

Consultoría de Desarrollo e Implementación de Datos Abiertos

Entregable 5

Middleware de Recolección de Datos Abiertos

Manual de usuario

Programa de Democracia y Gobernabilidad

Componente 2: “Rendición de cuentas y esfuerzos anticorrupción fortalecidos en instituciones públicas claves”

Consultor: SODEP S.A.

23/Setiembre/2014

Índice

Tabla de contenido

1	Introducción	3
2	Arquitectura Propuesta para la Catalogación de Datasets.....	5
3	Herramienta para modalidades Data Hunting y Recolecta	7
3.1	Manual de Instalación	9
3.2	Manual de Usuario.....	11
	Extensión para Federación de CKAN	16
3.3	Manual de Instalación	16
3.4	Descripción de Funcionalidades	21
3.5	Federación de Catálogos CKAN.....	28
4	Control de Versiones de Datasets en CKAN	31
4.1	Control de Versiones General del Catálogo de Datos	31
4.2	Control de Versiones General de un Dataset	33
4.3	Ejemplo Práctico de Control de Cambios de un Dataset.....	34

1 Introducción

El presente documento tiene como objetivo presentar el *Entregable 5: “Middleware de Recolección de Datos Abiertos: Manual de Usuario”* desarrollado como parte del proyecto de Desarrollo e Implementación de Datos Abiertos correspondiente al contrato de servicios Nro. CS-2014-018-C2 - Datos Abiertos, en el marco del Componente 2 “Rendición de cuentas y esfuerzos anticorrupción fortalecidos en instituciones públicas claves” del Programa de Democracia y Gobernabilidad (USAID-CEAMSO).

El Catálogo Nacional de Datos Abiertos Gubernamentales es un portal web destinado a la publicación de datos abiertos gubernamentales, para su consulta y utilización por parte de los ciudadanos. Como apoyo al proceso de apertura de datos, se realizó la implementación de un middleware que automatice la recolección de metadatos para la catalogación de datasets de otras instituciones.

Según el contrato de servicios Nro. CS-2014-018-C2 se identifican tres modalidades de recolección de metadatos:

1. Modalidad Data Hunting: en la cual se realiza scraping, crawling y parsing de páginas HTML que incluyen metadatos correspondientes a los datasets de acuerdo a especificaciones como microdata¹ y rdfa².
2. Modalidad Recolecta: en la cual se obtienen los metadatos a partir de un archivo data.json, que se encuentra en la raíz del sitio web que contiene los datasets. El contenido del archivo data.json describe el catálogo de datasets de acuerdo al formato propuesto por el Project Open Data³, del Gobierno de los Estados Unidos de Norteamérica.
3. Modalidad Federada: en la cual los datasets se obtienen a partir de otra instancia de CKAN, operando ambos catálogos de datos entre sí a través de la API REST de CKAN.

Las modalidades de Data Hunting y Recolecta se implementan mediante una herramienta que presenta una interfaz por línea de comandos, la cual podrá ser utilizada por el administrador del Catálogo Nacional de Datos Abiertos Gubernamentales para importar los datasets al mismo.

Entre los componentes software utilizados para el desarrollo de esta herramienta se encuentran:

¹ <http://www.w3.org/TR/microdata/>

² <http://www.w3.org/TR/rdfa-syntax/>

³ <http://project-open-data.github.io/metadata-resources/>

- Scrapy⁴: un framework Python de web crawling y scraping que facilita el recorrido automático de páginas web.
- rdflib⁵: librería Python utilizada para parsear las etiquetas con formato rdfa.
- microdata⁶: librería Python utilizada para parsear las etiquetas en formato microdata.
- Requests⁷: librería Python que provee una abstracción sobre las solicitudes HTTP, utilizada para interactuar con la API REST de CKAN.

La modalidad Federada se implementa mediante la instalación y modificación de la extensión ckanext-harvest, que se encuentra disponible como proyecto de código abierto en Github.

Las herramientas desarrolladas facilitan la catalogación de datasets y fomentan el control de calidad de la información disponible en el Catálogo Nacional de Datos Abiertos Gubernamentales, mediante un proceso de actualización colaborativa entre la SENATICS y las demás instituciones del estado.

El código fuente de las mismas puede encontrarse en Github, en los siguientes repositorios:

- <https://github.com/SENATICS/DataCrawler>
- <https://github.com/ckan/ckanext-harvest/tree/stable>

Este documento se organiza de la siguiente manera:

- En primer lugar, se presentan los distintos escenarios posibles para el problema de la catalogación automatizada de datasets y la arquitectura general de la solución propuesta en cada caso.
- Posteriormente, se describe la herramienta implementada para las modalidades de data hunting y recolecta, incluyendo guías de instalación y uso de la misma.
- Seguido, se presenta la extensión de federación de CKAN. De igual manera, se incluyen los pasos de instalación, configuración y uso de esta herramienta.
- Finalmente, se describen las funcionalidades de versionamiento de datasets de CKAN, las cuales pueden ser utilizadas por el administrador del catálogo para el control de los cambios realizados por las herramientas de catalogación.

⁴ <http://scrapy.org/>

⁵ <https://rdflib.readthedocs.org/en/latest/>

⁶ <https://github.com/edsu/microdata>

⁷ <http://docs.python-requests.org/en/latest/>

2 Arquitectura Propuesta para la Catalogación de Datasets

La arquitectura del sistema fue desarrollada por el Responsable de Datos Abiertos del Programa de Democracia y Gobernabilidad. Esta arquitectura se incluye en el presente documento por motivos de completitud.

El problema de la catalogación automatizada de datasets presenta características y desafíos técnicos diversos, de acuerdo a las precondiciones que cumplen los sitios web a partir de los cuales se importan los datos al Catálogo Nacional de Datos Abiertos Gubernamentales disponible en los servidores de la SENATICS en la URL <http://datos.gov.py>. Dicho catálogo está basado en la herramienta de código abierto CKAN⁸.

La siguiente figura presenta los tres escenarios posibles que se consideraron para la implementación de las herramientas que forman parte de este entregable.

