

Presupuestos de Aragón

Documentación Técnica

8/1/2013. Versión 0.1: Primera versión.

19/1/2013. Versión 1.0: Añadidas instrucciones sobre ejecución presupuestaria.

13/1/2014. Versión 2.0: Actualizado para incluir información sobre la Fase 2 del proyecto.

27/1/2014. Versión 2.1: Corregidas erratas. Extendido proceso de instalación.

Introducción

Los Presupuestos Generales de una administración pública, a cualquier nivel, han sido tradicionalmente un objeto opaco a la ciudadanía, que - a pesar de ser el proveedor de los fondos públicos a través de sus impuestos - suele recibir información parcial, descontextualizada, no suficientemente explicada y en formatos poco manejables. La prensa y los organismos públicos no han realizado el trabajo didáctico necesario para la comprensión por parte del ciudadano-lector, que ha tenido que conformarse con el análisis particular del medio de comunicación, y con las declaraciones de los distintos actores políticos al paso de los Presupuestos por el parlamento.

El objetivo de la primera fase del proyecto consistió en ofrecer una visualización de los Presupuestos Generales de Aragón suficientemente intuitiva como para ser comprendida por personas sin experiencia previa, pero haciendo a la vez disponibles los detalles de cada elemento del presupuesto para las personas interesadas en profundizar más. La primera fase, ya completada, muestra la realidad del presupuesto en su conjunto, cubriendo tanto el lado de los ingresos como el de los gastos, y tanto las cantidades previstas como los finalmente realizadas (cuando la información esté disponible).

El objetivo de la segunda fase es extender la transparencia de los presupuestos de la comunidad a niveles inferiores: municipios y comarcas.

Las principales funcionalidades de la aplicación son:

- Visualización de gastos e ingresos presupuestados, de forma jerárquica y según las cuatro clasificaciones usadas en los presupuestos: funcional (para qué se gasta), económica (en qué se gasta, o cómo se ingresa), financiación (origen y tipo de los fondos) y orgánica/institucional (quién gasta/ingresa).
- Mostrar la información de los programas presupuestarios al máximo nivel existente, el nivel de partida.
- Mostrar la evolución de los presupuestos desde 2006.
- Búsqueda de texto libre en los presupuestos, facilitando encontrar cualquier dato.

