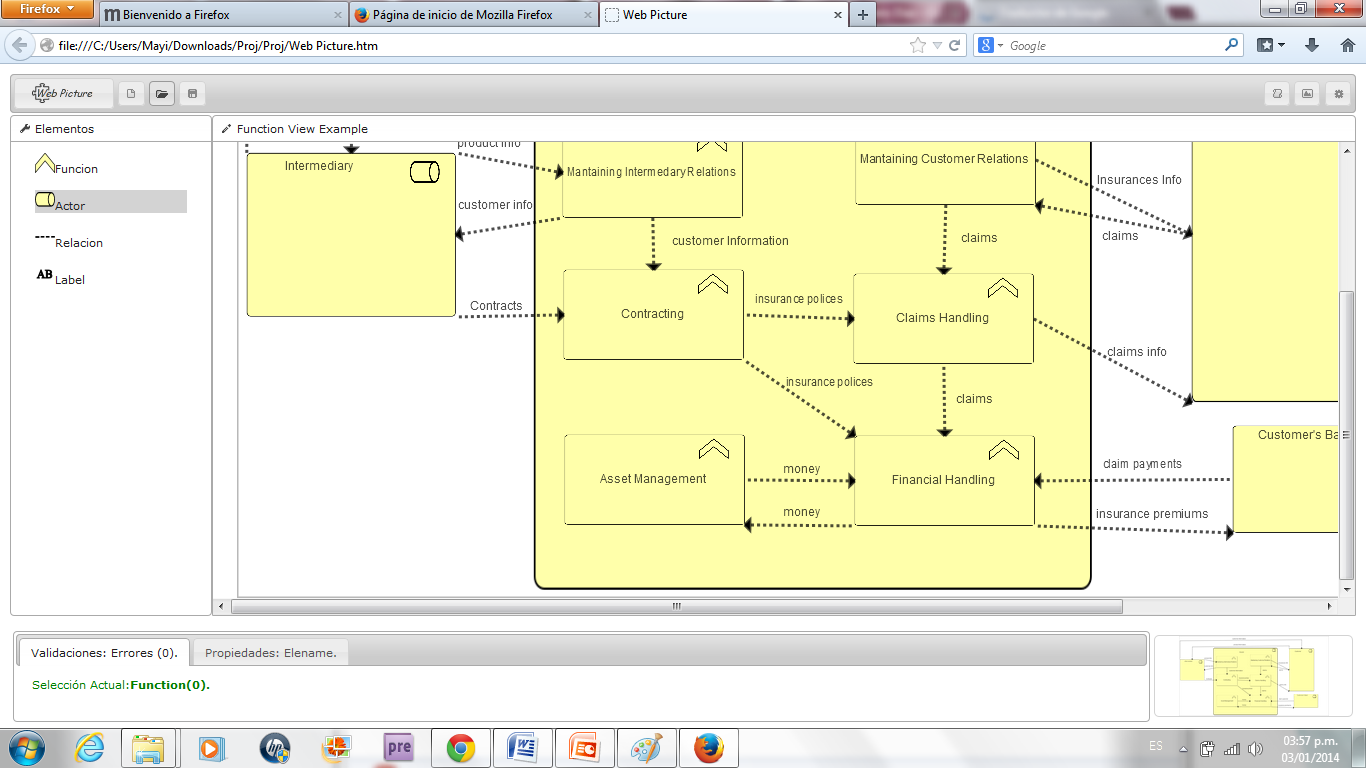


**Asegúrese que los valores de los campos coincidan con el tipo de dato. (los tamaños deben ser enteros, no nulos, y el nombre no debe ser nulo). WP no valida el formato de estos campos, por lo que datos mal asignados pueden causar comportamientos inesperados.**

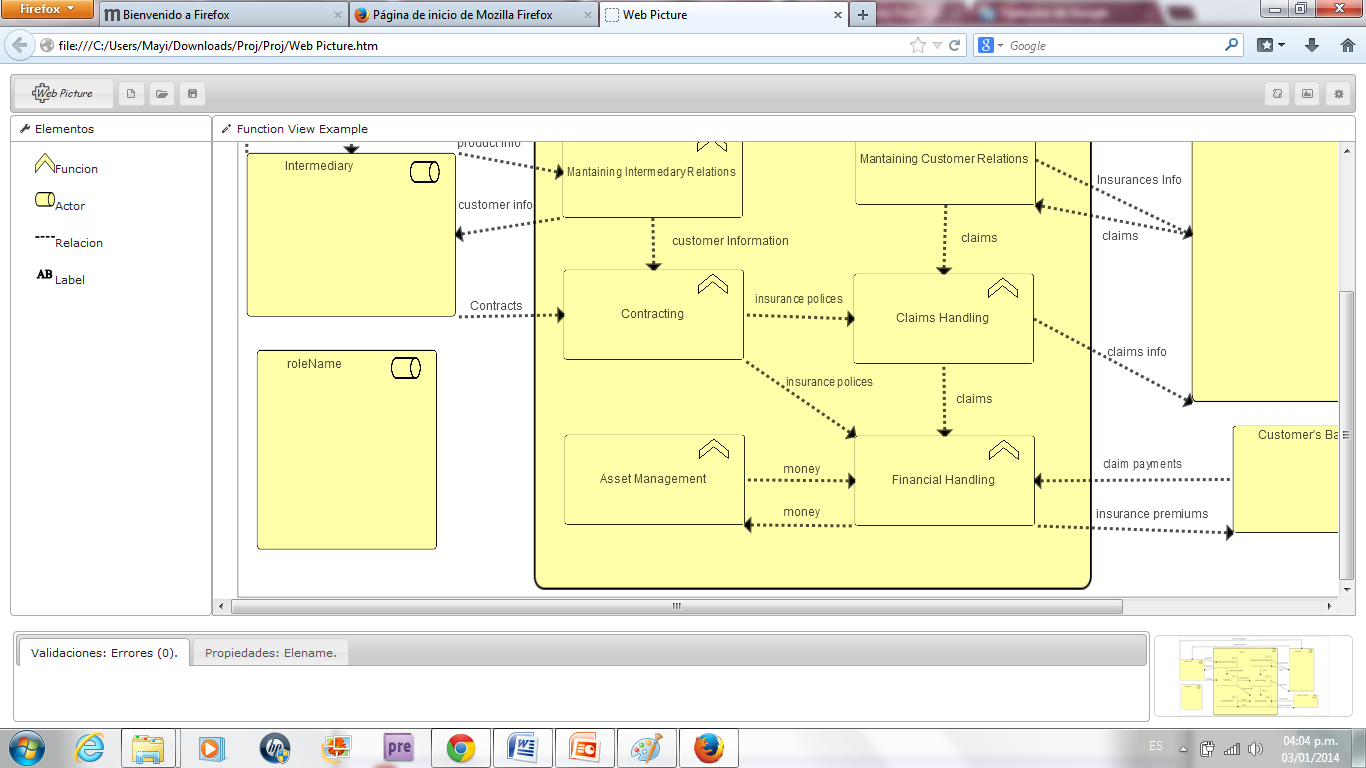
**1.8. Edición del modelo.**

**1.8.1. Agregar una Instancia**

Para agregar una nueva instancia al modelo, diríjase a la paleta de elementos y seleccione el elemento que desee agregar al modelo haciendo clic izquierdo sobre este. En este caso se agregará, otro elemento de tipo Actor.