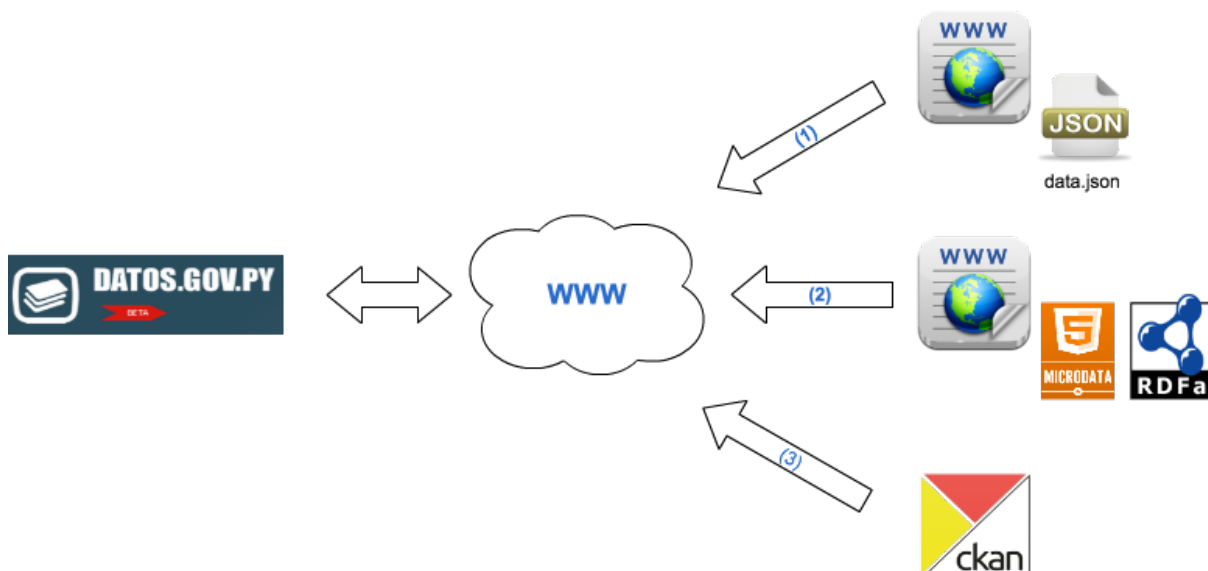


Figura 1. Posibles escenarios de catalogación automatizada de datasets

En el primer escenario, el sitio web del cual se importan los datasets al Catálogo Nacional de Datos Abiertos Gubernamentales expone en su directorio raíz un archivo data.json que describe el contenido del mismo. Este archivo incluye metadatos acerca de los datasets que se encuentran en el portal web, de acuerdo a la especificación propuesta por el Project Open Data.

En este caso, la herramienta por línea de comandos desarrollada obtiene el archivo data.json mediante una solicitud GET del protocolo HTTP al servidor donde se aloja el sitio web. Posteriormente, cada dataset especificado en el archivo se persiste o actualiza (en caso de

⁸ <https://github.com/SENATICS/ckanext-opendatagovpy>

haber sido registrado previamente) en el catálogo de datos abiertos gubernamentales de la SENATICs utilizando la API REST de CKAN para tal efecto.

El segundo escenario plantea la disponibilidad de un sitio web que describa los datasets que contiene utilizando etiquetas en formato microdata o rdfa. En este caso, y a diferencia del primer escenario, los metadatos correspondientes a los datasets y sus recursos se encuentran distribuidos en las distintas páginas que componen el sitio. El Portal de Datos Abiertos del Ministerio de Educación y Cultura representa un ejemplo de este escenario en particular.

En estas circunstancias, la herramienta por línea de comandos implementada lleva a cabo un proceso de crawling y parsing de las páginas web del sitio en cuestión, de modo a extraer y centralizar los datos correspondientes a los datasets. El resultado de este proceso es un archivo data.json similar al del primer escenario, de modo que la importación a CKAN se realiza de manera idéntica.

En el tercer y último escenario de catalogación, los datasets que desean importarse se encuentran almacenados en otra instancia de CKAN, como por ejemplo, el catálogo de datos del Tribunal Superior de Justicia Electoral del Paraguay disponible en <http://datosabiertos.tsje.gov.py/>. Este caso de uso, en el cual ambos catálogos se comunican a través la API REST de CKAN, recibe el nombre de federación.

Al tratarse de una situación que se presenta con frecuencia al administrar un Catálogo de Datos Abiertos, el equipo de desarrollo de CKAN propone una solución a través de la extensión ckanext-harvest. Esta extensión fue adaptada a los requerimientos de internacionalización y autorización propios del Catálogo Nacional de Datos Abiertos Gubernamentales.

3 Herramienta para modalidades Data Hunting y Recolecta

DataCrawler es una herramienta por línea de comandos encargada de realizar crawling y parsing sobre Portales de Datos Abiertos del Gobierno. Está desarrollada en el lenguaje de programación Python.

La extracción de información de sitios web mediante la utilización de programas de software recibe el nombre de web scraping. El primer paso para realizar scraping consiste en el recorrido de las distintas páginas que componen el sitio a ser procesado, simulando la navegación de un usuario. Esta etapa se conoce como web crawling.

Posteriormente, y por cada página web visitada, se analiza el contenido HTML de las mismas y se extraen los datos relevantes disponibles en las mismas. Estos datos se encuentran incrustados en las páginas HTML, utilizando una sintaxis predefinida como microdata o rdfa. Este paso se denomina web parsing o HTML parsing.

Datacrawler hace uso de las técnicas previamente mencionadas para automatizar el proceso de catalogación de datasets. Para cada dominio de búsqueda, esta herramienta navega por los diferentes enlaces contenidos en las páginas del dominio, extrae la información anotada con microdata o rdfa, devuelve un archivo data.json con la estructura estandarizada de Project Open Data y se encarga de transferir los resultados al Catálogo Nacional de Datos Abiertos Gubernamentales.

La estructura y módulos principales se describen a continuación:

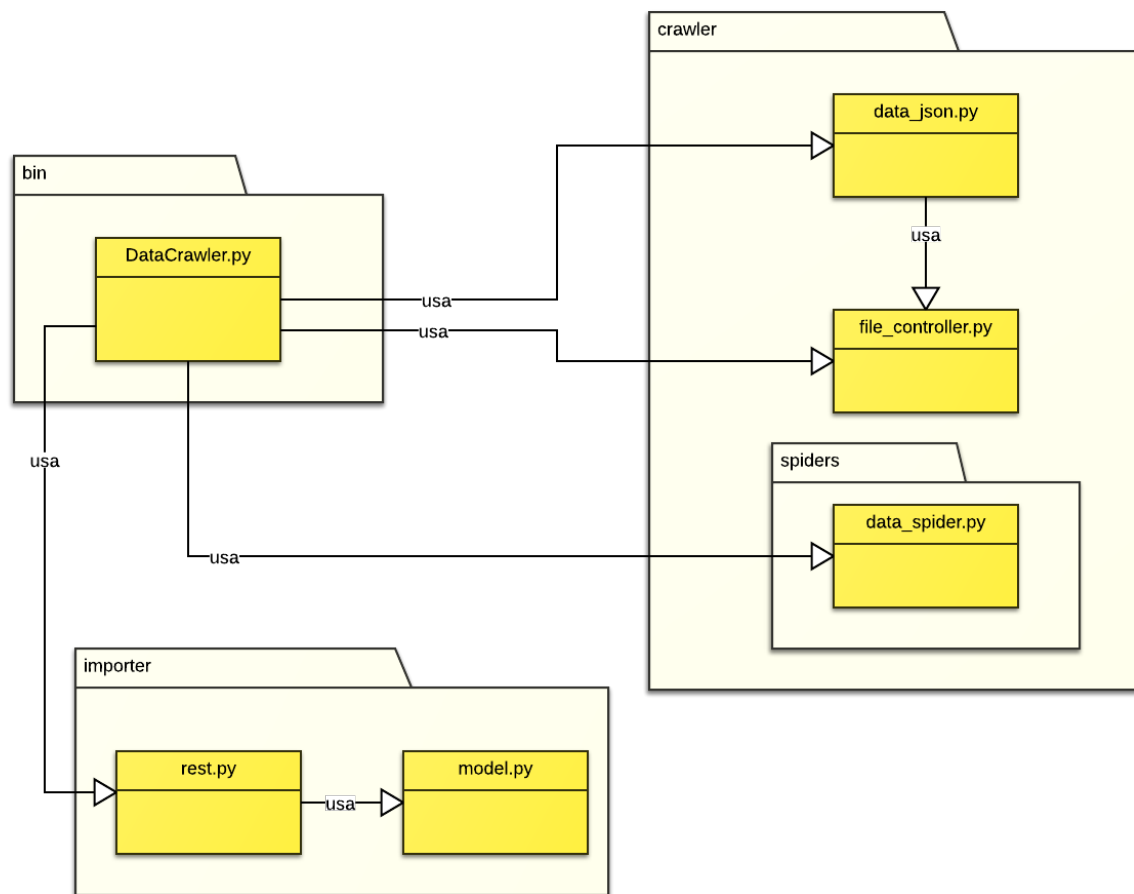


Figura 2. Estructura de la herramienta DataCrawler

DataCrawler: es el módulo principal y se encarga de lanzar el proceso (`data_spider`) que navegará a través de los diferentes enlaces que existen en la/s página/s de los dominios de búsqueda especificados para el crawling. Primero, levanta un servidor splash⁹, el cual tiene la función de renderizar los elementos javascript de las páginas antes de que sean analizadas. Luego lanza el `data_spider` (su estructura y función se explicarán más adelante). A continuación pasa los resultados del `data_spider` al `data_json` (su estructura y función se explicarán más adelante). Para finalizar, se encarga de llamar al módulo `importer` (su estructura y función se explicarán más adelante), para transferir los datos recolectados al catálogo.

