

# TYP03 CMS 7.3 – Qué hay Nuevo

Resumen de las nuevas características, cambios y mejoras

Creado por:

Patrick Lobacher y Michael Schams

Traducción en Español por:

Michel Mix y Sergio Catalá

17/June/2015

Creative Commons BY-NC-SA 3.0



# TYPO3 CMS 7.3 - Qué hay Nuevo

---

## Resumen de Capítulos

Introducción

Interfaz de Usuario de Backend

TSconfig & TypoScript

Cambios en Profundidad

Extbase & Fluid

Funciones Obsoletas/Eliminadas

Fuentes y Autores

## Introducción

## Los Hechos

# Introducción

---

## TYPO3 CMS 7.3 – Los Hechos

- Fecha de lanzamiento: 16 de Junio 2015
- Tipo de lanzamiento: "Lanzamiento Sprint"
- Visión: Adoptar, Innovar, Lanzar
- Foco principal: Ecosistema de Paquetes, Composer y Manejo de Extensiones

# Introducción

---

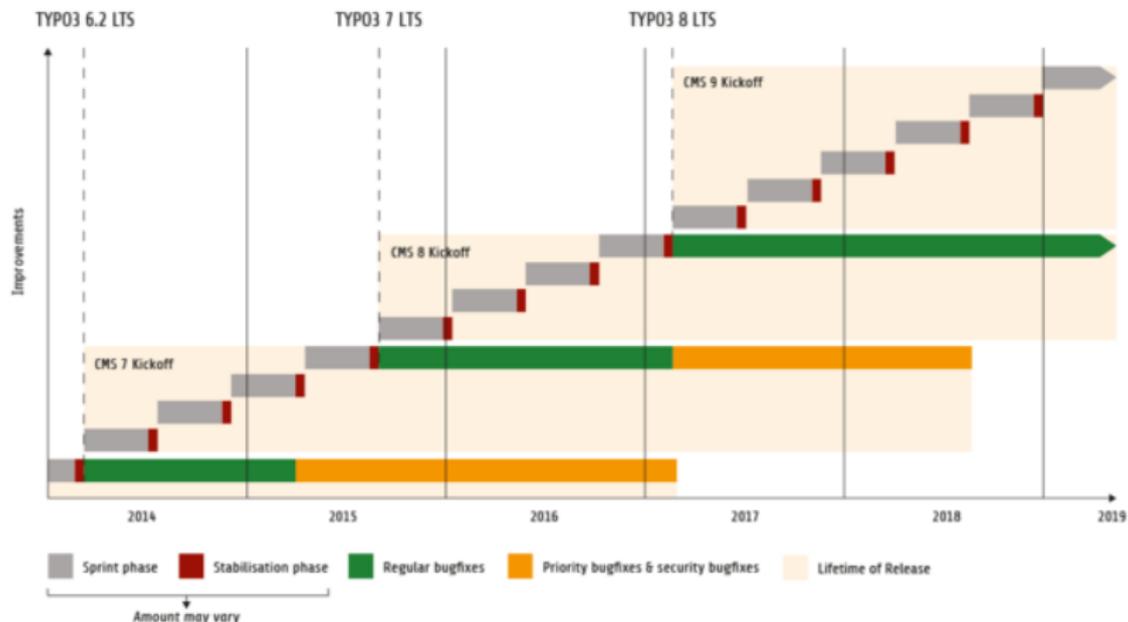
## Requisitos del Sistema

- PHP\*: v5.5.0 - v5.6.x
- MySQL: v5.5.x - v5.6.x (modo no estricto)
- Espacio de disco: min 200 MB
- Ajustes de PHP:
  - `memory_limit`  $\geq$  128M
  - `max_execution_time`  $\geq$  240s
  - opción de compilación `-disable-ipv6` no debe ser usada
- Backend requiere IE  $\geq$  9 o cualquier otro navegador moderno

\*) Detalles adicionales: [Requisitos Mínimos de PHP para TYPO3 CMS 7](#)

# Introducción

## Línea de tiempo de Desarrollo y Lanzamiento



# Introducción

---

## Línea de lanzamiento de TYPO3 CMS

Fechas de lanzamiento estimadas y sus enfoques principales:

- v7.0 02/Dic/2014 Revisión de Backend Vol 1
- v7.1 24/Feb/2015 Optimización & Limpieza del núcleo
- v7.2 28/Apr/2015 Frontend
- **v7.3 16/Jun/2015 Ecosistema de Paquetes, Composer y Manejo de Extensiones**
- v7.4 04/Ago/2015 Revisión de Backend Vol 2
- v7.5 29/Sep/2015 (*por determinar...*)
- v7.6 xx/xxx/2015 **TYP03 CMS 7 LTS** (Soporte a Largo Plazo)

<https://typo3.org/typo3-cms/roadmap/>

<http://typo3.org/news/article/embrace-and-innovate-typo3-cms-7/>

# Introducción

---

## Instalación

- Procedimiento de instalación oficial bajo Linux/Mac OS X (DocumentRoot por ejemplo /var/www/site/htdocs):

```
$ cd /var/www/site
$ wget --content-disposition get.typo3.org/7.3
$ tar xzf typo3_src-7.3.0.tar.gz
$ cd htdocs
$ ln -s ../typo3_src-7.3.0 typo3_src
$ ln -s typo3_src/index.php
$ ln -s typo3_src/typo3
$ touch FIRST_INSTALL
```

- Enlaces simbólicos bajo Microsoft Windows:
  - Use junction en Windows XP/2000
  - Use mlink en Windows Vista y Windows 7

# Introducción

---

## Actualización a TYPO3 CMS 7.x

- Actualizaciones sólo posibles desde TYPO3 CMS 6.2 LTS
- TYPO3 CMS < 6.2 deberá ser actualizado a TYPO3 CMS 6.2 LTS primero
- Instrucciones de actualización:  
[http://wiki.typo3.org/Upgrade#Upgrading\\_to\\_7.3](http://wiki.typo3.org/Upgrade#Upgrading_to_7.3)
- Guía oficial de TYPO3 "Instalación y Actualización de TYPO3":  
<http://docs.typo3.org/typo3cms/InstallationGuide>
- Enfoque general:
  - Comprobar requisitos mínimos del sistema (PHP, MySQL, etc.)
  - Revisar **deprecation\_\*.log** en instancia antigua de TYPO3
  - Actualizar todas las extensiones a las últimas versiones
  - Desplegar fuentes nuevas y ejecutar Herramienta de Instalación  
→Asistente de Actualización
  - Revisar el módulo de inicio para usuarios backend (opcionalmente)

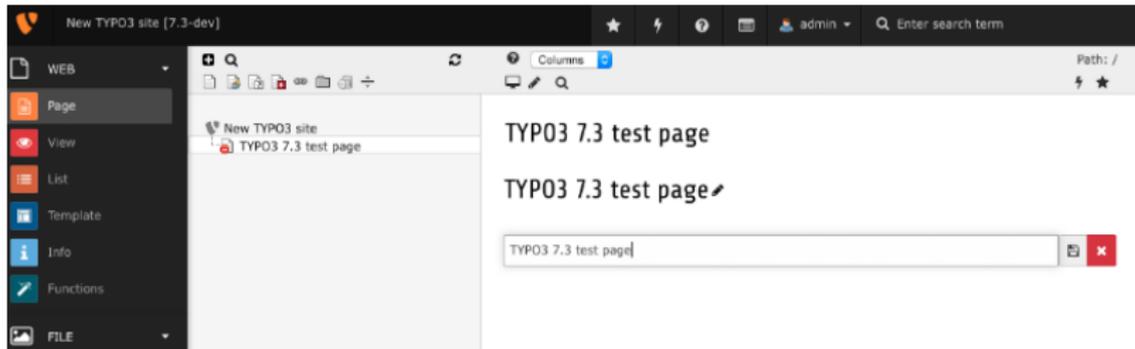
## Capítulo 1: Interfaz de Usuario de Backend

# Interfaz de Usuario de Backend

---

## Título de Página Title en Módulo Página y Lista

Los usuarios pueden editar los títulos de página en los módulos "Página" y "Lista" haciendo doble click en la cabecera del título o en el icono de edición.



