

# Desarrollo de Aplicaciones en Plone

**Mikel Larreategi**  
**mlarreategi@codesyntax.com**

2008/12/15

# CodeSyntax

Internet Solutions  
[www.codesyntax.com](http://www.codesyntax.com)

Azitain Poligonoa 3 - K  
E-20600 EIBAR

Tel: ++34 943 821 780  
Fax: ++34 943 820 669

## Recordemos

- ZPT
- Python
- Buildout
- Plone básico

## Buildout

- Archivo de configuración formato INI
- Ahí decimos qué cosas se instalan en Plone
- Podemos crear nuestros productos para que se puedan instalar con buildout

## Productos para Plone: eggs

- *Un egg es Python lo que un jar es a Java*
- Es simplemente un archivo ZIP que contiene el código y metadatos para su gestión de cierta forma automatizada
- Veamos el esqueleto de un huevo

## ¿Cómo creamos un huevo?

- A mano
- A máquina:
  - *\$ paster create --list-templates*
  - *\$ paster create -t plone*
  - *\$ paster create -t plone\_app*
  - *\$ paster create -t plone3\_theme*
  - *\$ paster create -t archetype*
- Esto hay que combinarlo con `--svn-repository` para incluirlo en SVN (o en cualquier otro a mano)

## Plone Theme (I)

- *\$ paster create -t plone3\_theme*
- Veamos la estructura del producto
- Expliquemos lo que hay en cada sitio
  - setup.py
  - \_\_init\_\_.py
  - configure.zcml
  - profiles/default
  - GenericSetup
  - skins

## Plone Theme (II)

- Skins
  - Personalización de plantillas y scripts de plone
  - Creación de nuevas plantillas
- ¿Por qué funciona?
  - portal\_skins
  - Concepto de layer o capa
  - Concepto de skin

## Plone Theme (III)

- Ejemplos:
  - Personalicemos una plantilla
  - Creemos una nueva plantilla
- Macros, slots, viewlets, etc.

## Plone Theme (y IV)

- Vamos a utilizar GenericSetup para que la plantilla que hemos añadido se pueda utilizar como portada de la página principal

## ¿Qué es GenericSetup?

- Una forma para exportar la configuración de Plone a archivos XML para que pueda ser reproducible
- ¿De qué configuración hablamos?
  - De casi todos los elementos que hay en el ZMI
    - portal\_actions, portal\_catalog, portal\_types...
  - Registrar nuevos tipos de objetos, perfiles de actualización, ...
- Veamos el perfil básico de Plone

## Archetypes (I)

- ¿Qué es Archetypes?
  - Herramientas para crear nuevos tipos de objeto en Plone
    - Los objetos básicos de Plone están hechos con Archetypes
  - Schema, Field, Widget
- Veamos cómo está hecha una noticia en Plone

## Archetypes (II)

- Creemos el nuestro:
  - *\$ paster create -t archetype*
- Sin embargo cada vez que añadamos un nuevo objeto, hay que “tocar” muchos archivos:
  - Crear el .py, el interfaz
  - Añadirlo en el perfil GS (types.xml, types y factory.xml)
  - Crear permisos, por si acaso ,
  - ...

## Archetypes (y III)

- *\$ paster addcontent --list*
- *\$ paster addcontent contenttype*
- Ahora añadiremos a mano algún campo
- Pero eso incluso se puede hacer a máquina:
  - *\$ paster addcontent atschema*
  - Tiene sus “peros”, pero bueno, sirve para empezar.

## Aplicaciones en Plone (I)

- Concepto de vista en Zope3
- ¿Por qué una vista y no un script de skins?
  - Todo el poder de Python
  - Código restringido
- Cómo añadimos una vista
  - A mano (.py y configure.zcml)
  - A máquina (sólo en productos que hayamos creado con la plantilla *archetype*):
    - *\$ paster addcontent view*

## Aplicaciones en Plone (II)

- Hagamos un ejemplo:
  - Un formulario que recoja un string y dos enteros y que muestra el string en negrita, los dos enteros en una lista y la suma de los dos enteros también en negrita.

## Aplicaciones en Plone (III)

- `portal_catalog`
  - Plone utiliza ZODB
  - Base de Datos orientada a objetos
  - No hacemos consultas SQL para buscar cosas
  - Las buscamos en una cosa llamada Catálogo, en la que se indexan todos los objetos y guardan cierta información
  - Veámos qué tiene nuestro catálogo

## Aplicaciones en Plone (IV)

- Veámos cómo hace Plone una búsqueda en el catálogo. Normalmente se utilizan vistas (aunque se puede hacer desde scripts)
  - `CMFPlone/browser/portlets/news.py`
- Concepto de brain y concepto de objeto
  - Datos indexados en el catálogo que obtenemos directamente del brain: Metadata
- Añadiendo nuevos índices y metadata utilizando GS

## Aplicaciones en Plone (V)

- Hagamos que nuestro tipo de objeto indexe uno de sus atributos en el catálogo
- Hagamos que algún atributos de nuestro objeto se indexe en el índice de texto libre

## Aplicaciones en Plone (VI)

- Una plantilla, que utilizando una vista, nos devuelva ciertos objetos de Plone ordenados
- Una plantilla y una vista que hagan lo mismo que hace el formulario de búsqueda de Plone

## Y ahora qué... (I)

- Consultas SQL
  - Hay librerías en python que se pueden utilizar
    - MySQLdb
    - PostgreSQL
    - SQLite
    - cx\_Oracle
  - Añadiríamos como egg y utilizaríamos la librería como si fuera un programa python
  - SQLAlchemy, un ORM para Python

## Y ahora qué (y II)

- Con las vistas y las plantillas podéis hacer lo que se os ocurra

## Bibliografía

- Professional Plone Development. Martin Aspeli. PacktPub 2007 (también en PDF)
- Web Component Development with Zope 3. Philipp von Weitershausen. Springer 2007
- Documentación online: <http://plone.org>