data_spider: su función es la de navegar a través de los enlaces de una página a partir de un conjunto de páginas de inicio (`start_urls`) y un conjunto de dominios permitidos (`allowed_domains`). Para cada respuesta ejecuta una función de scraping (`parse_item`) en donde se realizan las operaciones necesarias para extraer los datos. Los datos que se recolectan son aquellos que están anotados ya sea con microdata o con rdfa. Para extraer

⁹ <https://github.com/scrapinghub/splash>

datos anotados con microdata se utiliza la librería `microdata.py`. Para extraer datos anotados con `rdfa`, primero se convierten las anotaciones de `rdfa` a `microdata`, y luego se utiliza la librería `microdata.py`. Finalmente, todos los datos extraídos son almacenados en un archivo `“.json”`.

data_json: su función es parsear el archivo `“.json”` obtenido por el módulo `data_spider`. Por cada ítem del archivo `“.json”`, extrae sus atributos y los copia o convierte si fuera necesario, para finalmente, almacenarlos en un archivo `“data.json”` con la estructura estandarizada de Project Open Data (como utilitario de creación/escritura de archivos utiliza la clase `file_controller.py`).

importer: su función es leer el archivo `“data.json”` proveído por el módulo `data_json`. Por cada uno de los datasets, realiza las conversiones necesarias para transferir los datos al Catálogo Nacional de Datos Abiertos Gubernamentales (para las transformaciones utiliza la clase `model.py`). Para conectarse y realizar cambios en el catálogo requiere la URL y credenciales de acceso válidas (usuario/API Key) del Catálogo.

3.1 Manual de Instalación

Datacrawler requiere un ambiente Python mínimo para su instalación y ejecución. Los pasos de esta guía se probaron para un ordenador con:

- Ubuntu 14.04
- Python 2.7

Las demás dependencias de la herramienta, las cuales se describen brevemente a continuación, se incluirán en el ambiente base como parte del proceso de instalación:

- `microdata.py`: esta librería se utiliza para la extracción de anotaciones `microdata` en páginas `htm/html`.
- `rdflib`: es una librería de Python para trabajar con `RDF`, un lenguaje simple pero potente para representar la información. La herramienta desarrollada utiliza esta librería para transformar anotaciones `RDFa` a `Microdata`, previa conversión a `RDF`.
- `rdf-translator`: es una herramienta de conversión de múltiples formatos de marcado estructurado. Se utiliza junto con `rdflib` para la transformación de páginas `htm/html` anotadas con `rdfa` a `microdata`.
- `requests`: esta librería permite enviar peticiones `HTTP`. Permite agregar encabezados, datos de formularios y parámetros con los diccionarios de Python, así como acceder a los datos de respuesta de la misma forma. Se utiliza para obtener archivos

data.json en caso de que los Portales ya lo tengan publicado. También es utilizada por el módulo “importer” para la creación y o actualización de datasets en el Catálogo Nacional de Datos Abiertos Gubernamentales.

- Scrapy: es un framework que provee funcionalidades para realizar web crawling y scraping de alto nivel, es utilizado para rastrear sitios web y extraer datos estructurados de sus páginas. Se puede utilizar para una amplia gama de propósitos, desde data-mining hasta monitoreo y testeo automatizado. La herramienta desarrollada utiliza para su funcionamiento los métodos y funciones ofrecidas por esta librería.
- splash: es un servicio de ejecución de javascript con una API HTTP. Este servidor debe levantarse antes de que el módulo data_spider se ponga en funcionamiento. Su función es renderizar las páginas con javascript antes de pasar la respuesta a la librería microdata.py (encargada de extraer las anotaciones Microdata).
- twisted: implementa una variedad de protocolos de comunicación y redes y los expone como llamadas a métodos de los objetos Python. Implementaciones cliente y servidor se proporcionan para los distintos protocolos estándar, incluyendo: HTTP, DNS, POP, IMAP, SSH, entre otros. Esta librería es un requerimiento tanto para la librería Scrapy como para el servidor splash.
- PyQt: es una alternativa Python para la programación de Interfaces Gráficas de Usuario (GUI). Esta librería es un requerimiento para el servidor splash.
- click: es un kit de creación de interfaces de línea de comandos, de manera que sean más estéticamente agradables y fácilmente configurables.
- lxml: potente librería de procesamiento de XML. Esta librería es un requerimiento para la librería Scrapy.

Para instalar el DataCrawler deben seguirse los siguientes pasos:

3.1.1 Crear un entorno virtual:

```
virtualenv datacrawler
```

3.1.2 Activar el entorno virtual:

```
source path_to_datacrawler/bin/activate
```

3.1.3 Clonar el repositorio:

```
git clone https://github.com/SENATICS/DataCrawler.git
```

3.1.4 Instalar el módulo:

```
cd DataCrawler
```

```
install.sh path_to_your_virtualenv
```

```
python setup.py develop
```

3.2 Manual de Usuario

A continuación se presenta la serie de pasos a ser realizados por el administrador del Catálogo Nacional de Datos Abiertos Gubernamentales para la importación. Para este ejemplo se asumirá que:

- El Catálogo Nacional de Datos Abiertos Gubernamentales se encuentra alojado en `datos.gov.py`.
- La siguiente es una API Key válida de CKAN:
`1a2b3456-c7d8-91ef-a234-b567cd891e23`

Una API KEY válida de CKAN puede obtenerse en el perfil de un usuario administrador, al cual se ingresa haciendo clic en el ícono del usuario que se encuentra en la barra superior del sitio.



Figura 2. Ícono del Usuario en CKAN

Nombre de usuario

senatics

Dirección de correo electrónico Privado

None

Miembro Desde

Junio 18, 2014

Estado

active

Clave API Privado32f7c255-115a-473d-a0e7-
7d956ec27947

Figura 3. API Key en perfil de Usuario Administrador

- Existen datos anotados con microdata en algunas páginas del dominio mec.gov.py.

El entorno de ejecución en el que se probaron estos comandos es similar al entorno de instalación, esto es:

- Ubuntu 14.04
- Python 2.7

3.2.1 Activar el entorno virtual:

```
source path_to_datacrawler/bin/activate
```

3.2.2 Renombrar el archivo settings-example.py por settings.py y modificar los valores especificados más abajo según las configuraciones locales:

SPLASH_URL: URL donde se levanta el servidor splash

CATALOG_URL: URL del Catálogo Nacional de Datos Abiertos Gubernamentales

API_KEY: API Key del Catálogo Nacional de Datos Abiertos Gubernamentales

Ejemplo

`SPLASH_URL: 'http://localhost:8050/'`

`CATALOG_URL: 'http://datos.gov.py/api/3/action/'`

`API_KEY: '1a2b3456-c7d8-91ef-a234-b567cd891e23'`

- 3.2.3** Crear el archivo con la lista de dominios sobre los cuales se realizará el crawling. La lista debe contener los diferentes dominios, separados por saltos de línea.