# Interfaz de Usuario de Backend

---

## Herramienta de Instalación: Borrar Ficheros Procesados

En su sección "Clean up", la Herramienta de Instalación proporciona una nueva función para eliminar ficheros procesados (p.e. previsualizaciones de imágenes) desde FAL ahora.

Esto es útil si se han cambiado los ajustes relacionados con los gráficos o tras una actualización de GraphicsMagick/ImageMagick para forzar que todas las imágenes sean regeneradas.

### Clear processed files

The File Abstraction Layer stores a database record for every file it needs to process. (e.g. image thumbnails) In case you modified some graphics settings (All Configuration [GFX]) and you need all processed files to get regenerated, you can use this tool to remove the existing ones. The new processed files are created once they are needed.

Clear processed files

# Interfaz de Usuario de Backend

---

## Nuevo Campo en Metadatos FAL

Se ha añadido el campo "**Copyright**" a los metadatos de un registro FAL (extensión del sistema: filemetadata).

Edit File Metadata "test.txt" on root level

The screenshot shows the 'Edit File Metadata' interface for 'test.txt' on the root level. The 'Metadata' tab is selected, displaying several input fields for metadata. The fields are organized as follows:

- Creator**: A single text input field.
- Creator Tool** and **Publisher**: Two text input fields side-by-side.
- Source** and **Copyright**: Two text input fields side-by-side.
- Geo Location**: A section containing three text input fields for **Country**, **Region**, and **City**.

At the bottom right of the form, there is a button labeled 'File Metadata [6]'.

## Capítulo 2: TSconfig & TypoScript

# TSconfig & TypeScript

---

## Nueva Función `stdWrap strtotime` (1)

- Nueva propiedad TypeScript `stdWrap strtotime` permite la conversión de fechas formateadas a marcas de tiempo Unix, p.e. para llevar a cabo cálculos de fecha

# TScnfig & TypoScript

---

## Nueva Función stdWrap strtotime (2)

- Valores válidos son 1 o cualquier cadena de tiempo que sea usada como el primer argumento de la función PHP strtotime()

```
date_as_timestamp = TEXT
date_as_timestamp {
    value = 2015-04-15
    strtotime = 1
}
```

```
next_weekday = TEXT
next_weekday {
    data = GP:selected_date
    strtotime = + 2 weekdays
    strftime = %Y-%m-%d
}
```

# TSconfig & TypoScript

---

## GPmerged en Condiciones

- Usar GP en condiciones TypoScript sólo devuelve la variable POST, si la petición contiene ambas, las variables POST y GET
- Nueva opción GPmerged fusiona ambos métodos y devuelve el resultado

```
[globalVar = GPmerged:tx_demo|foo = 1]
  page.90 = TEXT
  page.90.value = DEMO
[global]
```

# Tsconfig & TypoScript

---

## Nuevas Opciones para `stdWrap.case`

- Se han añadido las opciones `uppercamelcase` y `lowercamelcase` a `stdWrap.case`
- Ejemplo:

```
tt_content = CASE
tt_content {
    key.field = CType
    my_custom_ctype =< lib.userContent
    my_custom_ctype {
        file = EXT:site_base/Resources/Private/Templates/SomeOtherTemplate.html
        settings.extraParam = 1
    }
    default =< lib.userContent
    default {
        file = TEXT
        file.field = CType
        file.stdWrap.case = uppercamelcase
        file.wrap = EXT:site_base/Resources/Private/Templates/|.html
    }
}
```

# Tsconfig & TypeScript

---

## Propiedad `integrity` Añadida para Ficheros JavaScript (1)

- Se ha añadido la propiedad `integrity` a las inclusiones de ficheros JavaScript para especificar una clave ISS para permitir la verificación del recurso  
(ISR: Integridad del Sub-Recurso, ver siguiente diapositiva)
- Esto afecta a las propiedades de Typoscript PAGE  
`page.includeJSLibs`, `page.includeJSFooterlibs`, `includeJS`  
y `includeJSFooter`
- Ejemplo:

```
page {  
  includeJS {  
    jQuery = https://code.jquery.com/jquery-1.11.3.min.js  
    jQuery.external = 1  
    jQuery.disableCompression = 1  
    jQuery.excludeFromConcatenation = 1  
    jQuery.integrity = sha256-7LkWEzqTdpEfELxcZZlS6wAx5Ff13zZ831Y02/ujj7g=  
  }  
}
```