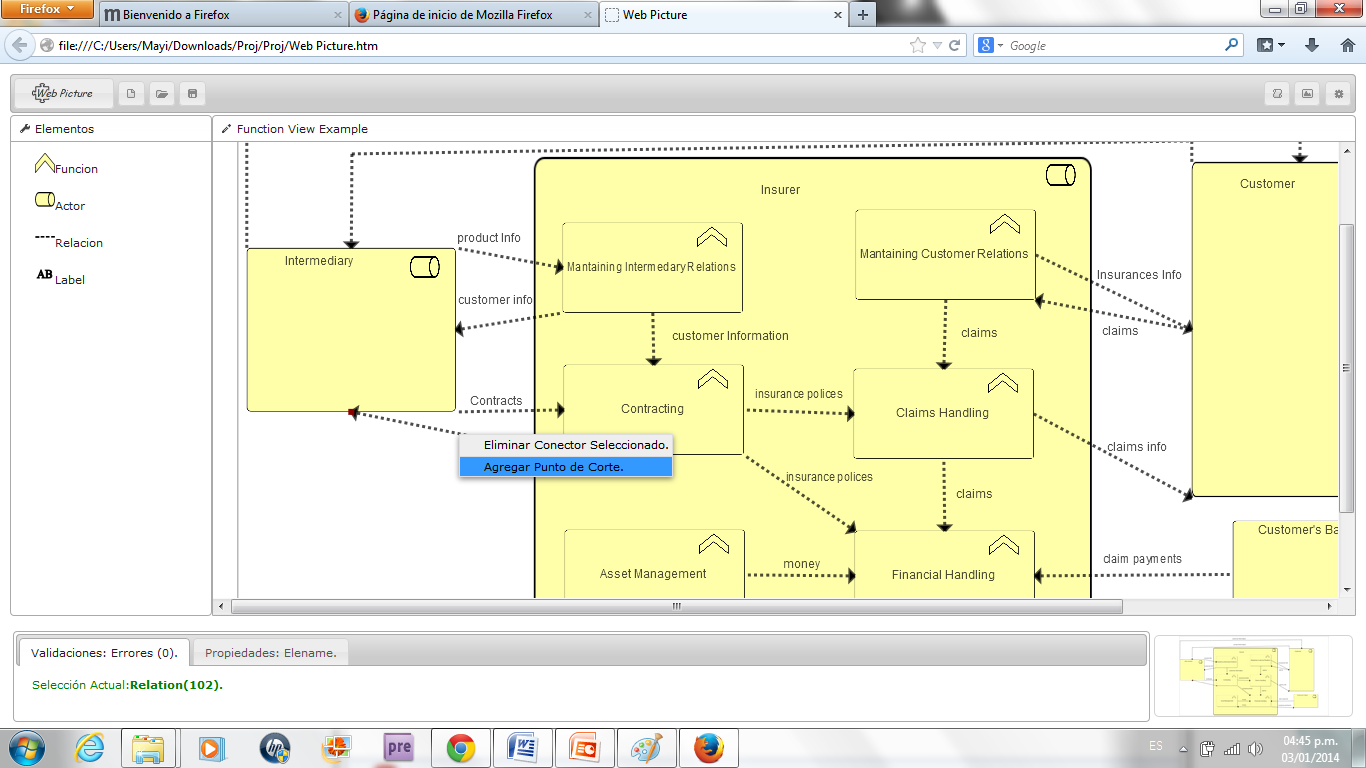
****

Una vez seleccionado diríjase a la sección de editor del modelo y haga clic izquierdo sobre donde desee ubicarlo.

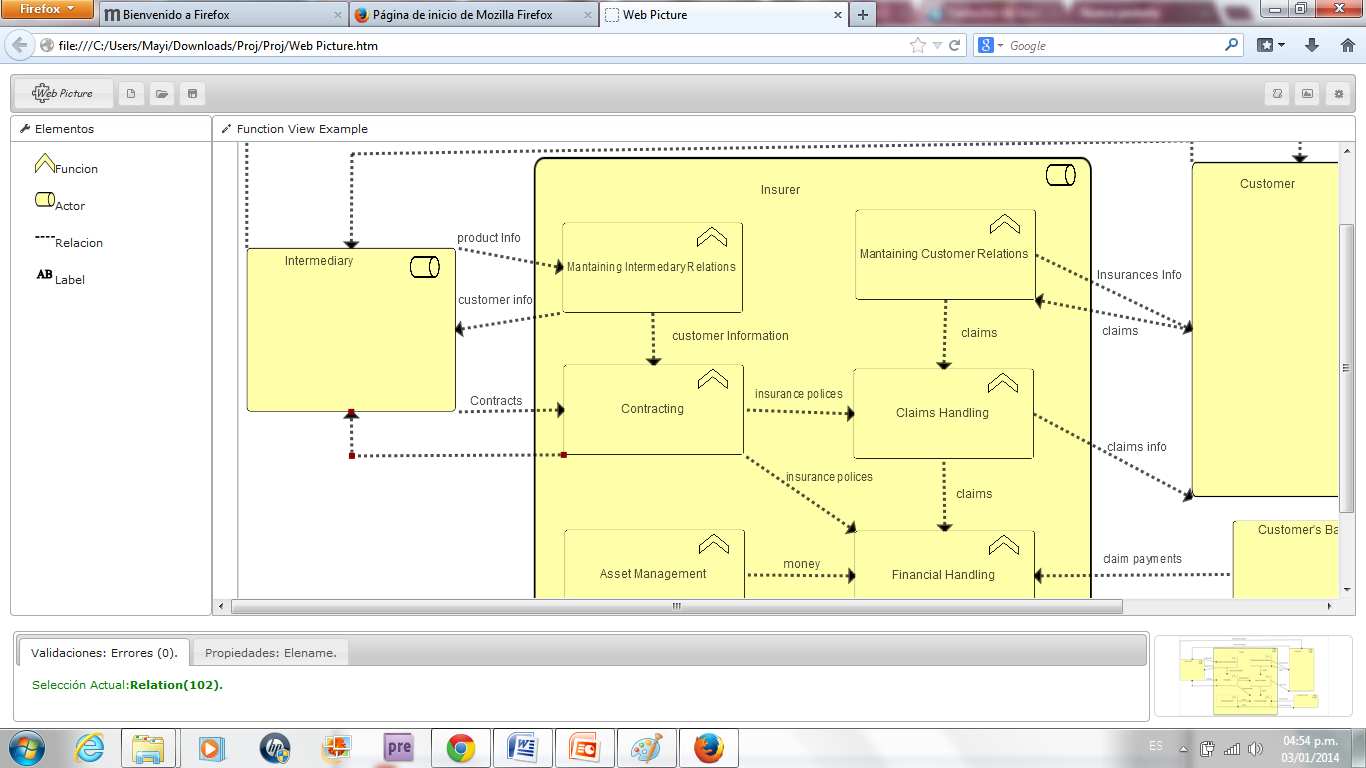
****

**1.8.2 Doblar un conector.**

Seleccione el conector que desea doblar haciendo clic sobre este (este se encontrara seleccionado si muestra en sus extremos unos cuadrados pequeños de color rojo) Una vez seleccionado haga clic derecho y seleccione la opción “Agregar Punto de Corte”