Ejemplo

Nombre del archivo: `domains.txt`

Contenido:

`mec.gov.py`

- 3.2.4** Ejecutar el siguiente comando en una terminal:

```
python DataCrawler/bin/DataCrawler.py
```

```
--file path_to_your_file_with_domains_to_crawl
```

```
--virtualenv path_to_your_virtual_enviroment
```

Parámetros:

- `path_to_your_file_with_domains_to_crawl`: ruta absoluta a la ubicación del archivo que contiene la lista de los dominios sobre los cuales se realizará el crawling.
- `path_to_your_virtual_enviroment`: ruta absoluta a la ubicación del entorno virtual donde se instaló el DataCrawler.

Resultados de la ejecución

Para cada dominio de la búsqueda se crea un directorio con el nombre del dominio, el cual contiene dentro un archivo “data.json” con la estructura propuesta por Project Open Data. A continuación se incluye un extracto del archivo data.json generado a partir del Portal de Datos Abiertos del Ministerio de Educación y Cultura, el cual fue utilizado para el ejemplo de ejecución.

```
[
  {
    "description": "Contiene informaci3n sobre la ubicaci3n geogr3fica de
los establecimientos escolares tales como el departamento,
distrito,barrio/localidad donde se encuentran asentados, adem3s de la zona
(urbana o rural) a la cual pertenecen, y los datos georreferenciadas como las
coordenadas planas (en metros) y geograficas. El Sistema de informaci3n de
Estadística Continua (SIEC) considera "Establecimiento Escolar" a la
construcci3n que existe dentro de un predio (terreno) que se emplea para la
enseñanza, donde puede funcionar una o m3s instituciones educativas con sus
respectivos niveles/modalidades de educaci3n.",
    "contactName": "Juan Barrios",
    "accessLevel": "public",
    "publisher": "Ministerio de Educaci3n y Cultura",
    "landingPage": "http://datos.mec.gov.py/data/establecimientos",
    "keyword": [
      "educaci3n",
      "establecimientos",
      "escolar",
      "geografico"
    ],
    "license": "https://creativecommons.org/licenses/by/4.0/legalcode",
    "title": "ESTABLECIMIENTOS ESCOLARES",
    "temporal": "2012-01-01/2012-12-31",
    "version": "1.0",
    "distribution": [
      {
        "accessURL":
"https://mega.co.nz/#!x0InHDpZ!5VZp63YKAvfP3qm2absgZO-kpqK-qEYXxnjqePcRf-Q",
        "format": "pdf"
      },
      {
        "accessURL": "/data/establecimientos_2012.xlsx",
        "format": "xls"
      },
      {
        "accessURL": "/data/establecimientos_2012.csv",
        "format": "csv"
      },
      {
        "accessURL": "/data/establecimientos_2012.json",
        "format": "json"
      }
    ],
    "mbox": "datosabiertos@mec.gov.py"
  }
]
```

Figura 4. Extracto de data.json correspondiente al Ministerio de Educaci3n y Cultura

Por cada uno de los archivos generados se realiza el proceso de conversión para transferirlos al Catálogo Nacional de Datos Abiertos Gubernamentales. Por cada dataset encontrado, la herramienta por línea de comandos interactúa con el usuario final preguntando si se desea o no crear/actualizar el dataset en el Catálogo Nacional de Datos Abiertos Gubernamentales.

```
,
Dataset: ESTABLECIMIENTOS ESCOLARES

valid_until: 2012-12-31
maintainer: Juan Barrios
tags: educación, establecimientos, escolar, geografico, data-hunting
private: True
maintainer_email: datosabiertos@mec.gov.py
modalidad: recolecta
valid_from: 2012-01-01
name: establecimientos-escolares
license: cc-by
author: Juan Barrios
author_email: datosabiertos@mec.gov.py
notes:
    Contiene información sobre la ubicación geográfica de los establecimientos escolares tales como el departamento, distrito,barrio/localidad donde se encuentran asentados, además de la zona (urbana o rural) a la cual pertenecen, y los datos georreferenciados como las coordenadas planas (en metros) y geograficas. El Sistema de información de Estadística Continua (SIEC) considera "Establecimiento Escolar" a la construcción que existe dentro de un predio (terreno) que se emplea para la enseñanza, donde puede funcionar una o más instituciones educativas con sus respectivos niveles/modalidades de educación.

owner_org: 183d079d-644a-4dc6-ac4e-6fd6294ce208
version: 1.0
title: ESTABLECIMIENTOS ESCOLARES
Recurso Nro. 1
url: https://mega.co.nz/#!84gjFSjT!Tx_gIHhtqFC5LworhkqIxI8nDsrNbfJGS16f27NlQjM
name: ESTABLECIMIENTOS ESCOLARES PDF
format: pdf
Recurso Nro. 2
url: /data/establecimientos_2012.xlsx
name: ESTABLECIMIENTOS ESCOLARES XLS
format: xls
Recurso Nro. 3
url: /data/establecimientos_2012.csv
name: ESTABLECIMIENTOS ESCOLARES CSV
format: csv
Recurso Nro. 4
url: /data/establecimientos_2012.json
name: ESTABLECIMIENTOS ESCOLARES JSON
format: json

Desea crear un nuevo dataset con los valores anteriores? (s/n) ☐
```

Figura 5. Confirmación de Creación de Dataset con DataCrawler

Extensión para Federación de CKAN

La extensión ckanext-harvest¹⁰ proporciona una interfaz por línea de comandos y una interfaz web de usuario para la gestión de fuentes y trabajos de recolección. Una fuente de recolección es básicamente una instancia de CKAN a cuyos datasets se puede acceder mediante la API REST correspondiente. A su vez, un trabajo de recolección es una instancia de un proceso de importación de datasets correspondiente a una determinada fuente de recolección.

Un trabajo de recolección es un proceso asíncrono, para evitar bloquear la interfaz de usuario de CKAN. Los trabajos se almacenan en pilas implementadas utilizando un middleware de comunicación entre procesos, vaciándose las mismas periódicamente mediante la ejecución de tareas programadas a nivel de sistema operativo.

Como parte del trabajo realizado para este entregable, fueron implementados dos cambios importantes para la extensión de federación:

- Internacionalización de la extensión: varias de las etiquetas utilizadas en la interfaz de usuario de la extensión no se encontraban disponibles en español. Las mismas se tradujeron e integraron con la extensión.
- Aprobación de Datasets: por defecto, los datasets importados mediante la extensión de federación mantenían el mismo nivel de visibilidad que en la fuente original. Para el Catálogo Nacional de Datos Abiertos Gubernamentales, los datasets importados son privados por defecto, de modo a pasar por la aprobación del responsable, como se propone en el Manual del Administrador¹¹.

Cabe destacar que la administración de fuentes y trabajos de recolección es una responsabilidad del administrador del Catálogo Nacional de Datos Abiertos Gubernamentales.