## Propiedad `integrity` Añadida para Ficheros JavaScript (2)

- ISR es una especificación W3C que permite a los desarrolladores web asegurarse que los recursos alojados en servidores de terceros no han sido manipulados con
- Genera hashes de integridad:
  - Opción 1: <https://srihash.org>
  - Opción 2: use el siguiente comando de shell

```
cat FILENAME.js | openssl dgst -sha256 -binary | openssl enc -base64 -A
```

- Leer más:
  - <http://www.w3.org/TR/SRI/>

## Capítulo 3: Cambios en Profundidad

# Cambios en Profundidad

---

## Integración Symfony/Console en CommandController (1)

El CommandController hace uso ahora de Symfony/Console internamente y proporciona varios métodos:

- TableHelper

- `outputTable($rows, $headers = NULL)`

- DialogHelper

- `select($question, $choices, $default = NULL, $multiSelect = false, $attempts = FALSE)`
  - `ask($question, $default = NULL, array $autocomplete = array())`
  - `askConfirmation($question, $default = TRUE)`
  - `askHiddenResponse($question, $fallback = TRUE)`
  - `askAndValidate($question, $validator, $attempts = FALSE, $default = NULL, array $autocomplete = NULL)`
  - `askHiddenResponseAndValidate($question, $validator, $attempts = FALSE, $fallback = TRUE)`

# Cambios en Profundidad

---

## Integración Symfony/Console en CommandController (2)

- ProgressHelper

- `progressStart($max = NULL)`
- `progressSet($current)`
- `progressAdvance($step = 1)`
- `progressFinish()`

(ver siguientes diapositivas para ejemplos de código)

# Cambios en Profundidad

---

## Integración Symfony/Console en CommandController (3)

```
<?php
namespace Acme\Demo\Command;
use TYPO3\CMS\Extbase\Mvc\Controller\CommandController;

class MyCommandController extends CommandController {
    public function myCommand() {

        // render a table
        $this->output->outputTable(array(
            array('Bob', 34, 'm'),
            array('Sally', 21, 'f'),
            array('Blake', 56, 'm')
        ),
        array('Name', 'Age', 'Gender'));

        // select
        $colors = array('red', 'blue', 'yellow');
        $selectedColorIndex = $this->output->select('Please select one color', $colors, 'red');
        $this->outputLine('You choose the color %s.', array($colors[$selectedColorIndex]));

        [...]
    }
}
```

# Cambios en Profundidad

---

## Integración Symfony/Console en CommandController (4)

```
[...]  
// ask  
$name = $this->output->ask('What is your name?' . PHP_EOL, 'Bob', array('Bob', 'Sally', 'Blake'));  
$this->outputLine('Hello %s.', array($name));  
  
// prompt  
$likesDogs = $this->output->askConfirmation('Do you like dogs?');  
if ($likesDogs) {  
    $this->outputLine('You do like dogs!');  
}  
  
// progress  
$this->output->progressStart(600);  
for ($i = 0; $i < 300; $i++) {  
    $this->output->progressAdvance();  
    usleep(5000);  
}  
$this->output->progressFinish();  
}  
}  
?>
```

# Cambios en Profundidad

---

## Backend Login API (1)

- Se ha refactorizado completamente el login del backend y se ha introducido una nueva API
- Se ha extraído el formulario OpenID y ahora usa la nueva API (lo hace independiente de las clases centrales del Núcleo)
- El concepto del nuevo login del backend está basado en "login providers", que pueden ser registrados en el fichero `ext_localconf.php` como sigue:

```
$GLOBALS['TYPO3_CONF_VARS']['EXTCONF']['backend']['loginProviders'][1433416020] = [  
    'provider' => \TYPO3\CMS\Backend>LoginProvider\UsernamePasswordLoginProvider::class,  
    'sorting' => 50,  
    'icon-class' => 'fa-key',  
    'label' => 'LLL:EXT:backend/Resources/Private/Language/locallang.xlf:login.link'  
];
```