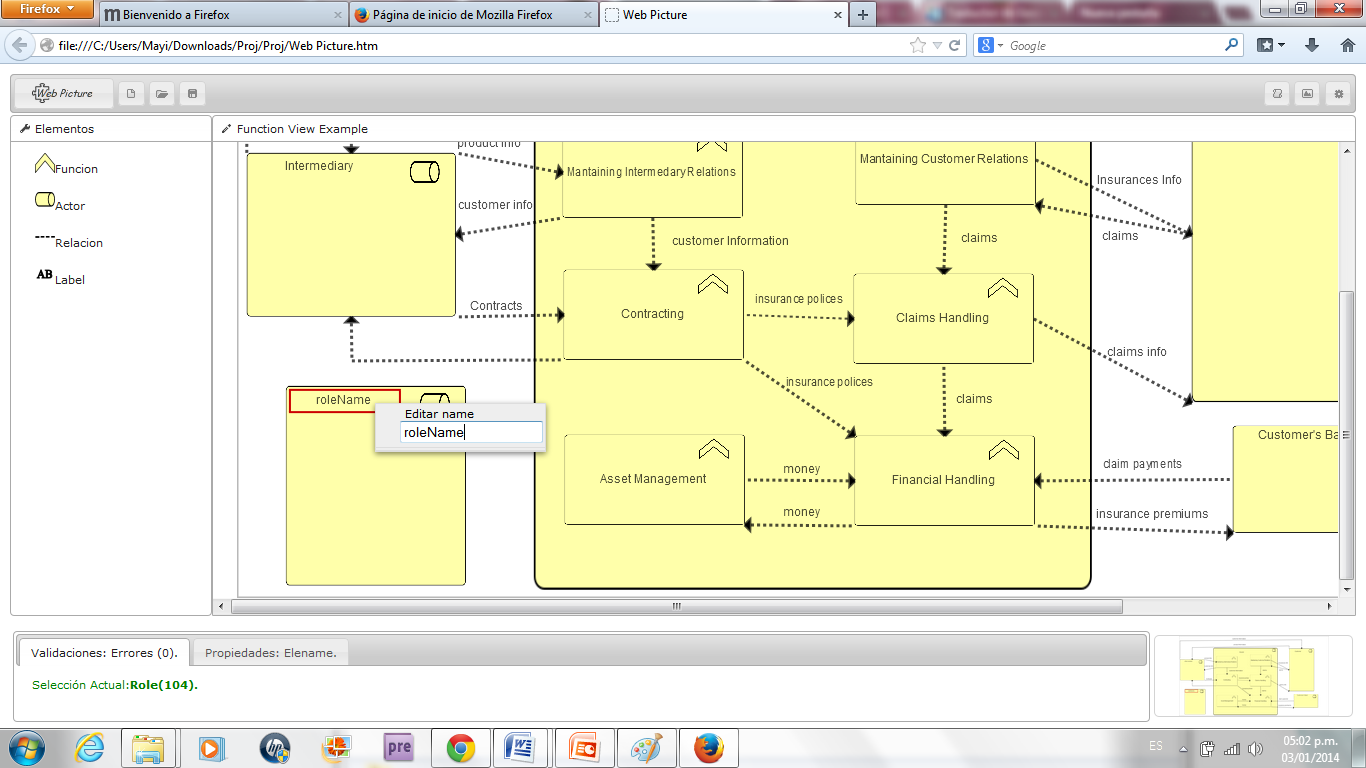
****

Se mostrará el conector con otro cuadrado de color rojo en la mitad desde el cual se podrá doblar el conector según se requiera. Para doblarlo, sitúese donde aparece este cuadrado y una vez el cursor cambie de forma y se transforme en una cruz, podrá acomodarlo a su disposición.

****

**1.8.3. Cambiar un valor de atributo.**

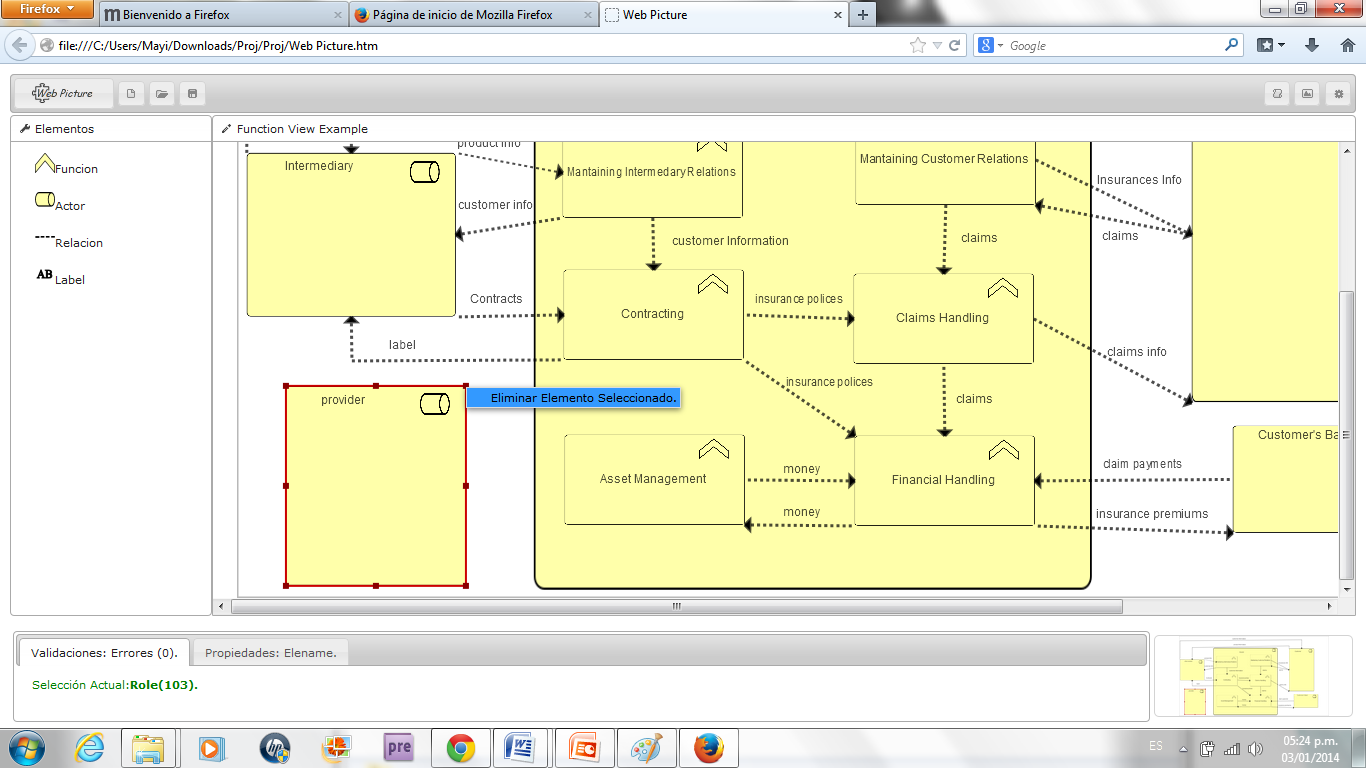
Para cambiar el valor de un atributo de un elemento haga clic sobre este para seleccionarlo (este se encontrara seleccionado si muestra un borde de color rojo). Al hacer clic derecho sobre este se mostrara un TextBox donde podrá modificar el texto según lo desee, una vez modificado, haga clic sobre cualquier parte del editor y el valor introducido quedará guardado automáticamente.