3.3 Manual de Instalación

La instalación de esta extensión de CKAN incluye la instalación de sus dependencias y la configuración de las tareas a ser ejecutadas periódicamente por el sistema operativo. La serie de pasos que se presenta a continuación toma como base una instalación de CKAN realizada de acuerdo al documento de Puesta en Producción de CKAN¹², la cual incluye:

¹⁰ <https://github.com/ckan/ckanext-harvest>

¹¹ <https://github.com/SENATICS/ckanext-opendatagovpy/wiki/Manual-del-Administrador:-Gestión-de-Usuarios-y-Datasets-aprobación-de-un-dataset>

¹² <https://github.com/SENATICS/ckanext-opendatagovpy/wiki/Puesta-en-Producción-de-CKAN>

- Ubuntu 14.04
- Python 2.7
- CKAN 2.2

Habiendo completado la guía de puesta en producción, deben seguirse estos pasos para la instalación de la extensión de federación:

3.3.1 CKAN y sus dependencias se instalan haciendo uso de un entorno virtual, de modo a aislar el ambiente de ejecución de las demás aplicaciones Python que pudiesen alojarse en el mismo servidor. Antes de realizar cualquier cambio sobre CKAN o alguna extensión es necesario activar el entorno virtual, lo cual se logra ejecutando el siguiente comando:

```
. /usr/lib/ckan/default/bin/activate
```

Para más información acerca de los entornos virtuales Python, su uso y configuración, se recomienda leer: <http://virtualenv.readthedocs.org/en/latest/>

3.3.2 Instalar el middleware de comunicación entre procesos, necesario para la ejecución asíncrona de los trabajos de recolección. En este caso se utiliza RabbitMQ¹³, aunque la extensión admite también su integración con Redis¹⁴. Esta dependencia se instala ejecutando el siguiente comando:

```
sudo apt-get install rabbitmq-server
```

Para más información acerca de RabbitMQ se recomienda consultar la documentación oficial disponible en: <http://www.rabbitmq.com/>

3.3.3 Clonar e instalar la extensión ckanext-harvest, en el directorio correspondiente a las extensiones CKAN. Esto puede realizarse mediante los siguientes pasos:

```
cd /usr/lib/ckan/default/src/  
pip install -e git+https://github.com/SENATICS/ckanext-  
harvest.git@stable#egg=ckanext-harvest  
cd ckanext-harvest  
pip install -r pip-requirements.txt
```

¹³ <http://www.rabbitmq.com/>

¹⁴ <http://redis.io/>

- 3.3.4** Actualizar la extensión ckanext-opendatagovpy, la cual incluye un harvester personalizado para el Catálogo Nacional de Datos Abiertos Gubernamentales.

```
cd ../ckanext-opendatagovpy
git pull
```

- 3.3.5** Incluir la extensión instalada entre las extensiones activas, añadiéndola a la línea correspondiente en el archivo de configuración de CKAN, que puede encontrarse en /etc/ckan/default/production.ini

```
ckan.plugins = ... harvest paraguay_harvester
```

- 3.3.6** Ejecutar el script para la creación de las tablas necesarias en la base de datos de CKAN:

```
paster --plugin=ckanext-harvest harvester initdb --
config=/etc/ckan/default/production.ini
```

- 3.3.7** La extensión ckanext-harvest requiere la ejecución de dos procesos adicionales que deben ejecutarse como servicios. Para lograr esto, se utiliza la herramienta de administración y monitorio de procesos Supervisor, la cual puede instalarse utilizando el siguiente comando:

```
sudo apt-get install supervisor
```

- 3.3.8** Para verificar que Supervisor se encuentre funcionando correctamente, ejecutar el siguiente comando:

```
ps aux | grep supervisord
```

La salida en consola de este comando debe ser similar a:

```
root      9224  0.0  0.3  56420 12204 ?        Ss      15:52
0:00 /usr/bin/python /usr/bin/supervisord
```

- 3.3.9** Crear el archivo de configuración de Supervisor, /etc/supervisor/conf.d/ckan_harvesting.conf, con el siguiente contenido:

```
[program:ckan_gather_consumer]
```

```
command=/usr/lib/ckan/default/bin/paster --plugin=ckanext-  
harvest harvester gather_consumer --  
config=/etc/ckan/default/production.ini
```

```
; user that owns virtual environment.  
user=www-data
```

```
numprocs=1  
stdout_logfile=/var/log/ckan/std/gather_consumer.log  
stderr_logfile=/var/log/ckan/std/gather_consumer.log  
autostart=true  
autorestart=true  
startsecs=10
```

```
[program:ckan_fetch_consumer]
```

```
command=/usr/lib/ckan/default/bin/paster --plugin=ckanext-  
harvest harvester fetch_consumer --  
config=/etc/ckan/default/production.ini
```

```
; user that owns virtual environment.  
user=www-data
```

```
numprocs=1  
stdout_logfile=/var/log/ckan/std/fetch_consumer.log  
stderr_logfile=/var/log/ckan/std/fetch_consumer.log  
autostart=true  
autorestart=true  
startsecs=10
```

3.3.10 Crear los archivos necesarios para el logging de los procesos, mediante los siguientes pasos:

```
sudo mkdir -p /var/log/ckan/std/
```

```
sudo touch /var/log/ckan/std/gather_consumer.log
sudo chown www-data /var/log/ckan/std/gather_consumer.log
sudo touch /var/log/ckan/std/fetch_consumer.log
sudo chown www-data /var/log/ckan/std/fetch_consumer.log
```

- 3.3.11** Habiendo culminado la configuración, arrancar las tareas Supervisor utilizando los siguientes comandos:

```
sudo supervisorctl reread
sudo supervisorctl add ckan_gather_consumer
sudo supervisorctl add ckan_fetch_consumer
sudo supervisorctl start ckan_gather_consumer
sudo supervisorctl start ckan_fetch_consumer
```

- 3.3.12** Verificar que los procesos correspondientes a ckanext-harvest se encuentren en ejecución utilizando el siguiente comando:

```
sudo supervisorctl status
```

La salida en terminal de este comando debe ser similar a:

```
ckan_fetch_consumer    RUNNING      pid 6983, uptime 0:22:06
ckan_gather_consumer   RUNNING      pid 6968, uptime 0:22:45
```

Para más información acerca de Supervisor, su utilización y los parámetros de configuración que admite, se recomienda recurrir a la documentación oficial del proyecto: <http://supervisord.org/index.html>

- 3.3.13** Crear o editar el archivo crontab para el usuario con el cual se configuraron los procesos Supervisor, a fin de incluir la tarea periódica que verificará la existencia de trabajos de recolección de datasets:

```
sudo crontab -e -u www-data
```

- 3.3.14** Añadir el siguiente contenido al archivo cron recientemente abierto:

```
*/15 * * * * /usr/lib/ckan/default/bin/paster
--plugin=ckanext-harvest harvester run
--config=/etc/ckan/default/production.ini
```

Este ejemplo en particular, verificará la pila de trabajos pendientes de ejecución cada 15 minutos.

En general, para más información sobre la extensión ckanext-harvest, sus dependencias, detalles de configuración y otros aspectos técnicos de la herramienta se recomienda consultar: <https://github.com/ckan/ckanext-harvest>

3.4 Descripción de Funcionalidades

La instalación de la extensión ckanext-harvest añade un conjunto de vistas al Catálogo Nacional de Datos Abiertos Gubernamentales, entre los cuales pueden mencionarse:

- 3.4.1** Listado de Fuentes de Recolección: página en la cual se observa un listado completo de las fuentes de recolección disponibles para el Catálogo, pudiendo filtrarse las mismas de acuerdo a su periodicidad de actualización y al tipo de recolección que realizan.

Además, es posible realizar filtrado mediante búsqueda textual sobre la lista y ordenar los resultados por criterios como el orden alfabético y la fecha de modificación de la fuente.



Figura 3. Listado de fuentes de recolección

- 3.4.2** Listado de Datasets por Fuente de Recolección: al hacer clic en una fuente de recolección en particular, el usuario puede visualizar la lista de datasets que han sido importados de la misma.