# Cambios en Profundidad

---

## Backend Login API (2)

- Las opciones se definen a continuación:
  - `provider`: nombre de la clase provider del login, que debe implementar `TYPO3\CMS\Backend>LoginProvider>LoginProviderInterface`
  - `sorting`: orden de los enlaces para los posibles providers de login en la pantalla de login
  - `icon-class`: nombre del icono font-awesome para el enlace de la pantalla de login
  - `label`: etiqueta para el enlace del proveedor de login en la pantalla de login

# Cambios en Profundidad

---

## Backend Login API (3)

- La LoginProviderInterface sólo contiene el método  

```
public function render(StandaloneView $view, PageRenderer $pageRenderer, LoginController $loginController);
```
- Los parámetros se definen a continuación:
  - `$view`:  
Fluid StandaloneView que renderiza la pantalla de inicio. Tiene que configurar el fichero template y puede añadir variables a la vista según sus necesidades.
  - `$pageRenderer`:  
La instancia PageRenderer provee la posibilidad de añadir recursos JavaScript necesarios.
  - `$loginController`:  
Instancia LoginController.

# Cambios en Profundidad

---

## Backend Login API (4)

- El template debe contener `<f:layout name="Login">` y `<f:section name="loginFormFields">` (para campos del formulario):

```
<f:layout name="Login" />
<f:section name="loginFormFields">
  <div class="form-group t3js-login-openid-section" id="t3-login-openid_url-section">
    <div class="input-group">
      <input type="text" id="openid_url"
        name="openid_url"
        value="{presetOpenId}"
        autofocus="autofocus"
        placeholder="{f:translate(key: 'openId', extensionName: 'openid')}}"
        class="form-control input-login t3js-clearable t3js-login-openid-field" />
      <div class="input-group-addon">
        <span class="fa fa-openid"></span>
      </div>
    </div>
  </div>
</f:section>
```

# Cambios en Profundidad

---

## CategoryRegistry con Nuevas Opciones

- El método `CategoryRegistry->addTcaColumn` acepta opciones para configurar `l10n_mode` y `l10n_display`:

```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::makeCategorizable(  
    $extensionKey,  
    $tableName,  
    'categories',  
    array(  
        'l10n_mode' => 'string (keyword)',  
        'l10n_display' => 'list of keywords'  
    )  
);
```

# Cambios en Profundidad

---

## Sprites en Módulos de Backend

- Los módulos del Backend (módulos principales tales como "Web" así como submódulos tales como "Filelist") pueden ahora usar sprites como iconos (¡sólo están disponibles iconos sprite conocidos para TYPO3!)
- Ejemplo:

```
\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::addModule(  
    'web',  
    'layout',  
    'top',  
    \TYPO3\CMS\Core\Utility\ExtensionManagementUtility::extPath($_EXTKEY) . 'Modules/Layout/',  
    array(  
        'script' => '_DISPATCH',  
        'access' => 'user,group',  
        'name' => 'web_layout',  
        'configuration' => array('icon' => 'module-web'),  
        'labels' => array(  
            'll_ref' => 'LLL:EXT:cms/layout/locallang_mod.xlf',  
        ),  
    ),  
);
```

# Cambios en Profundidad

---

## FormEngine NodeFactory API (1)

- Ahora es posible registrar nuevos nodos y sobrescribir nodos existentes

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['formEngine']['nodeRegistry'][1433196792] = array(  
    'nodeName' => 'input',  
    'priority' => 40,  
    'class' => \MyVendor\MyExtension\Form\Element\T3editorElement::class  
);
```

- El ejemplo de arriba registra una nueva clase `MyVendor\MyExtension\Form\Element\T3editorElement` como clase de renderizado para tipo TCA `input`, que debe implementar la interfaz `TYPO3\CMS\Backend\Form\NodeInterface`
- La clave del vector es la marca de tiempo Unix de la fecha cuando se añade un elemento de registro

# Cambios en Profundidad

---

## FormEngine NodeFactory API (2)

- En aquellos casos donde elementos de registro múltiples hayan sido registrados para el mismo tipo, se usa el resolvidor con la máxima prioridad (de 0 a 100)
- Un nuevo tipo TCA puede ser registrado como sigue:

TCA

