



**CEIS**  
CENTRO DE ESTUDIOS DE  
INGENIERÍA DE SOFTWARE

## Facultad de Ingeniería y Ciencias Centro de Estudios de Ingeniería de Software



UNIVERSIDAD  
DE LA FRONTERA



**CEIS**  
CENTRO DE ESTUDIOS DE  
INGENIERÍA DE SOFTWARE

## EXTENSION BUILDER - RELACIONES



Este instructivo tiene como finalidad mostrar la forma de uso de los distintos tipos de relaciones en el modelo de clases utilizado en extensión builder.

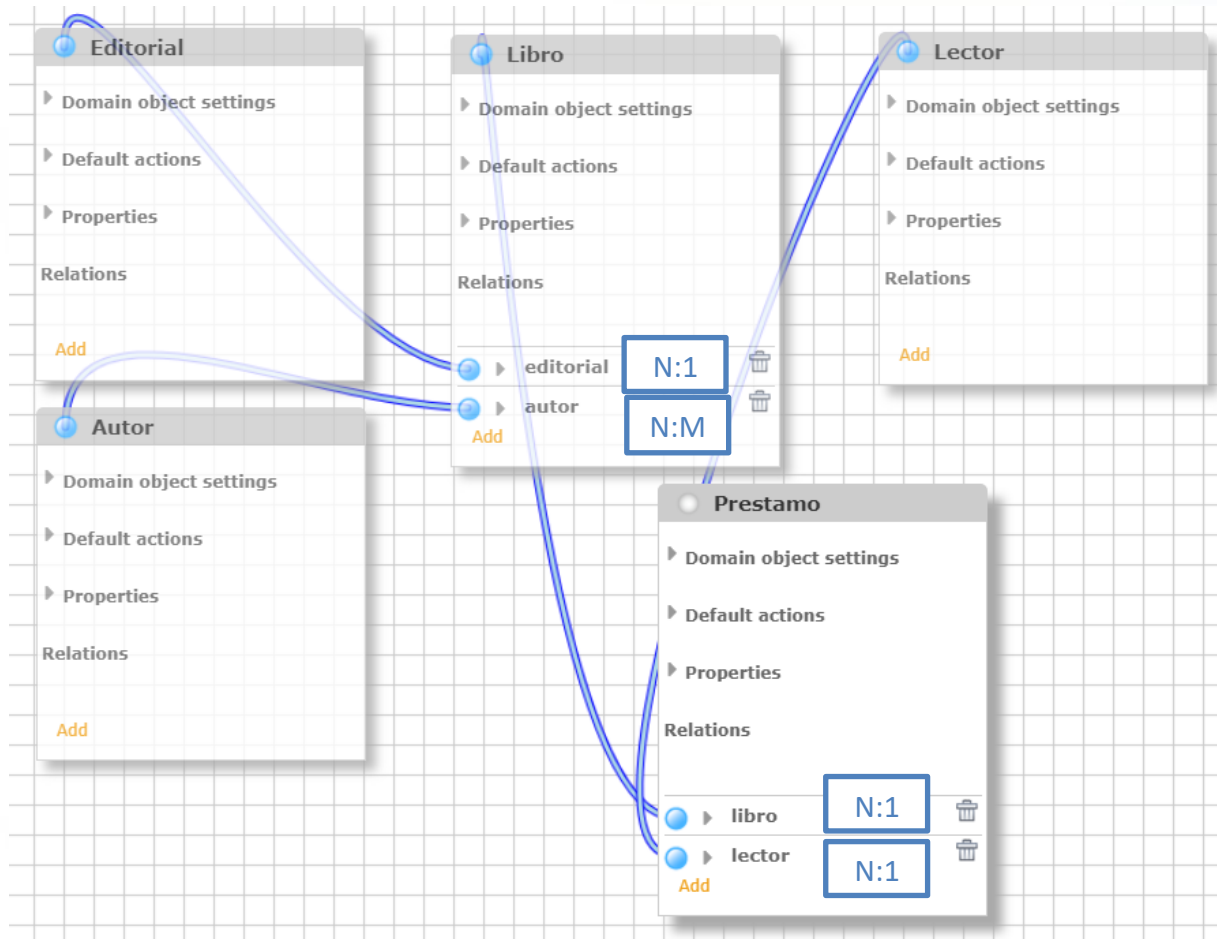
### CONTENIDOS

#### Construcción de Modelo y Plugin

- Paso 1: Crear modelo
- Paso 2: Agregar Mantenedores
- Paso 3: Modificar vistas

## Paso 1: Crear Modelo

Copia el modelo de la imagen y guárdalo



- 1:1 → Corresponde a la relación donde un elemento de una clase tiene solo una correspondencia con los elementos de la otra clase y viceversa.
- N:1 → Corresponde a la relación donde un elemento de una clase puede tener más de una correspondencia para un elemento de la otra clase.
- 1:N → Corresponde a la relación donde múltiples elementos de un objeto pueden tener correspondencia con un elemento de la otra clase.
- M:N → Corresponde a la relación donde múltiples elementos de una clase pueden tener correspondencia con múltiples elementos de la otra clase.



## Propiedades de las clases

**Lector**

- Domain object settings
- Default actions
- Properties
  - String : nombre
  - String: rut

Relations

Add

**Prestamo**

- Domain object settings
- Default actions
- Properties
  - Date: Inicio
  - Date: fin

Relations

libro	
lector	

Add

**Editorial**

- Domain object settings
- Default actions