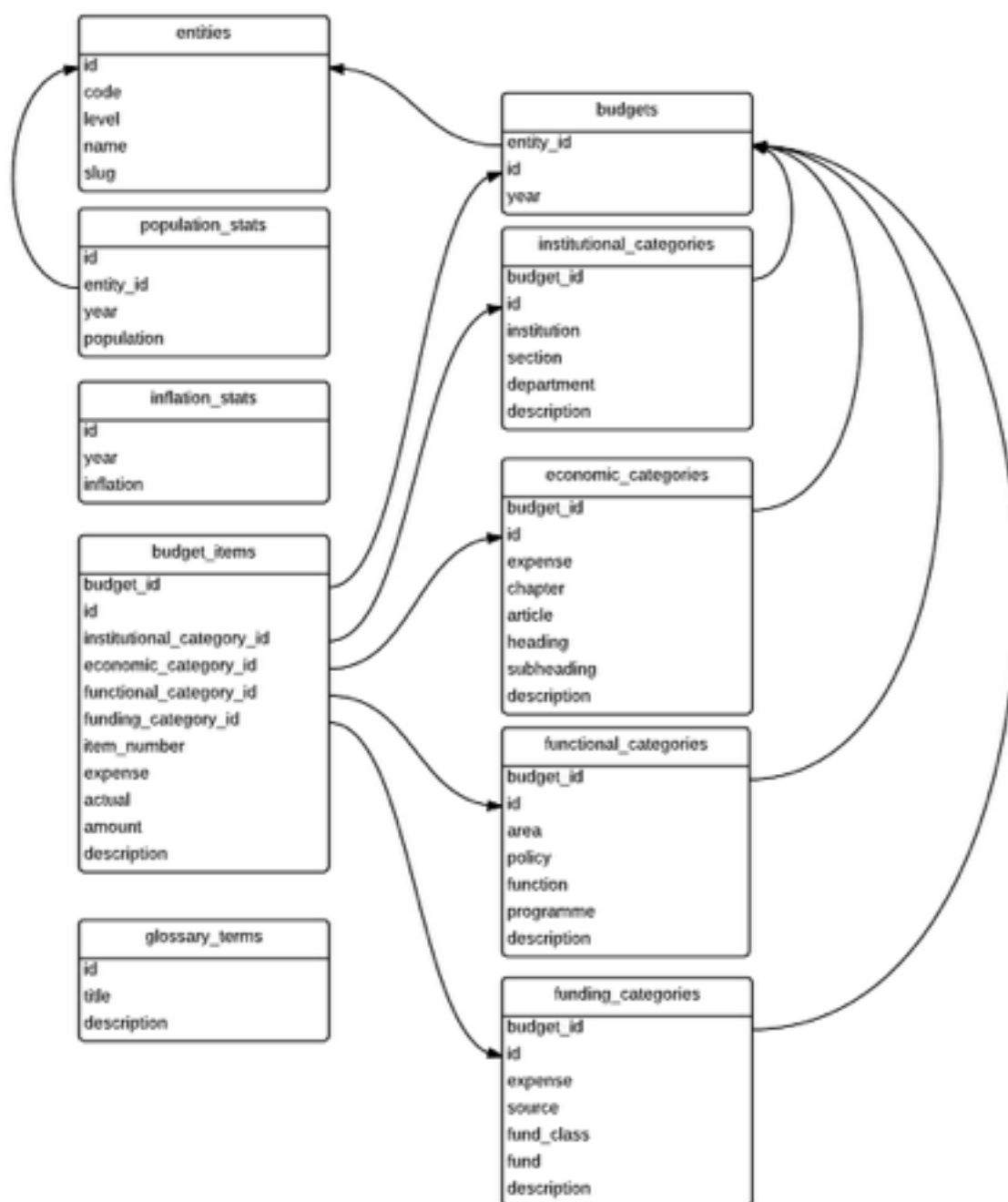
Arquitectura del software

La aplicación web está desarrollada sobre Python 2.6, sobre la plataforma de código abierto Django 1.4.3, utilizando la arquitectura MVC (Model-View-Controller) común en aplicaciones web.

- El modelo se almacena en las tablas de una base de datos relacional utilizando el componente de persistencia incluido por defecto en Django. Actualmente la aplicación sólo es compatible con PostgreSQL 8+. El modelo de datos detallado se incluye en la siguiente sección.
- Las páginas generadas por la aplicación están compuestas por mark-up compatible con HTML5, así como por código Javascript. Las visualizaciones están realizadas sobre la librería Javascript de código abierto D3, cuyo formato de salida es SVG, compatible sin necesidad de plug-ins adicionales con Firefox, Chrome, Safari e Internet Explorer 9 (o anteriores, a través de la extensión Google Chrome Frame), así como con las versiones móviles de estos navegadores. La aplicación no incluye ningún formato ni componente multimedia adicional, como Flash, vídeo o audio.
- El código de los controladores — encargados de procesar las peticiones de los usuarios, recuperar la información del modelo y entregársela a las vistas — está desarrollado en Python.

La aplicación no incluye ningún proceso de segundo plano (*daemons* o servidores adicionales), ni tareas *batch* que deban ejecutarse regularmente. Sí que incluye una serie de procesos de carga de datos, escritos en Python, como se detalla en la sección “Instalación”.

Esquema de datos



Instalación

Esta sección explica como instalar la aplicación web en un nuevo entorno.

Entrega del software y gestión de configuraciones

El código de la aplicación se entregará en un fichero .tar.gz, con el nombre 'presupuestos-aragon-x.x.tar.gz', donde x.x es el número de versión. El primer paso para proceder a la instalación es descomprimir el fichero, que contiene el código y los datos de la aplicación, en el directorio de destino, /apps/presupuestos-aragon.