```
'columns' => array(
    'bodytext' => array(
        'config' => array(
            'type' => 'text',
            'renderType' => '3dCloud',
        ),
    ),
),
)
```

ext\_localconf.php

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['formEngine']['nodeRegistry'][1433197759] = array(
    'nodeName' => '3dCloud',
    'priority' => 40,
    'class' => \MyVendor\MyExtension\Form\Element\ShowTextAs3dCloudElement::class
);
```

# Cambios en Profundidad

---

## Señal `postProcessMirrorUrl`

- Se ha movido la señal `postProcessMirrorUrl` a una nueva clase

### **BREAKING CHANGE!**

- El siguiente ejemplo de código tiene en cuenta la versión TYPO3:

```
$signalSlotDispatcher->connect(  
    version_compare(TYPO3_version, '7.0', '<')  
    ? 'TYPO3\CMS\Lang\Service\UpdateTranslationService'  
    : 'TYPO3\CMS\Lang\Service\TranslationService',  
    'postProcessMirrorUrl',  
    'Vendor\Extension\Slots\CustomMirror',  
    'postProcessMirrorUrl'  
);
```

## Capítulo 4: Extbase & Fluid

## ActionMenuItemGroupViewHelper (1)

- Usando este ViewHelper, pueden usarse grupos de opciones en el campo select del backend, que controla qué acción se selecciona
- Ejemplo:

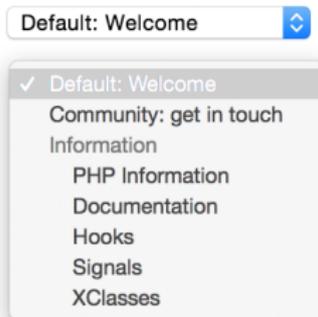
```
<f:be.menus.actionMenu>
  <f:be.menus.actionMenuItem label="Default: Welcome" controller="Default" action="index" />
  <f:be.menus.actionMenuItem label="Community: get in touch" controller="Community"
    action="index" />
  <f:be.menus.actionMenuItemGroup label="Information">
    <f:be.menus.actionMenuItem label="PHP Information" controller="Information"
      action="listPhpInfo" />
    <f:be.menus.actionMenuItem label="Documentation" controller="Information"
      action="documentation" />
    <f:be.menus.actionMenuItem label="Hooks" controller="Information" action="hooks" />
    <f:be.menus.actionMenuItem label="Signals" controller="Information" action="signals" />
    <f:be.menus.actionMenuItem label="XClasses" controller="Information" action="xclass" />
  </f:be.menus.actionMenuItemGroup>
</f:be.menus.actionMenu>
```

# Extbase & Fluid

---

## ActionMenuItemGroupViewHelper (2)

- El ejemplo de la diapositiva anterior produce la siguiente salida:



## Soporte de Template para FlashMessagesViewHelper

- El FlashMessagesViewHelper soporta templates ahora
- El nuevo atributo `as` permite especificar un nombre de variable, que puede usarse dentro de los elementos hijos del ViewHelper para acceder a los mensajes flash
- Ejemplo:

```
<f:flashMessages as="flashMessages">
  <ul class="myFlashMessages">
    <f:for each="{flashMessages}" as="flashMessage">
      <li class="alert {flashMessage.class}">
        <h4>{flashMessage.title}</h4>
        <span class="fancy-icon">{flashMessage.message}</span>
      </li>
    </f:for>
  </ul>
</f:flashMessages>
```

- Nota: opción `renderMode` está obsoleta ahora

# Extbase & Fluid

---

## Nuevas Propiedades del cObject FLUIDTEMPLATE (1)

- Se ha extendido el cObject FLUIDTEMPLATE con `templateRootPaths` y `templateName`
- Es posible configurar un nombre de template y al visualizar el template este nombre es usado junto con el formato de configuración para encontrar el template en los `templateRootPaths` proporcionados
- `templateRootPaths` ofrece la misma lógica de fallback como `layoutRootPath` y `partialRootPath`
  - `templateName`: `string/stdWrap`
  - `templateRootPaths`: vector de rutas de fichero con soporte de prefijo "EXT:"