****

Para cambiar el valor de un atributo de una conexión (Que en este caso se referiría a un label), se realiza el mismo procedimiento. Este valor también lo puede cambiar en la pestaña de propiedades.

**1.8.4. Eliminar Una instancia.**

Para eliminar un elemento del editor, selecciónelo haciendo clic sobre él y luego haga clic derecho y seleccione la opción Eliminar Elemento Seleccionado.

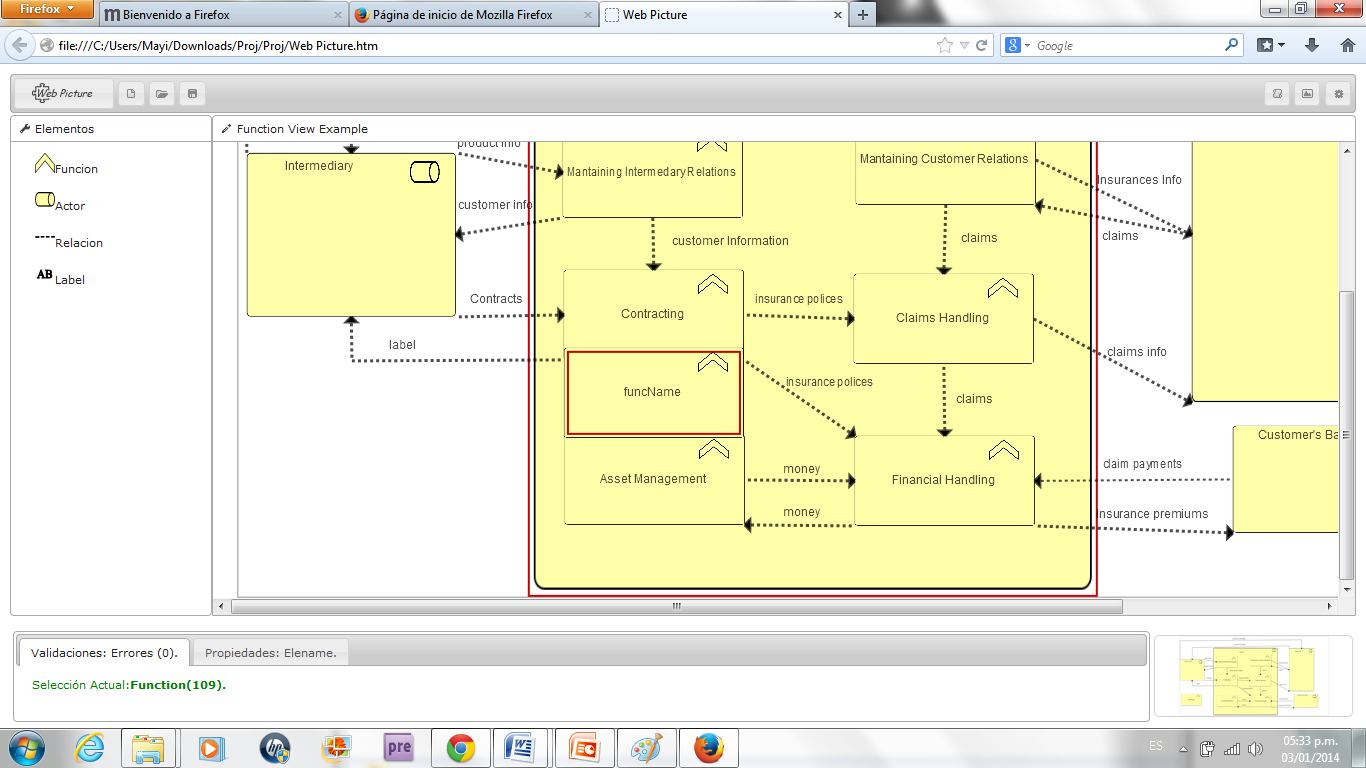
****



**Asegúrese que el elemento que desea eliminar este seleccionado correctamente. Para ello, note que el elemento debe mostrar un borde rojo con cuadrados pequeños en todos sus lados. Ubique el cursor en todo el borde de la figura para asegurar la selección correcta.**