- Properties
  - String: nombre

Relations

Add

**Autor**

- Domain object settings
- Default actions
- Properties
  - String: Nombre

Relations

Add

**Libro**

- Domain object settings
- Default actions
- Properties
  - String: nombre
  - Boolean: devuelto

Relations

editorial	
autor	

Add

**+** Recuerda Marcar todas las acciones "Default" de cada clase.

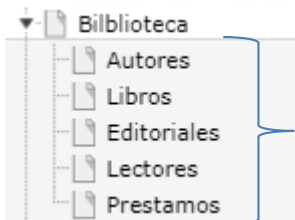
**+** No olvides generar los "Frontend Plugins" de cada una de las clases creadas.



## Paso 2: Agregar Mantenedores

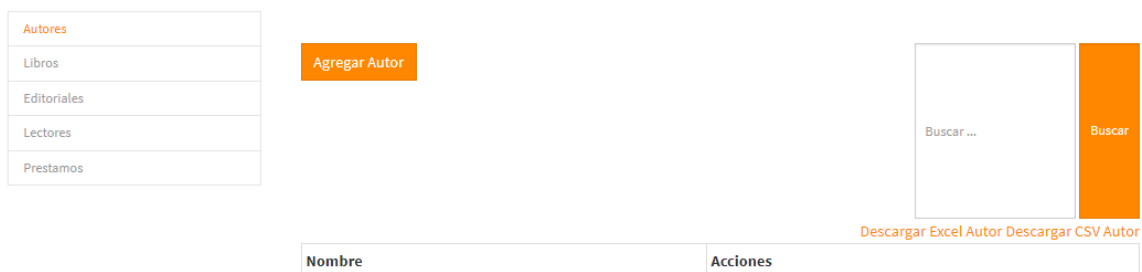
Nota: Recuerda que es necesario instalar la extensión creada para poder hacer uso de esta.

### Árbol de paginas



Crearás una pagina para cada clase creada en el modelo según la estructura de la Imagen.

Luego de agregar cada plugin a su respectiva pagina, se debería ver como la imagen siguiente. En caso de que no se visualice, borra el cache y recarga la pagina.



Hasta este punto podrás agregar y quitar datos de cada uno de los respectivos mantenedores



## Paso 3: Modificar Vistas

Será necesario acceder a los archivos de configuración de la extensión mediante algún editor de código.

`/var/www/html/rutaproyecto/typo3conf/ext`

Archivo de configuración de vista de formulario de edición de objetos.

Archivo de configuración de vista de lista en mantenedor

Archivo de configuración de vista para la creación de nuevos objetos

Archivo utilizado para la visualización de búsquedas personalizadas (no utilizado por defecto)

Archivo utilizado para la visualización de objetos (no utilizado por defecto).