## Nuevas Propiedades del cObject FLUIDTEMPLATE (2)

### ■ Ejemplo TypeScript:

```
lib.stdContent = FLUIDTEMPLATE
lib.stdContent {
    templateName = TEXT
    templateName.stdWrap {
        cObject = TEXT
        cObject {
            data = levelfield:-2,backend_layout_next_level,slide
            override.field = backend_layout
            split {
                token = frontend__
                1.current = 1
                1.wrap = |
            }
        }
        ifEmpty = Default
    }
    templateRootPaths {
        10 = EXT:frontend/Resources/Private/Templates
        20 = EXT:sitemodification/Resources/Private/Templates
    }
}
```

## Borrado de Atributos `xmlns` y Tags HTML (1)

- Con la introducción del uso de atributos `xmlns:*` para incluir ViewHelpers, es posible tener soporte IDE para templates Fluid. El problema es que los atributos `xmlns:*` y el tag correspondiente serán también renderizados, lo cual es usualmente no deseable.
- La solución es usar secciones, pero esta solución no es muy intuitiva y no disponible en layouts. También causa extra procesamiento de la cabecera.
- Atributos `xmlns:*` para namespaces de ViewHelper serán ahora eliminados antes del renderizado, si muestran la siguiente sintaxis:  
`http://typo3.org/ns/<phpNamespace>`  
(atributos `xmlns` para namespaces de no-ViewHelper son preservados)

# Extbase & Fluid

---

## Borrado de Atributos xmlns y Tags HTML (2)

- Incluya namespaces de ViewHelper dentro del tag HTML y el atributo `data-namespace-typo3-fluid="true"` para prevenir el renderizado del tag HTML al completo

```
<html data-namespace-typo3-fluid="true"
  xmlns:f="http://typo3.org/ns/TYPO3/CMS/Fluid/ViewHelpers"
  xmlns:n="http://typo3.org/ns/GeorgRinger/News/ViewHelpers">

  <f:if condition="{newsItem.title}">
    <f:then>
      <n:titleTag>{newsItem.title}</n:titleTag>
    </f:then>
    <f:else>
      <n:titleTag>News-Detail</n:titleTag>
    </f:else>
  </f:if>

</html>
```

## Nuevos Métodos en Fluid-StandaloneView

- StandaloneView es extendida con  
setTemplateRootPaths(\$templatePaths) y  
setTemplate(\$templateName, \$throwException = TRUE)
- Misma funcionalidad que cObject FLUIDTEMPLATE
- Ejemplo (renderizado de un template de email):

```
$view = GeneralUtility::makeInstance(StandaloneView::class);  
$view->setLayoutRootPaths(array(GeneralUtility::getFileAbsFileName(  
    'EXT:my_extension/Resources/Private/Layouts')));  
$view->setPartialRootPaths(array(GeneralUtility::getFileAbsFileName(  
    'EXT:my_extension/Resources/Private/Partials')));  
$view->setTemplateRootPaths(array(GeneralUtility::getFileAbsFileName(  
    'EXT:my_extension/Resources/Private/Templates')));  
$view->setTemplate('Email/Notification');  
$emailBody = $view->render();
```

## Procesamiento de Datos para cObject FLUIDTEMPLATE (1)

- Se ha extendido el cObject FLUIDTEMPLATE con `dataProcessing`
- Puede configurarse este parámetro para añadir uno o múltiples procesadores para manipular la variable `$data` del cObject actualmente renderizado (p.e. `tt_content` o `page`)
- El procesador debe implementar la interfaz `FluidTemplateDataProcessorInterface` y contiene el siguiente método:

```
function process(array &$data, array $processorConfiguration,  
    array $configuration, StandaloneView $view) {  
    [...]  
}
```

## Procesamiento de Datos para cObject FLUIDTEMPLATE (2)

### ■ Ejemplo:

```
my_custom_ctype = FLUIDTEMPLATE
my_custom_ctype {
    templateRootPaths {
        10 = EXT:your_extension_key/Resources/Private/Templates
    }
    templateName = CustomName
    settings {
        extraParam = 1
    }
    dataProcessing {
        1 = Vendor\YourExtensionKey\DataProcessing\MyFirstCustomProcessor
        2 = AnotherVendor\AnotherExtensionKey\DataProcessing\MySecondCustomProcessor
        2 {
            options {
                myOption = SomeValue
            }
        }
    }
}
```