Figura 4. Listado de datasets por fuente de recolección

- 3.4.3** Creación de Fuentes de Recolección: haciendo clic en el botón “Agregar Fuente de Recolección”, el usuario accede a un formulario de creación de fuentes de recolección con los siguientes campos:

- URL de la instancia de CKAN de la cual se desean importar los datasets.
- Título descriptivo de la fuente de recolección.
- Descripción breve de la fuente de recolección.
- Tipo de Fuente de Recolección: se refiere al origen de los datos, a partir del cual se importan los datasets. Como parte de este entregable se ha implementado el soporte para importación de otros catálogos CKAN, que recibe el nombre de “Paraguay CKAN Harvester” en el formulario.

Sin embargo, es posible añadir soporte para otro tipo de fuentes de recolección a través de la instalación de extensiones adicionales como `ckanext-spatial`¹⁵ o mediante el desarrollo de nuevos plugins que implementen la interfaz determinada¹⁶.

- Frecuencia de Actualización: que puede ser manual o periódica (diaria, semanal, mensual, etc.).
- Configuración, que se representa mediante un objeto JSON. Los detalles de la sintaxis del objeto de configuración, así como algunos valores recomendados, se

¹⁵ <https://github.com/ckan/ckanext-spatial>

¹⁶ <https://github.com/ckan/ckanext-harvest-the-harvesting-interface>

describen más adelante. Para una referencia completa, se recomienda consultar la documentación correspondiente¹⁷.

- Organización a la cual pertenece la fuente de recolección.

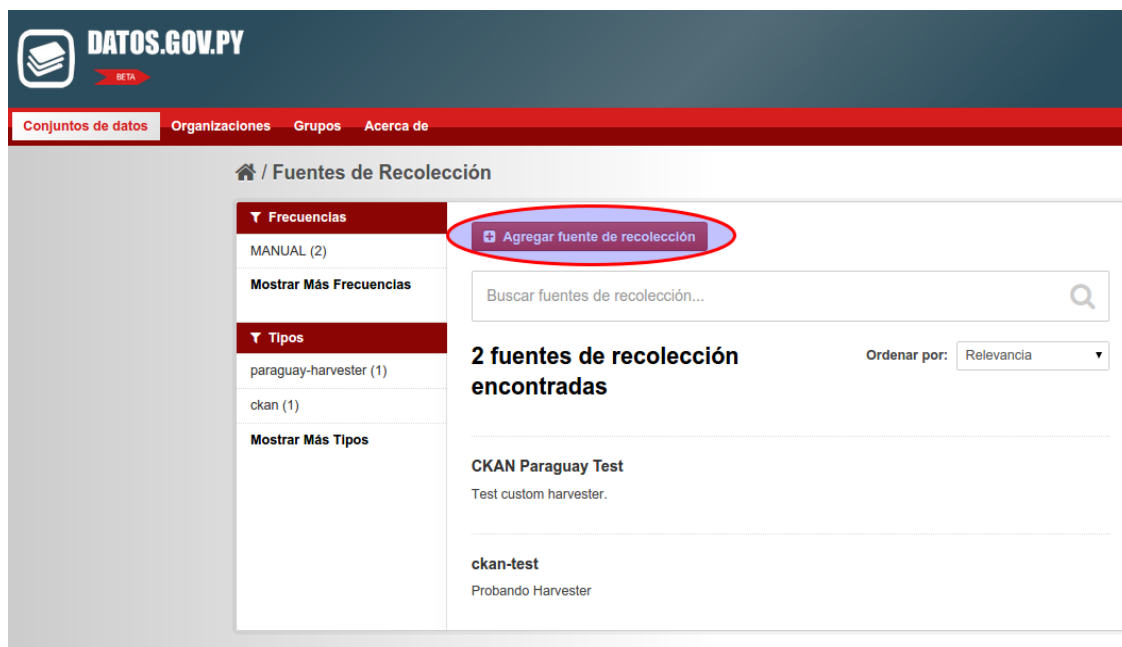


Figura 5. Botón para añadir una nueva fuente de recolección

¹⁷ <https://github.com/ckan/ckanext-harvest - the-ckan-harvester>

[Inicio](#) / Fuentes de Recolección / **Crear Fuente de Recolección**

Fuentes de recolección

Las fuentes de recolección permiten importar metadatos remotos a este catálogo. Fuentes remotas pueden ser otros catálogos así como otras instancias de CKAN, servidores CSW o Web Accessible Folders (WAF) (depende de la fuente de recolección actual activada para esta instancia).

URL:


Debe incluir la sección http:// de la URL

Título:

URL: [Editar](#)

Descripción:

Puede utilizar **formato de marcado** aquí

Tipo de fuente: ☒ Paraguay CKAN Harvester 

Frecuencia de actualización:

Configuración:

Organización:

[Guardar](#)

Figura 6. Formulario de creación de una nueva fuente de recolección

3.4.4 Administración de Fuentes de Recolección: en el listado de datasets por fuente de recolección, haciendo clic en el botón “Administrador”, el usuario puede acceder a un conjunto de funcionalidades de administración.

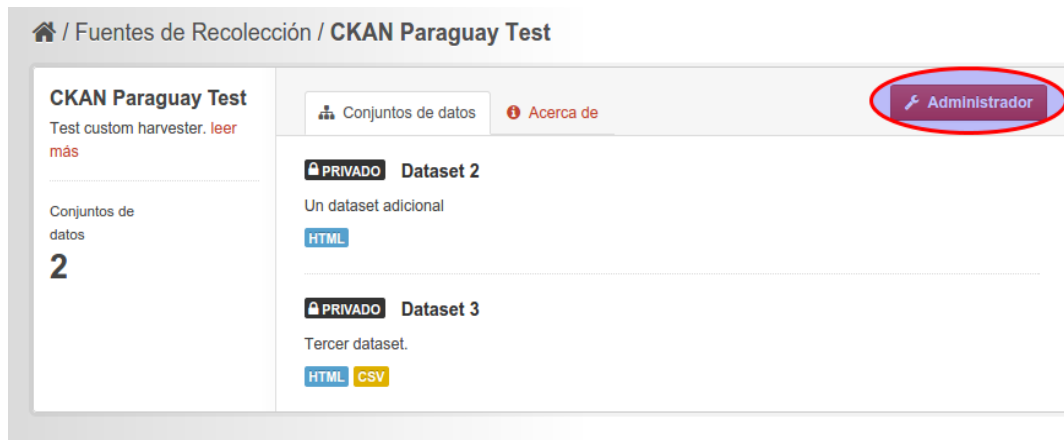


Figura 7. Botón para administrar una fuente de recolección

Esta vista de administración ofrece tres modos, los cuales se exponen al usuario como pestañas:

- Pizarra: donde se muestra un resumen del último trabajo de recolección incluyendo:
 - Cantidad de errores de importación.
 - Cantidad de datasets creados, actualizados y eliminados.
 - Identificador único del trabajo.
 - Etiquetas de tiempo de la creación del trabajo e inicio y fin de la ejecución del mismo.
 - Estado del trabajo, que puede ser finalizado, en ejecución, etc.