Estructura de archivos de los plugin de frontend creados mediante extensionbuilder



## Modificar Template Lista

## Mejora a búsqueda

Agregar Editorial

Antartica

Buscar

Limpiar

Descargar Excel Editorial Descargar CSV Editorial

Nombre	Acciones
Antartica	<div style="display: flex; gap: 10px;"> <div style="border: 1px solid #ccc; padding: 2px 5px; display: flex; align-items: center;"> <span style="font-size: 1.2em;">✎</span> Editar         </div> <div style="background-color: #c0392b; color: white; padding: 2px 5px; display: flex; align-items: center;"> <span style="font-size: 1.2em;">✕</span> Eliminar         </div> </div>

```
<div class="form-group form-group-sm">
  <f:form.textfield name="searchTerm" placeholder="Buscar ..." class="form-control input-sm" value="{searchTerm}" />
  <f:form.submit value="Buscar" class="form-control btn btn-primary"/>
  <f:if condition="{searchTerm}">
    <f:form.submit value="Limpiar" class="form-control btn btn-primary" name="limpiar"/>
  </f:if>
</div>
```

Se elimina "input-sm"

Se elimina "form-control"

Luego de que se realizan las modificaciones, se visualizará como aparece en la siguiente imagen

Agregar Editorial

antartica

Buscar

Limpiar

Descargar Excel Editorial Descargar CSV Editorial

Nombre	Acciones
Antartica	<div style="display: flex; gap: 10px;"> <div style="border: 1px solid #ccc; padding: 2px 5px; display: flex; align-items: center;"> <span style="font-size: 1.2em;">✎</span> Editar         </div> <div style="background-color: #c0392b; color: white; padding: 2px 5px; display: flex; align-items: center;"> <span style="font-size: 1.2em;">✕</span> Eliminar         </div> </div>

```
<div class="form-group form-group-sm">
  <f:form.textfield name="searchTerm" placeholder="Buscar ..." class="" value="{searchTerm}" />
  <f:form.submit value="Buscar" class="btn btn-primary"/>
  <f:if condition="{searchTerm}">
    <f:form.submit value="Limpiar" class="btn btn-primary" name="limpiar"/>
  </f:if>
</div>
```



## Modificar Template Lista

## Reparar Vista de Relación M:N

Notarás que los mantenedores que contiene objetos **M:N** no se visualizan lanzando un error en su despliegue.

Autores
Libros
Editoriales
Lectores
Prestamos

Oops, an error occurred! Code: 2017100311102530e3ebb5

```
<td>{libro.nombre}</td>
<td><f:if condition="{libro.prestamo}"><f:then><span class="glyphicon glyphicon-ok">
<td>{libro.editorial.uid}</td>
<td>{libro.autor}</td>
```

N:M

Este objeto esta compuesto por múltiples autores, por lo que será necesario mostrarlos a todos por separado

```
<td>{libro.nombre}</td>
<td><f:if condition="{libro.prestamo}"><f:then><span class="glyphicon glyphicon-ok">
<td>{libro.editorial.uid}</td>
<td>
  <f:for each="{libro.autor}" as="autor">
    {autor.nombre}
  </f:for>
</td>
```

Despliegue de objeto autor modificado para listarlos.

Luego de guardar los cambios, se podrá visualizar de forma correcta el mantenedor

Agregar Libro

[Descargar Excel Libro](#) [Descargar CSV Libro](#)

Nombre	Prestamo	Editorial	Autor	Acciones
Notas de vida	✘	1	Leonardo Homero	<input type="button" value="Editar"/> <input type="button" value="✘ Eliminar"/>