## Capítulo 5: Funciones Obsoletas/Eliminadas

# Funciones Obsoletas/Eliminadas

---

## Refactorizado del FormEngine

### TCA:

- Opciones `_PADDING`, `_VALIGN` and `DISTANCE` han sido eliminadas de `TCA['aTable']['columns']['aField']['config']['wizards']`
- La clave `TCA['aTable']['ctrl']['mainPalette']` ha sido eliminada

### TSconfig:

- Las claves `mod.web_layout.tt_content.fieldOrder` y `TCEFORM.aTable.aField.linkTitleToSelf` han sido eliminadas

### Hooks:

- Hooks usan la clave `type` en lugar de `form_type` ahora
- Ha sido eliminado el hook `getSingleFieldClass`

# Funciones Obsoletas/Eliminadas

---

## Eliminación de IdentityMap de la Persistencia de Extbase

- Se ha eliminado la clase IdentityMap de la persistencia de Extbase (una ReflectionException es lanzada si se usa todavía)
- El acceso a las previamente existentes propiedades IdentityMap dentro de DataMapper y Repository fallará ahora y la creación de instancias IdentityMap no es posible más
- Use la persistencia de "Sessions" en su lugar:

```
$session = GeneralUtility::makeInstance(ObjectManager::class)->get(
    \TYPO3\CMS\Extbase\Persistence\Generic\Session::class
);

$session->registerObject($object, $identifier);

if($session->hasIdentifier($identifier)) {
    $object = $session->getObjectByIdentifier($identifier, $className);
}
```

# Funciones Obsoletas/Eliminadas

---

## Miscelánea (1)

- El fichero `typo3conf/extTables.php` está obsoleto. Use el siguiente fichero en su lugar:  
`<your_extension>/Configuration/TCA/Overrides/pages.php`
- Se ha eliminado la configuración `$TYPO3_CONF_VARS[GFX][png_to_gif]`
- En instalaciones de TYPO3 CMS, que no tienen la extensión `rsaauth` instalada, passwords del login del BE serán transferidos en texto plano ahora  
(solución: instale la extensión `rsaauth` o use HTTPS para el BE)
- Método `exec_SELECTgetRows()` valida parámetro `$uidIndexField` ahora. Si el campo especificado no está presente en el resultado de la base de datos, una `InvalidArgumentException` es lanzada.

# Funciones Obsoletas/Eliminadas

---

## Miscelánea (2)

- Se ha eliminado la opción DBAL `config.classFile`
- Han sido marcadas ahora como obsoletas las opciones `iconOnly` y `styleAttributes` de `CshViewHelper`
- La opción TypoScript `page.bgImg` está obsoleta ahora
- El método `isEnabled()` de la clase `T3editor` está obsoleto ahora
- El viejo `ClassLoader TYPO3` ha sido eliminado en pro del `ClassLoader` de `Composer`

## Capítulo 6: Fuentes y Autores

# Fuentes y Autores

---

## Fuentes

### Noticias de TYPO3:

- <http://typo3.org/news>

### Informaciones de Lanzamiento:

- [http://wiki.typo3.org/TYPO3\CMS\\_7.3.0](http://wiki.typo3.org/TYPO3\CMS_7.3.0)
- [INSTALL.md](#) y [Changelog](#)
- [typo3/sysex/core/Documentation/Changelog/7.3/](http://typo3.org/sysex/core/Documentation/Changelog/7.3/)\*

### Sistema de seguimiento de errores de TYPO3:

- <https://forge.typo3.org/projects/typo3cms-core>

### Repositorios Git de TYPO3:

- <https://git.typo3.org/Packages/TYPO3.CMS.git>
- <https://git.typo3.org/Packages/TYPO3.Fluid.git>

## Diapositivas TYPO3 CMS Qué hay Nuevo:

Patrick Lobacher

(Investigación, recogida de información y versión en Alemán)

Michael Schams

(Líder del proyecto y versión en Inglés)

### Traducciones por:

Andrey Aksenov, Sergio Catala, Jigal van Hemert, Michel Mix, Sinisa Mitrovic,  
Angeliki Plati, Nena Jelena Radovic, Roberto Torresani

<http://typo3.org/download/release-notes/whats-new>

Licencia bajo Creative Commons BY-NC-SA 3.0