🏠 / Fuentes de Recolección / CKAN Paraguay Test / Administrador

CKAN Paraguay Test

Test custom harvester. [leer más](#)

Conjuntos de datos

2

[Recolectar](#) [Limpiar](#) [Ver fuente de recolección](#)

[Pizarra](#) [Trabajos](#) [Editar](#)

Último Trabajo de Recolección

0 errores 0 agregados 0 actualizados 0 eliminado

Id	a8785e26-c4fb-46f9-b9ec-2c410d3187a2
Creado	11 Septiembre, 2014, 14:54
Iniciado	11 Septiembre, 2014, 15:00
Finalizado	
Estado	Finalizado

[Ver informe completo del trabajo](#)

Figura 8. Pizarra de administración de una fuente de recolección

- **Trabajos:** donde se puede acceder a un resumen similar al modo pizarra, pero para el historial completo de trabajos de la fuente de recolección.

🏠 / Organizaciones / Tribunal Superior de Justicia Electoral (TSJE) / Fuentes de Recolección / tsje / Administrador

tsje

Datos Abiertos del Tribunal Superior de Justicia Electoral

[leer más](#)

Conjuntos de datos

8

[Recolectar](#) [Limpiar](#) [Ver fuente de recolección](#)

[Pizarra](#) [Trabajos](#) [Editar](#)

Trabajos de Recolección

Trabajo: d5e7a4da-0310-4796-a97c-07515979b02c

Iniciado: 12 Septiembre, 2014, 17:06 — Finalizado: 12 Septiembre, 2014, 17:07

8 agregados 0 actualizados 0 eliminado

Figura 6. Listado de Trabajos de una Fuente de Recolección

- **Editar:** modo en el que se permite al usuario modificar los valores establecidos al momento de la creación de la fuente de recolección.

🏠 / Organizaciones / Tribunal Superior de Justicia Electoral (TSJE) / Fuentes de Recolección / tsje / Administrador

tsje

Datos Abiertos del Tribunal Superior de Justicia Electoral

[leer más](#)

Conjuntos de datos

8

Recolectar
Limpiar
Ver fuente de recolección

Pizarra
Trabajos
Editar

URL:

Debe incluir la sección http:// de la URL

Título:

URL: [Editar](#)

Descripción:

Puede utilizar [formato de marcado](#) aquí

Tipo de fuente: ☒ Paraguay CKAN Harvester [?](#)

Frecuencia de actualización:

Configuración:

Organización:

Borrar
Guardar

Figura 7. Formulario de Edición de una Fuente de Recolección

Además, un menú de botones en la parte superior de la vista de administración ofrece al usuario las siguientes funcionalidades:

- **Recolectar:** permite al administrador generar una nueva tarea de recolección.
- **Limpiar:** elimina todas las tareas finalizadas y los datasets correspondientes a la fuente de recolección. Además, cancela las tareas en ejecución.
- **Ver fuente de recolección:** redirige al usuario al listado de datasets por fuente de recolección.



Figura 9. Botones con las acciones que pueden realizarse sobre una fuente de recolección

3.5 Federación de Catálogos CKAN

Haciendo uso de las funcionalidades que añade ckanext-harvest, es posible realizar importaciones de datasets que se encuentren disponibles en otras instancias de CKAN mediante los siguientes pasos:

- 3.5.1** Crear una nueva fuente de recolección: completar los campos del formulario de creación según resulte apropiado. En particular, se sugiere el siguiente objeto JSON como valor para el atributo configuración:

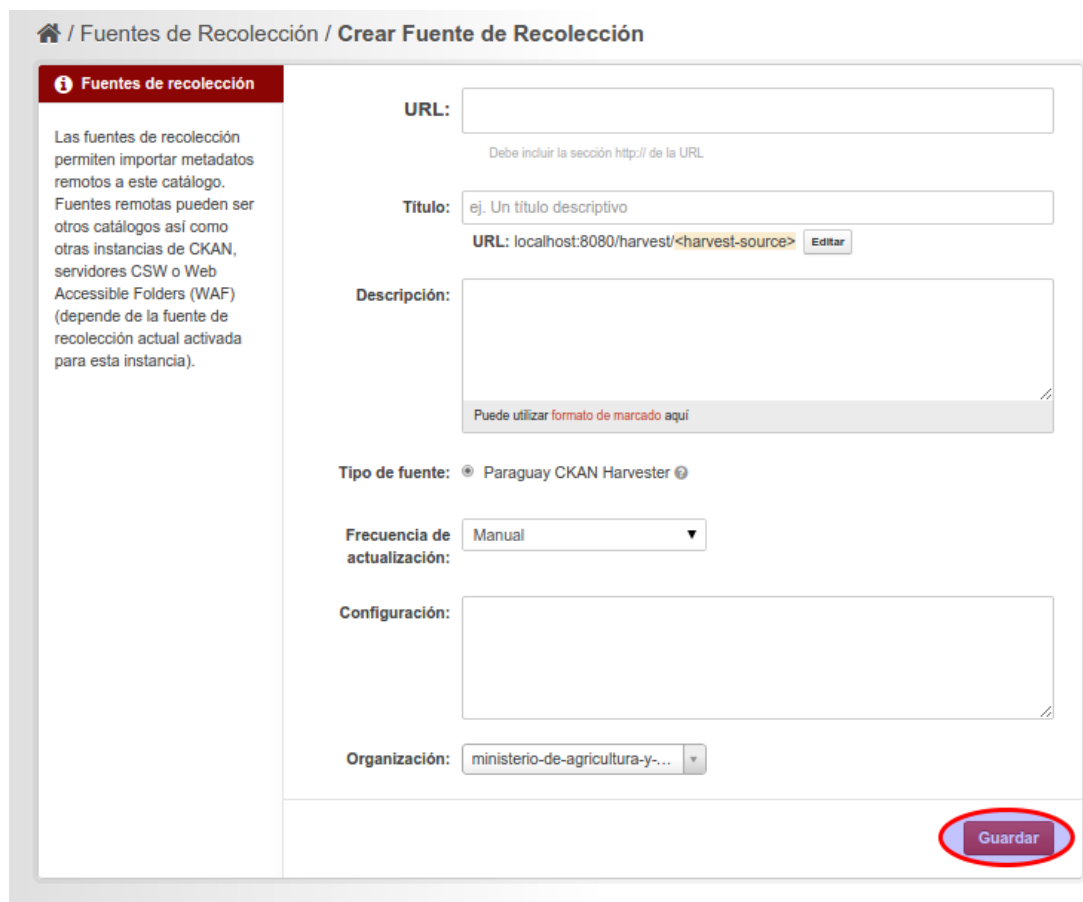
```
{
  "default_tags": ["federada"],
  "remote_groups": "only_local",
  "remote_orgs": "only_local",
  "default_extras":
    {"harvest_url":
      "{harvest_source_url}/dataset/{dataset_id}"
    },
  "user": "federador",
  "api_key": "clave-api-del-usuario"
}
```

Esta configuración en particular especifica lo siguiente:

- Todos los datasets importados de esta fuente de recolección, a través de la extensión, tendrán la etiqueta “federada” para facilitar su identificación.
- Los valores correspondientes al grupo y organización del dataset se importarán únicamente si en el CKAN destino existen un grupo y una organización idéntica.
- La URL original de un dataset importado se almacenará en el atributo extra “harvest_url” del mismo.
- La importación se llevará a cabo con el usuario “federador”.
- La clave de acceso a la API del usuario “federador” es “clave-api-del-usuario”.

Es importante que la SENATICS cree un usuario exclusivo para esta modalidad de catalogación y que sea este el usuario que se especifique en la configuración de la extensión.