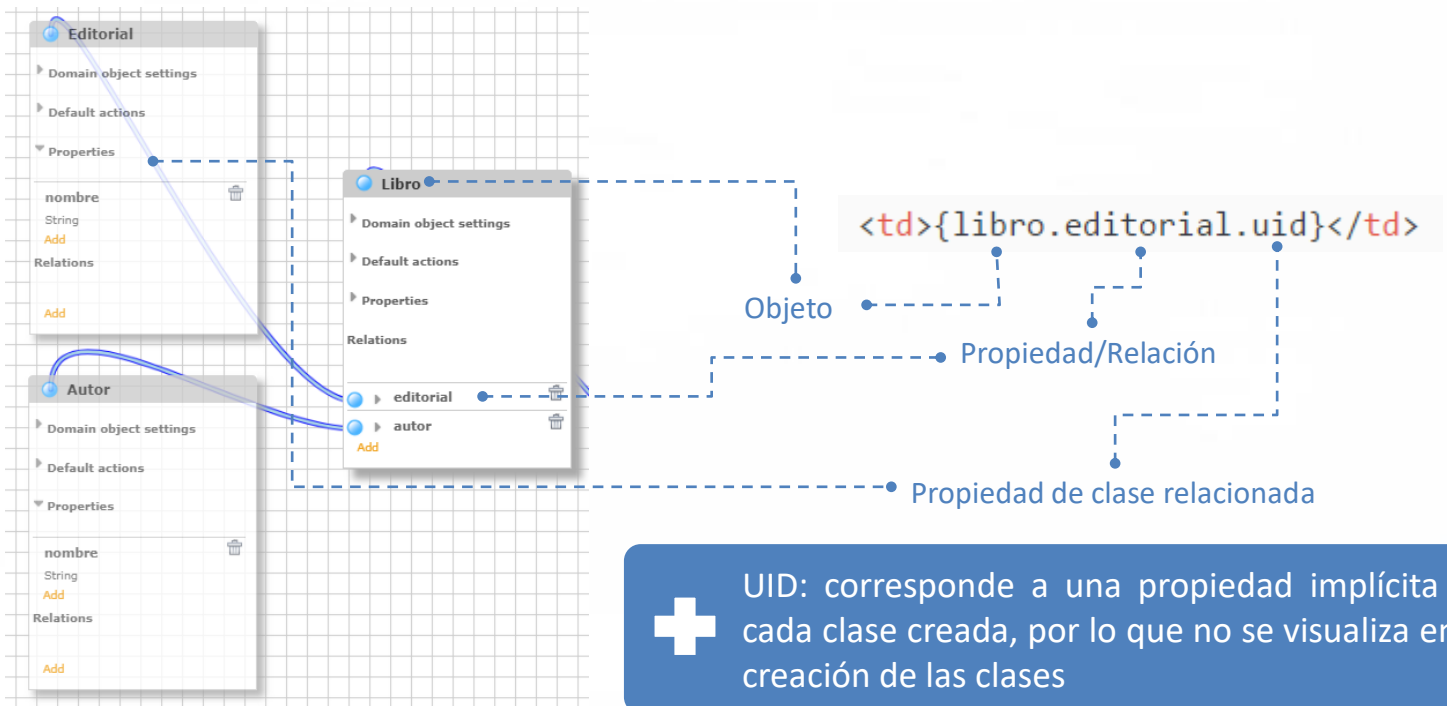
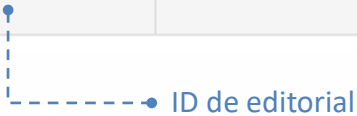


## Modificar Template Lista

### Reemplazar id con propiedad de objeto

Notarás que los mantenedores muestran solo el "ID" del atributo del objeto relacionado.

Nombre	Prestamo	Editorial	Autor	Acciones
Notas de vida	✘	1	Leonardo Homero	<input type="button" value="✎ Editar"/> <input type="button" value="✘ Eliminar"/>



**+** UID: corresponde a una propiedad implícita en cada clase creada, por lo que no se visualiza en la creación de las clases

Luego de modificar la propiedad del objeto a mostrar, se visualizara como la imagen a continuación

```
<td>{libro.editorial.nombre}</td>
```

Nombre	Prestamo	Editorial	Autor	Acciones
Notas de vida	✘	Andres Bello	Leonardo Homero	<input type="button" value="✎ Editar"/> <input type="button" value="✘ Eliminar"/>

**+** Esta modificación también es posible realizarla en otros archivos de template, como edit.html, show.html, new.html.