Dependencias

La aplicación se ejecutará sobre la infraestructura técnica de Aragón Open Data, cuyas características principales son:

- Servidor web Apache
- Base de datos PostgreSQL 9.1
- Python 2.7

La aplicación necesita instalar una serie de paquetes Python:

- Coffin
- Django
- Jinja2
- Pygments
- psycopg2
- wsgiref

Las dependencias pueden ser instaladas utilizando gestores de paquetes de Python como **virtualenv** o **pip**. Pero también podemos instalar manualmente las versiones específicas de los componentes necesarios:

```
$ easy_install http://pypi.python.org/packages/source/C/Coffin/
Coffin-0.3.7.tar.gz#md5=a61c7820bd1c6ec4c9a013a4e7826f4b
$ easy_install http://pypi.python.org/packages/source/D/Django/
Django-1.4.3.tar.gz#md5=0b134c44b6dc8eb36822677ef506c9ab
$ easy_install http://pypi.python.org/packages/source/J/Jinja2/
Jinja2-2.6.tar.gz#md5=1c49a8825c993bfdcf55bb36897d28a2
$ easy_install http://pypi.python.org/packages/source/P/Pygments/
Pygments-1.6rc1.tar.gz#md5=633fbb59d2f05acedcafd8cd851d0935
```

```
$ easy_install http://pypi.python.org/packages/source/d/distribute/
distribute-0.6.32.tar.gz#md5=acb7a2da81e3612bfb1608abe4f0e568
$ easy_install http://pypi.python.org/packages/source/x/xlwt/
xlwt-0.7.5.tar.gz#md5=59cb5efd55319465dfcd25e6a485f03c
$ easy_install http://initd.org/psycopg/tarballs/PSYCOPG-2-5/psycopg2-2.5.2.tar.gz
```

El componente 'psycopg2' es el responsable de la conectividad con la base de datos Postgres, y tiene una parte nativa que debe ser compilada en el momento de la instalación. Para que la compilación se haga correctamente es necesario que el sistema operativo tenga instalados los paquetes 'postgresql-server-dev-9.1' y 'python-dev'

Configuración del acceso a la base de datos

La aplicación lee la configuración para su acceso a base de datos del fichero '**project/settings.py**'. Por defecto la aplicación intentará conectarse a una base de datos '**presupuestos-aragon**', con el usuario '**presus**' (y misma contraseña).

```
$ createuser -s -P presus
$ createdb -O presus presupuestos-aragon
```

En caso de que la configuración del entorno de destino sea distinto, el fichero '**project/settings.py**' deberá ser modificado en las líneas 23-32:

```
'default': {
    'ENGINE': 'django.db.backends.postgresql_psycopg2',
    'NAME': '<nombre de la base de datos>',
    'USER': '<nombre del usuario>',
    'PASSWORD': '<contraseña, si la hay>',
    'HOST': 'localhost',
    'PORT': '',
}
```

Una vez configurado el acceso a la base de datos debemos crear el esquema de la aplicación. Desde el directorio de la aplicación:

```
$ python manage.py syncdb
```

Carga del glosario

La aplicación incluye un glosario de términos relacionados con la economía y los presupuestos. Estos términos se actualizan a partir de un fichero de texto CSV, '**data/glosario.csv**', cuyo formato es el siguiente:

- Título/Término en castellano
- Descripción en castellano

Existe un script para manipular esta información:

```
python manage.py load_glossary
```

Al instalar la aplicación por primera vez debemos cargar los términos del glosario para que estén disponibles:

```
$ python manage.py load_glossary
Cargando glosario de ../data/glosario.csv...
Cargando término 'Inflación'...
...
```

Carga de la lista de municipios y comarcas

La aplicación necesita disponer de una lista de municipios y comarcas de Aragón. Esta lista se actualizan a partir de un fichero de texto CSV, `'data/entidades.csv'`, cuyo formato es el siguiente:

- Identificador único (en nuestro caso, proporcionado por el Instituto Nacional de Estadística)
- Tipo de entidad (comunidad autónoma, comarca o municipio)
- Nombre de la entidad

Existe un script para manipular esta información:

```
python manage.py load_entities
```

Al instalar la aplicación por primera vez debemos cargar la lista de organismos para que estén disponibles:

```
$ python manage.py load_entities
Cargando lista de organismos de ../data/entidades.csv...
...
```

Carga de estadísticas oficiales

La aplicación utiliza dos estadísticas oficiales — número de habitantes de comarcas y municipios, así como inflación anual en Aragón — a la hora de mostrar la información presupuestaria, permitiendo hacer cálculos de gasto per-cápita, y ajustando las cantidades a la inflación. Estos datos estadísticos se actualizan a partir de dos ficheros de texto CSV. El primero, `'data/inflacion.csv'`, tiene el siguiente formato:

- Año
- Inflación anual

El segundo, `'data/poblacion.csv'`, contiene las siguientes columnas:

- Identificador del organismo (comarca, municipio o comunidad autónoma)
- Nombre del organismo
- Año

- Número de habitantes

Existe un script para manipular esta información:

```
python manage.py load_stats
```

Al instalar la aplicación por primera vez debemos cargar las estadísticas para que estén disponibles:

```
$ python manage.py load_stats
Cargando estadísticas oficiales de .../data/estadisticas.csv...
Cargando estadísticas oficiales para el año 2002...
Cargando estadísticas oficiales para el año 2003...
...
```

Carga de los datos presupuestarios de Aragón

La aplicación lee la información presupuestaria a partir de los ficheros descargados de Aragón Open Data, sin necesidad de realizar ninguna modificación a estos. Los ficheros originales se almacenan en la carpeta '**data/**' de la aplicación, desde donde pueden ser cargados y analizados mediante el comando:

```
python manage.py load_budget <año_a_cargar>
```

Así, para cargar los datos del año 2013, y desde el directorio donde se haya instalado la aplicación:

```
$ python manage.py load_budget 2013
Cargando presupuesto de .../data/2013...
Cargando lista de secciones de .../data/2013/estructura_organica.csv...
Cargando jerarquía económica de .../data/2013/estructura_economica.csv...
...
```

A la hora de estimar el espacio de almacenamiento en base de datos necesario para el funcionamiento de la aplicación podemos fijarnos exclusivamente en el espacio utilizado por los presupuestos, ya que son el único elemento de tamaño no despreciable (frente al glosario y las estadísticas oficiales que se describen a continuación):

- Los presupuestos se publican anualmente en Aragón Open Data como una serie de ficheros CSV que ocupan menos de 1MB por año.
- Una base de datos PostgreSQL con los presupuestos de 2006 a 2012 cargados ocupa, en el entorno de desarrollo, aproximadamente 5MB.

Ya que la aplicación no almacena ninguna información en el día a día, podemos esperar un crecimiento en su espacio necesario de aproximadamente 1MB al año.

Carga de datos de ejecución presupuestaria

La aplicación es capaz de mostrar los datos de ejecución presupuestaria a nivel de programa, si están disponibles. (Si no están disponibles se mostrará sólo el presupuesto, sin que suponga un problema.) Estos datos se actualizan a partir de dos fichero de texto CSV, 'data/[año]/ejecucion_gastos.csv' y 'data/[año]/ejecucion_ingresos.csv', cuyo formato es el mismo que el de los ficheros presupuestarios.

Existe un script para manipular esta información:

```
python manage.py load_execution <año_a_cargar>
```

Al instalar la aplicación por primera vez podemos cargar los datos de ejecución para que estén disponibles. Por ejemplo, para cargar la ejecución presupuestaria del año 2010:

```
$ python manage.py load_execution 2010
Cargando ejecución presupuestaria de ../data/2010...
Cargando ejecución de gastos de ../data/2010/ejecucion_gastos.csv...
Cargando ejecución de ingresos de ../data/2010/ejecucion_ingresos.csv...
...
```

Carga de los datos presupuestarios de municipios y comarcas

Los ficheros con información presupuestaria y de ejecución se almacenan en las carpetas 'data/comarca' y 'data/municipio' de la aplicación, desde donde pueden ser cargados y analizados mediante los comandos:

```
$ python manage.py load_budget_data <tipo_de_organismo> <periodo_a_cargar>
```

Al instalar la aplicación por primera vez debemos cargar los datos de comarcas y municipios para que estén disponibles. Por ejemplo, para cargar los datos de todas las comarcas de Aragón para el año 2011:

```
$ python manage.py load_budget_data comarca 2011
Cargando presupuesto para entidad 'La Jacetania' año 2011...
Cargando presupuesto para entidad 'Alto Aragón' año 2011...
...
```

Y de forma similar, para los municipios:

```
$ python manage.py load_budget_data municipio 2011
Cargando presupuesto para entidad 'Abiego' año 2011...
Cargando presupuesto para entidad 'Zuera' año 2011...
```

Rendimiento

Algunas de las consultas que la aplicación realiza en la base de datos implican leer una cantidad considerable de datos: así, para calcular el total de gasto para una política concreta, como "Sanidad", es necesario leer y sumar todas las líneas del presupuesto incluidas en esa política (varios cientos). Pero toda la información que maneja la aplicación es estática — no se realiza ninguna actualización en ningún momento —, y es independiente del usuario de la web, por lo que puede ser

cacheada indefinidamente sin ningún problema, utilizando los mecanismos estándares de Django. Así, el impacto de la aplicación en el rendimiento del servidor es mínimo después de haber servido las primeras peticiones.