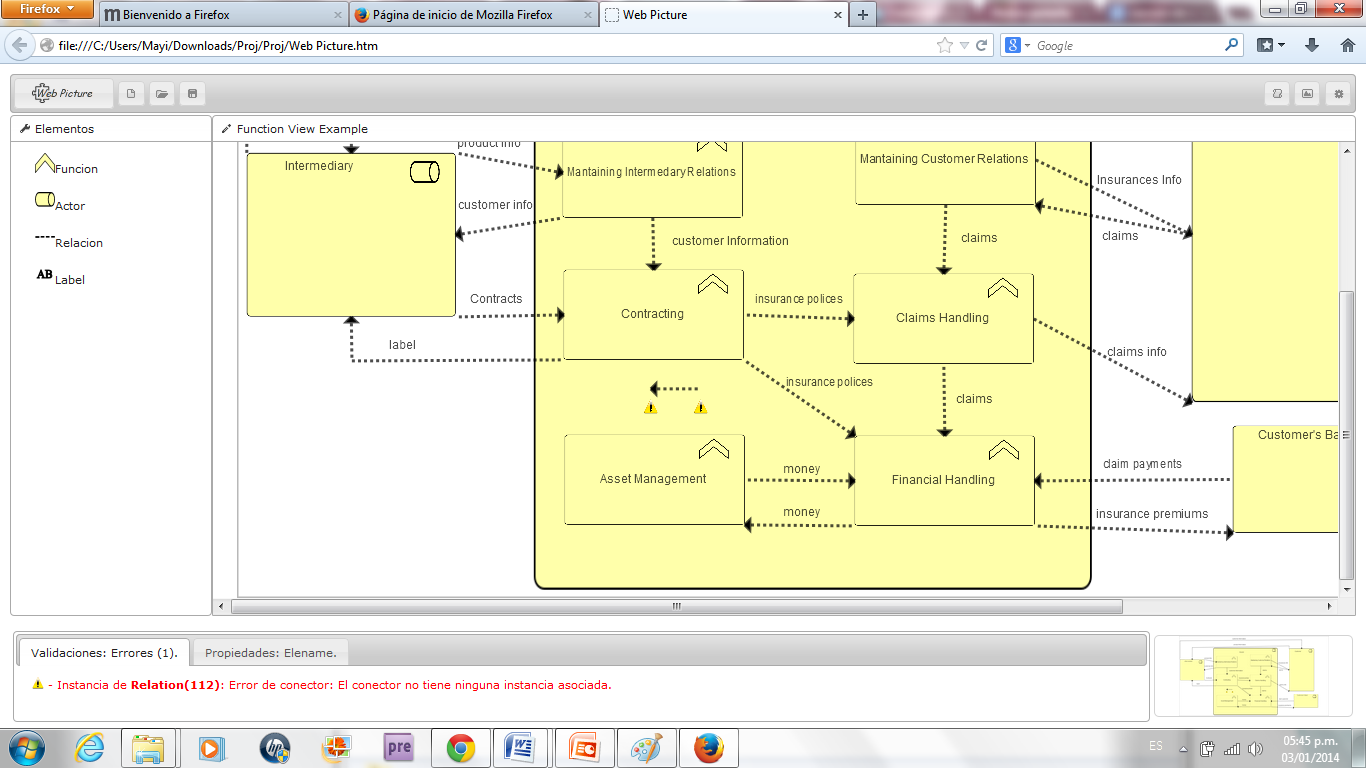
**1.8.5. Realizar Contenencia.**

Para realizar una contenencia de elementos seleccione el elemento que se desea incluir en otro y arrástrelo hasta la posición destino deseada (Es decir dentro del elemento destino). Note que una vez movido el elemento seleccionado hasta el otro elemento los dos tendrán un borde de color rojo indicando la contenencia. Una vez situado en la posición deseada, haga clic sobre cualquier parte del editor y la contenencia quedará automáticamente establecida.

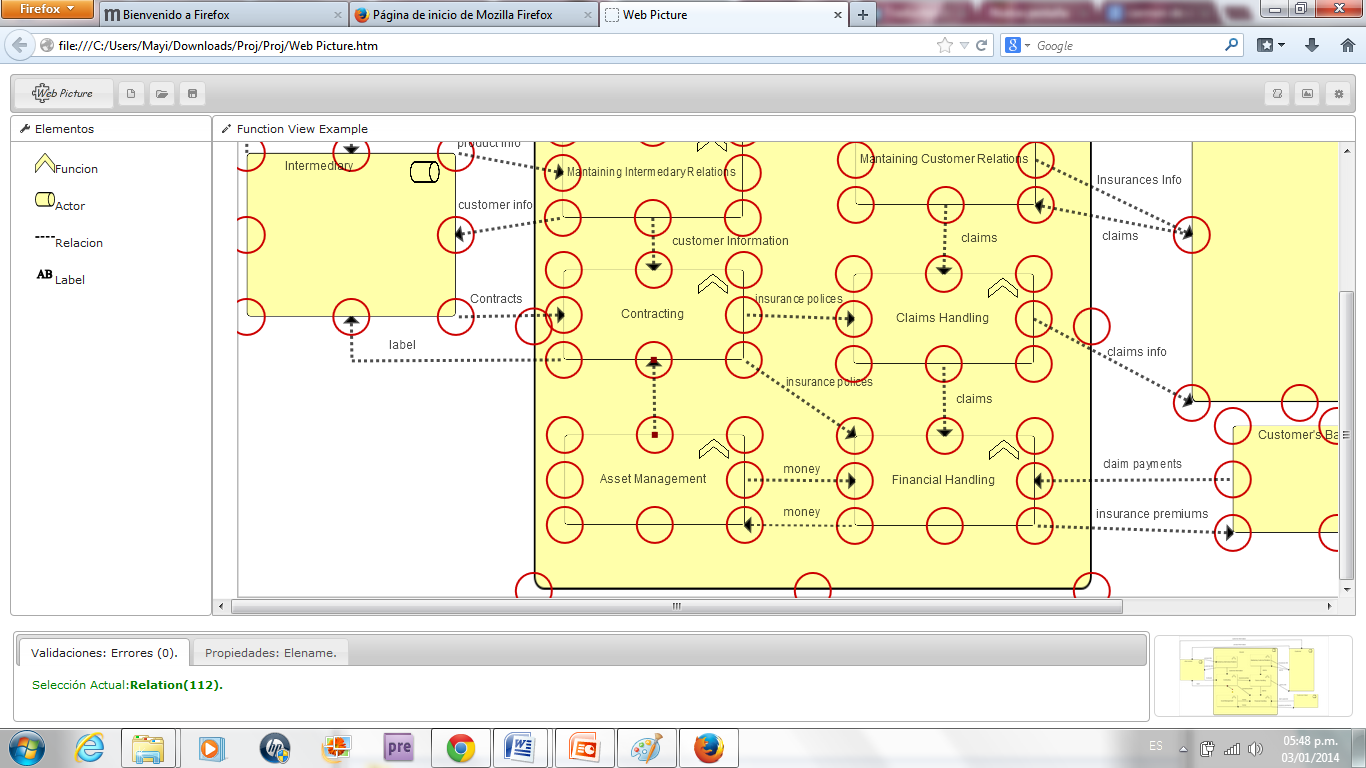
****

**1.8.6. Realizar Conexión.**

Para realizar una conexión entre dos elementos, seleccione la opción relación de la paleta de elementos y haga clic en la posición del editor del modelo donde la desee colocar. Una vez colocada, se mostrara una advertencia que indica que el conector añadido no posee origen ni destino asociados.



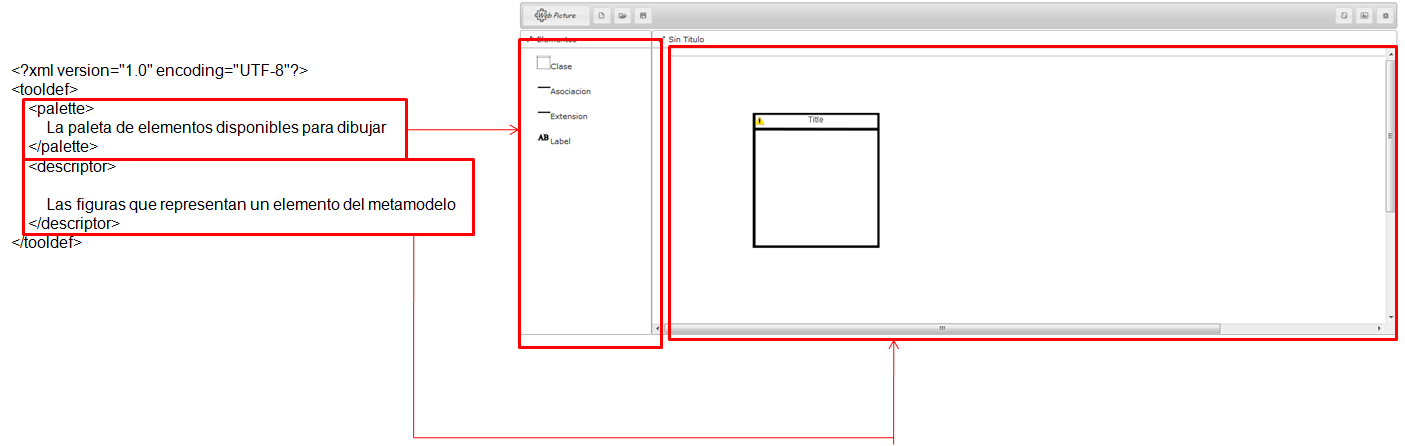
Para añadir su origen y destino, selecciónelo de tal manera que aparezcan cuadrados s rojos en sus extremos, sitúese sobre estos hasta que el cursor cambie de forma a una cruz y arrástrelos hasta el origen y el destino correspondientemente. Los origen y destino disponibles para ser utilizados se mostraran con círculos rojos en sus lados desde y hacia los cuales se podrá realizar la conexión.



# 2. Definir el descriptor de Visualización.

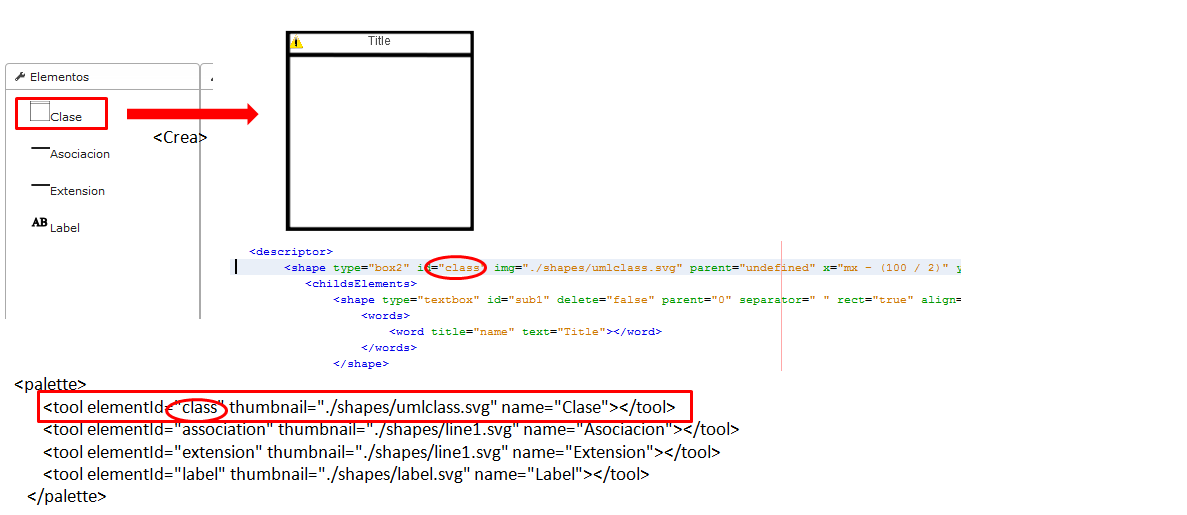
**2.1. Representación Gráfica.**

La representación gráfica de los modelos se basa esencialmente en el descriptor de visualización. El descriptor de visualización consiste en un archivo XML creado manualmente que contiene la definición de las figuras que representan un elemento de un metamodelo y la paleta disponible para definir los elementos. Este archivo posee la siguiente estructura:



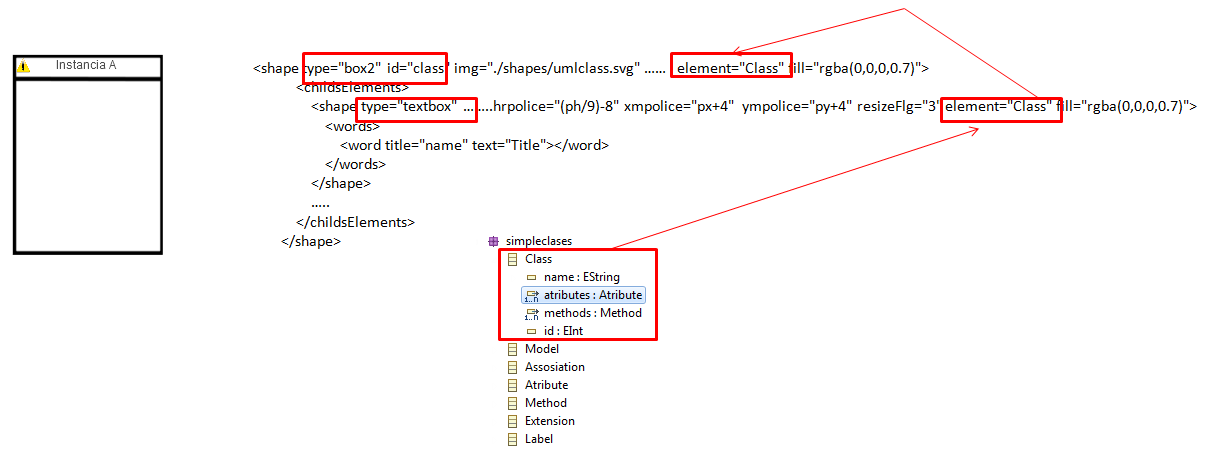
La paleta se define por una serie de tags con nombre "Tool", que posee las siguientes propiedades:

* **elementId:** El Id de la figura raíz existente en <descriptor> que esta herramienta va a dibujar.
* **thumbnail:** La imagen de muestra de la herramienta. Se recomienda utilizar imágenes relativas al directorio "shapes" de la herramienta.
* **name:** Texto que aparece en frente de la herramienta.



La definición de las figuras se realiza en la sección <descriptor> del xml. Una figura se define por el tag "shape". El <shape> raíz corresponde a la figura básica principal de representación de un elemento. Sus hijas, ubicadas en <childsElements> son otras <shape> que enriquecen a la principal. Las tres <shape> posibles en ese tag son Box2, FlowBox y Textbox. Solo es posible definir como raíz un shape de tipo Box2, y las demás como sus hijas. A continuación se muestran las propiedades básicas de definición de <shape>.

* **id:** El Identificador único de la figura (debe ser único incluso si es hija).
* **type:** El tipo de figura nativa que se va a definir. los valores posibles son box2,textbox,flowbox o link. Este atributo determina los otros posibles atributos del tag.
* **element:** Nombre del elemento del metamodelo al que hace referencia. Debe existir en el metamodelo.



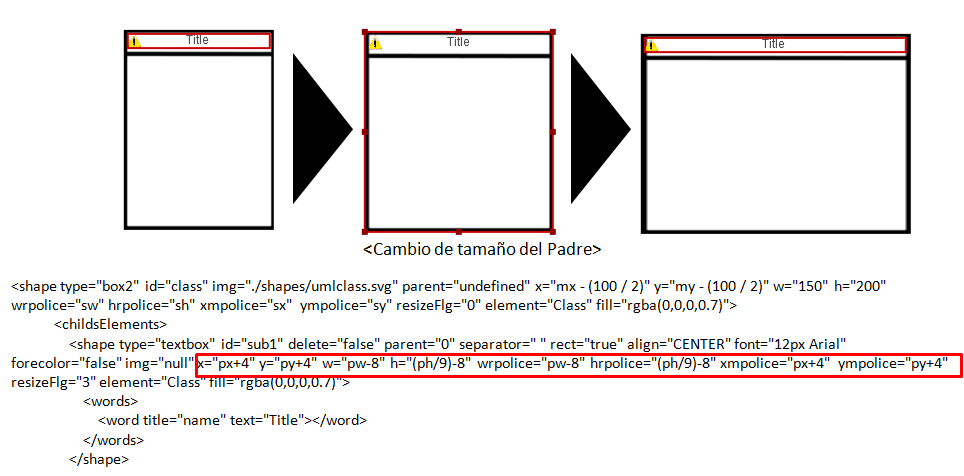
Cada figura <shape> básica posee propiedades de ubicación x, y, y tamaño w (ancho), h (alto), además de otras propiedades que hacen de cada figura única. El esquema de combinación de figuras básicas permite al usuario una alta flexibilidad a la hora de definir una forma de visualizar una instancia.