La aplicación utilizará el caché en memoria incluido por defecto en Django, sin ninguna dependencia con servicios externos de caché tipo 'memcached'.

Estadísticas

La aplicación incluye el código necesario para realizar el seguimiento del tráfico de la web vía Google Analytics.

Mantenimiento

Dependencias con otros componentes

La aplicación no tiene ninguna dependencia con ningún otro servicio o componente del entorno, más allá del servidor web y la base de datos. La aplicación:

- no manda correos
- no muestra vídeos, ni en streaming ni como descarga
- no tiene cuentas de usuario, por lo que no se conecta a ningún servidor de autenticación (LDAP o similar)

Otros procesos y tareas

La aplicación no necesita ejecutar ningún proceso (*daemon*) adicional a los encargados de servir las peticiones de los usuarios. Tampoco necesita configurar ningún tipo de tarea periódica / *batch*.

Seguridad y accesos

La información disponible en la aplicación está formada por datos públicos ya disponibles en otras webs de Aragón Open Data. No existe, por tanto, ninguna información confidencial.

Por otro lado, la aplicación no permite crear ningún tipo de cuenta de usuario, ni almacena ninguna información de sus usuarios.

Las tareas de mantenimiento (carga de datos) se realizan vía *rake* a través de una conexión SSH al servidor. El control de acceso de la aplicación es, por tanto, el de las credenciales necesarias para acceder al servidor vía SSH.

Tareas regulares de mantenimiento

La única tarea regular de mantenimiento que precisa la aplicación es la rotación y/o limpieza de sus logs periódicamente. El tamaño de estos logs es similar al de cualquier otra aplicación web, y depende exclusivamente del volumen de tráfico que reciba, por lo que es difícil de estimar a priori.

No existen tareas adicionales, ya que:

- La información mostrada por la aplicación es estática, no crece en el día a día, por lo que no son necesarios procesos de limpieza y archivo en la base de datos.
- Al no existir cuentas de usuario, no existen peticiones de alta, baja o soporte.
- Al no existir datos privados, en caso de fallo grave del sistema (pérdida del sistema de ficheros o base de datos) la aplicación puede restaurarse a partir de cualquier copia de seguridad. Si no existieran o se perdieran es posible reinstalar

el sistema desde cero, a partir del código y datos almacenados en SVN — cómo se explicó en la sección previa — sin que se pierda información.

Tareas extraordinarias de soporte

Anualmente será necesario actualizar la información almacenada en la aplicación, para añadir nuevos presupuestos. Esta tarea se compone de los siguientes pasos:

- Descargar los datos del nuevo presupuesto de Aragón Open Data, y agregarlos al repositorio de gestión de versiones de la aplicación.
- Actualizar los ficheros '`data/inflacion.csv`' y '`data/poblacion.csv`', añadiendo la población de Aragón y la inflación en ese año. (El fichero actualmente contiene los enlaces a las páginas del IAEST donde se actualizan estos datos.)
- Desplegar la nueva versión del repositorio en producción.
- Cargar el nuevo presupuesto en la base de datos, utilizando la tarea '`load_budget[año]`' descrita anteriormente. Si disponemos además de información de ejecución presupuestaria o recaudación podemos cargar ésta utilizando las tareas correspondientes descritas anteriormente.
- Cargar las nuevas estadísticas en la base de datos, utilizando la tarea '`load_stats`' descrita anteriormente.
- Limpiar el caché de la aplicación, para que los nuevos datos sean visibles. La aplicación utilizará el sistema de caché en memoria de Django, que no utiliza ficheros: para limpiar la caché basta por tanto con reiniciar la aplicación.