Una característica importante para la combinación de figuras es la alineación de estas respecto a un cambio de tamaño del padre. Por tal motivo, Diagramas web proporciona el uso de diez variables globales en la definición de posición y tamaño de las figuras nativas .

* **mx:** Obtiene la posición en x del cursor en el lienzo canvas.
* **my:** Obtiene la posición en y del cursor en el lienzo canvas.
* **sx:** Obtiene la posición en x de la figura básica que lo invoca.
* **sy:** Obtiene la posición en y de la figura básica que lo invoca.
* **sw :** Obtiene el ancho de la figura básica que lo invoca.
* **sy:** Obtiene el alto de la figura básica que lo invoca.
* **px:** Obtiene la posición en x de la figura básica padre que lo contiene.
* **py:** Obtiene la posición en y de la figura básica padre que lo contiene.
* **pw:** Obtiene el ancho de la figura básica padre que lo contiene.
* **ph:** Obtiene el alto de la figura básica padre que lo contiene.

Estas variables se pueden utilizar en las propiedades de x , y, w , h (disponibles en el Box2, FlowBox y Text Box) que definen la posición inicial de la figura al ser creada. Para manejar los eventos de cambio de tamaño y posición respecto al padre que los contiene, se manejan las siguiente propiedades adicionales.

* **xmpolice:** Indica la posición de la figura en x cuando su padre es movido/ cambiado de tamaño.
* **xypolice:** Indica la posición de la figura en y cuando su padre es movido/ cambiado de tamaño.
* **wrpolice:** Indica el ancho de la figura cuando su padre es movido/ cambiado de tamaño.
* **hrpolice:** Indica el alto de la figura cuando su padre es movido/ cambiado de tamaño.

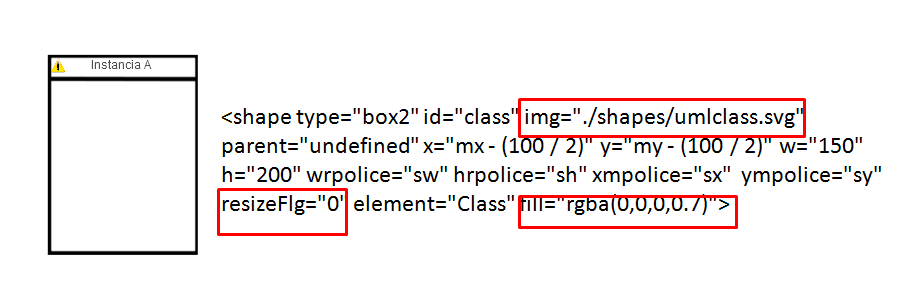
****

A continuación se explican las figuras nativas posibles y sus propiedades adicionales.

**Box2**

El Box 2 es la figura más básica del editor. Es un rectángulo, y es la única que permite interacción directa con el usuario (puede ser cambiada de tamaño y arrastrada. Posee las siguientes propiedades adicionales.

* **img:** La imagen fomrato svg de fondo. Se recomienda utilizar imágenes relativas al directorio "shapes" de la herramienta.
* **resizeflg:** 0 si la figura se puede cambiar de tamaño por el usuario, 1 si no se desea.
* **fill:** color de relleno de la figura en formato rgba(x,x,x,x).
* **parent:** debe ser siempre 0.



*Figura 11: Ejemplo definición Box2.*

**Flow Box**

El FlowBox es un rectángulo que cuya posición y tamaño son relativas a un Box2 que lo contenga como padre. Adicionalmente, éste puede contener textbox en su interior con un orden en particular, y pueden ser añadidos por un menú contextual siempre y cuando existan textbox en la raíz del descriptor, que representen elementos con referencias de contenencias válidas de este como padre. Las propiedades adicionales que presenta esta figura son las siguientes.

* **img:** La imagen fomrato svg de fondo. Se recomienda utilizar imágenes relativas al directorio "shapes" de la herramienta.
* **stmode:** 0 el orden de los textbox que se añaden es horizontal, 1 si el orden es vertical.
* **fill:** color de relleno de la figura en formato rgba(x,x,x,x).
* **parent:** debe ser siempre 0.



De estas maneras se pueden utilizar los FlowBox para contenidos de imagen estática o texto.

Nótese que para añadir instancias desde el menú contextual, es necesario que el atributo id del textbox ubicado en la raíz sea igual al nombre del elemento del metamodelo (el mismo valor del atributo "element" del textbox) que se va a contener.

**Text Box**

El textbox es la única figura nativa que soporta texto. Éste se encarga de representar los atributos de los elementos. El texto que contiene la figura se puede dividr por palabras que puden contener diferentes atributos, separados por un "token" que se define en las propiedades adicionales. Las propiedades son las siguientes.

* **img:** La imagen fomrato svg de fondo. Se recomienda utilizar imágenes relativas al directorio "shapes" de la herramienta.
* **delete:** Detemina si la figura es eliminable por el usuario (true o false).
* **separator:** detemina el "token" de separación de las palabras del texto con diferentes atributos.
* **rect:** determina si el cuadro se bordea con un rectángulo del color de la propiedad fill.
* **align:** CENTER,LEFT,RIGHT. Alineación del texto en el rectángulo de la figura.
* **forecolor:** color de fuente. Si n oes definido se utiliza el definido en fill.
* **resizeflg:** debe ser siempre 3.
* **parent:** debe ser siempre 0.
* **font:** tamaño y tipo de fuente. Ejemplo: 12px Arial.
* **stmode:** 0 el orden de los textbox que se añaden es horizontal, 1 si el orden es vertical.
* **fill:** color de relleno de la figura en formato rgba(x,x,x,x).

Esta figura posee en su interior el tag <words>. Dentro de éste es necesario definir al menos un tag <word>, que corresponde al mapeo atributo,valor del elemento al que representa dicho textbox. El tag <word> posee las siguientes propiedades.

* **title:** Nombre del atributo, que debe existir en el elemento definido en "element" del textbox.
* **text:** Valor inicial del atributo.



Nótese que el separador en la imagen anterior es un espacio.

**Link**

El Link es la única figura básica para realizar conectores. Esta posee la propiedad de conectarse únicamente con Box2 es sus extremos. Adicionalemente, puede doblarse hasta cinco veces. La propiedades adicionales que posee son las siguientes.

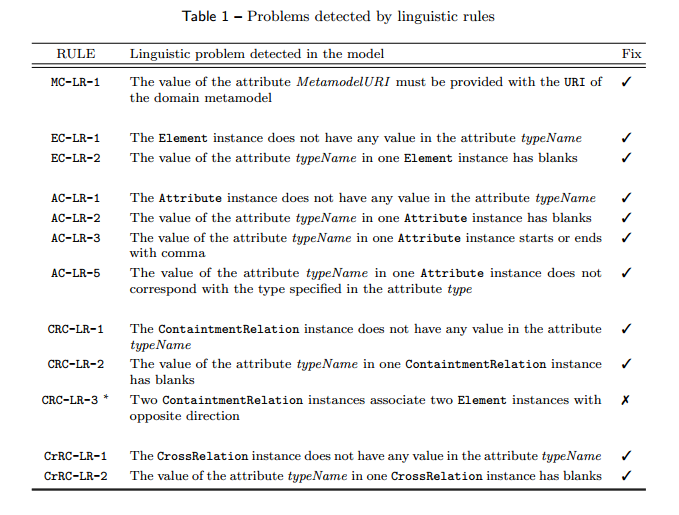
* **p1Rel - p2Rel:** Nombres de las referencias (separadas por comas y existentes en el metamodelo) que el conector tendrá en cuenta al momento de conectarse a otras instancias.
* **w:** determina el ancho de la línea.
* **arrowStyle:** Estilo de las puntas de la flecha. Valores de 1 a 10. 0 sin puntas.
* **arrowSize:** Tamaño de las cabezas de la flecha.
* **arrowColor:** Relleno de las cabezas de la flecha. poner "no" indica que las cabeceras se dibujan con una flecha sin relleno.
* **fill:** color de relleno de la línea rgba(x,x,x,x).

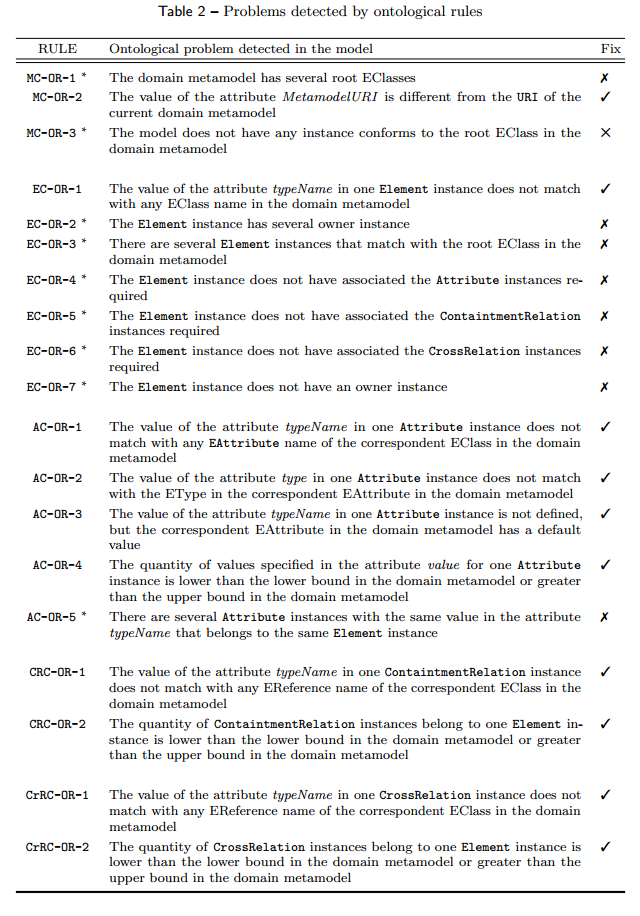


Nótese que en el interior de la figura se encuentran los tags <pone>, <ptwo> y <bendpoints>. Dentro de los tags <pone> y <ptwo> se definen los puntos iniciales al agregar el conector al modelo (tag <point x="posición en x" y="posición en y">) puede utilizar únicamente la variable mx y my. El tag <bendpoins> dejarlo en blanco.

**2.2. Validaciones Disponibles.**

Las reglas de validación lingüísticas y ontológicas [9] en las que se basa la herramienta para evaluar la sintaxis de los modelos son las siguientes.

****

****

La herramienta valida las reglas anteriores de la siguiente manera.

|  |  |  |
| --- | --- | --- |
| **Regla** | **Forma de Validar** | **Validada?** |
| **Lingüísticas** | | |
| MC-LR-1 | No aplica, no es el validador de EMF. | No |
| EC-LR-1 | Validado con el Descriptor grafico del Metamodelo (cuando es creado). | Si |
| EC-LR-2 | Si |
| AC-LR-1 | Si |
| AC-LR-2 | Si |
| AC-LR-3 | Si |
| AC-LR-5 | Si |
| CRC-LR-1 | Si |
| CRC-LR-2 | Si |
| CRC-LR-3 | No Contemplado en versión preliminar. | No |
| CrRC-LR-1 | Validado con el Descriptor grafico del Metamodelo (cuando es creado). | Si |
| CrRC-LR-2 | SI |
| **Ontológicas** | | |
| MC-OR-1 | No aplica- No Probada. | No |
| MC-OR-2 | No aplica, no es el validador de EMF. | No |
| MC-OR-3 | Validado cuando se crea el descriptor de figuras del metamodelo. | Si |
| EC-OR-1 | Si |
| EC-OR-2 | Si |
| EC-OR-3 | No aplica, no es el validador de EMF. | No |
| EC-OR-4 | Validado cuando se crea el descriptor de figuras del metamodelo. | Si |
| EC-OR-5 | Si |
| EC-OR-6 | Si |
| EC-OR-7 | No aplica para la versión preliminar del software. | No |
| AC-OR-1 | Validado por el descriptor de figuras del metamodelo. | Si |
| AC-OR-2 | Si |
| AC-OR-3 | Si |
| AC-OR-4 | No aplican restricciones de longitud del valor en la versión preliminar del software. | No |
| AC-OR-5 | Validado por el descriptor de figuras del metamodelo. | Si |
| CRC-OR-1 | Si |
| CRC-OR-2 | La aplicación web maneja las relaciones de manera distinta. En la versión preliminar se realizaran las siguientes validaciones: De contenencia, upper limit y lower limit, y de conectores creados como elementos (EClass con Ereference origen y destino). | Si |
| CRRC-OR-1 | Validado con el descriptor de figuras del metamodelo. | Si |
| CRRC-OR-2 | La aplicación web maneja las relaciones de manera distinta. En la versión preliminar se realizaran las siguientes validaciones: De contenencia, upper limit y lower limit, y de conectores creados como elementos (EClass con Ereference origen y destino). | Si |

Las validaciones de primera jerarquía[9] son validadas por el descriptor de la figura (cuando este se construye) lo cual hace que todas las validaciones programáticas, no sean dependientes.

# 3. Requerimientos del Sistema

Para tener un correcto funcionamiento del editor, se debe tener en cuenta lo siguiente:

1. Se debe utilizar la versión más reciente de Mozila firefox.
2. Memoria RAM mayor o igual a 2 Gb.



**La ejecución del editor únicamente es soportada por Firefox y Chrome como navegadores, actualmente no hay una versión estable para realizarlo con ningún otro navegador.**



**En el navegador google chrome, debido a la última actualización de seguridad, no es posible exportar imágenes ni visualizar la previsualización del modelo.**