Existen atributos adicionales que pueden incluirse en el campo de configuración de la fuente de recolección. Para una referencia completa de estos atributos, consultar: <https://github.com/ckan/ckanext-harvest - the-ckan-harvester>



Fuentes de Recolección / Crear Fuente de Recolección

Fuentes de recolección

Las fuentes de recolección permiten importar metadatos remotos a este catálogo. Fuentes remotas pueden ser otros catálogos así como otras instancias de CKAN, servidores CSW o Web Accessible Folders (WAF) (depende de la fuente de recolección actual activada para esta instancia).

URL:

Debe incluir la sección http:// de la URL

Título:

URL: [Editar](#)

Descripción:

Puede utilizar formato de marcado aquí

Tipo de fuente: ☒ Paraguay CKAN Harvester

Frecuencia de actualización:

Configuración:

Organización:

Guardar

Figura 10. Formulario de creación de una nueva fuente de recolección

3.5.2 Una vez creada la fuente de recolección, desde la vista de administración, presionar el botón “Recolectar”. Suponiendo que la nueva tarea se crea a las 14:08, y la periodicidad establecida en el archivo cron es de 15 minutos, es de esperarse lo siguiente:

- La creación de la tarea se registrará a las 14:08.
- El inicio de la tarea se registrará a las 14:15.
- La finalización de la tarea, con la posibilidad de ver los resultados finales, se registrará a las 14:30.



Figura 11. Botón para ir a la administración de una fuente de recolección



Figura 12. Botón para iniciar el trabajo de recolección



Figura 13. Reporte del trabajo de recolección

4 Control de Versiones de Datasets en CKAN

Las herramientas implementadas permiten la catalogación automatizada de datasets al Catálogo Nacional de Datos Abiertos Gubernamentales. Sin embargo, la automatización de este proceso conlleva la necesidad de mecanismos de control de cambios que permitan el aseguramiento de la calidad de los datos por parte del administrador del Catálogo.

CKAN provee dos funcionalidades para el control de cambios: la primera es el control de versiones general del catálogo; y la segunda es el control de versiones general de un datasets.

Mantener versionados tanto el catálogo como cada uno de los datasets aporta una serie de ventajas: permite un control detallado de los cambios que se llevan a cabo, a nivel de propiedades de los datasets y el catálogo; además de incluir información del autor y el momento en el que se realizaron los cambios.

4.1 Control de Versiones General del Catálogo de Datos

Para acceder a la pantalla de control de versiones general del catálogo de datos, se debe añadir /revision a la URL del mismo: datos.gov.py/revision

Se puede observar la lista de revisiones del catálogo, y por medio del enlace del número de revisión se puede acceder al detalle de la misma. Los datos que se proveen son: número de revisión, la marca de tiempo (o fecha en que tuvo lugar el cambio), el autor del cambio, a qué entidad se refiere, un mensaje de registro y los cambios realizados.

🏠 / Revisiones

Social				
Google+				
Twitter				
Facebook				

Historial de revisiones

Revisión	Marca de tiempo	Autor	Entidad	Mensaje de registro
4f9f11...	10 Septiembre, 2014, 21:15	 rparra	Matriculaciones por Departamento y Distrito	REST API: actualización de objeto matriculaciones-por-departamento-y-distrito
3f21f3...	10 Septiembre, 2014, 21:15	 rparra	Matriculaciones por Departamento y Distrito	REST API: actualización de objeto matriculaciones-por-departamento-y-distrito
e62d15...	10 Septiembre, 2014, 21:10	 rparra	Matriculaciones por Departamento y Distrito	REST API: actualización de objeto matriculaciones-por-departamento-y-distrito
ba67f9...	10 Septiembre, 2014, 21:10	 rparra	Matriculaciones por Departamento y Distrito	REST API: actualización de objeto matriculaciones-

Figura 14. Lista de revisiones del catálogo de datos

🏠 / Revisiones / 4f9f11c8-02fc-42a6-93fd-26ed51b4181...

Social	
Google+	
Twitter	
Facebook	

Revisión: 4f9f11c8-02fc-42a6-93fd-26ed51b41813

Autor:  rparra

Marca de tiempo: 10 Septiembre, 2014, 21:15

Mensaje de registro:
REST API: actualización de objeto matriculaciones-por-departamento-y-distrito

Cambios

Conjuntos de datos

Etiquetas de los conjuntos de datos

Grupos

Figura 15. Detalle de una revisión del catálogo de datos

4.2 Control de Versiones General de un Dataset

Para acceder a la pantalla de control de versiones general de un dataset, se debe añadir /history antes del nombre del dataset: datos.gov.py/dataset/history/cualquier-dataset

Se puede observar la lista de revisiones de un dataset determinado, y por medio del enlace del número de revisión se puede acceder al detalle de la misma manera que para las revisiones del catálogo.

Una funcionalidad adicional de esta pantalla es la de ver las diferencias entre revisiones, para ello se deben escoger las revisiones sobre las cuales se quiere realizar la comparación y hacer click en el botón “Comparar”.



Home / Organizaciones / Ministerio de Educación y ... / Contrataciones

Contrataciones

Seguidores
0
[Seguir](#)

Organización


Ministerio de Educación y Cultura (MEC)
No existe una descripción para esta organización

Social

[Conjunto de datos](#) [Grupos](#) [Flujo de Actividad](#) [Relacionados](#) [Administrar](#)

	Revisión	Marca de tiempo	Autor	Mensaje de registro
	d914ee...	8 Septiembre, 2014, 23:06	 rparra	
	f8e758...	8 Septiembre, 2014, 22:04	 rparra	REST API: actualización de objeto contrataciones
	4e0d56...	8 Septiembre, 2014, 22:04	 rparra	REST API: actualización de objeto contrataciones
	21526d...	8 Septiembre, 2014, 22:04	 rparra	REST API: Crear objeto contrataciones

[Comparar](#)

Figura 16. Lista de revisiones de un dataset

En la pantalla de comparación de revisiones se puede observar el detalle de los cambios que se realizaron sobre el dataset y sobre sus recursos.

Conjuntos de datos / Contrataciones / Diferencias en las revisiones -

Social

Google+

Twitter

Facebook

Diferencias en las revisiones - - Contrataciones

From: 4e0d564f-5831-4371-8f4d-c76abceb19c9 - 8 Septiembre, 2014, 22:04

To: 21526dbe-56fc-4e70-b548-a8f0a09792c2 - 8 Septiembre, 2014, 22:04

Campo	Diferencias
Resource-a27c-hash	- 597ff8a035523f12c1235cbe2e833801db1162d0 +
Resource-a27c-last_modified	- 2014-09-08 18:04:19.008118 + None
Resource-a27c-mimetype	- application/vnd.openxmlformats-officedocu ment.spreadsheetml.sheet + None
Resource-a27c-size	- 7654 + None

Figura 17. Diferencia entre dos revisiones de un dataset

Ejemplo Práctico de Control de Cambios de un Dataset

Para comprender la utilidad de las funcionalidades de control de versiones de CKAN, se presenta a continuación un caso práctico de aplicación. Con este fin, supongamos la siguiente secuencia de pasos:

- 4.2.1** Se crea una fuente de recolección, correspondiente al Catálogo de Datos Abiertos del Tribunal Superior de Justicia Electoral, con la siguiente configuración:

🏠 / Organizaciones / Tribunal Superior de Justicia Electoral (TSJE) / Fuentes de Recolección / tsje / Administrador

tsje

Datos Abiertos del Tribunal Superior de Justicia Electoral
[leer más](#)

Conjuntos de datos

8

Recolectar
Limpiar
Ver fuente de recolección

Pizarra
Trabajos
Editar

URL:

Debe incluir la sección http:// de la URL

Título:

URL: [Editar](#)

Descripción:

Puede utilizar [formato de marcado](#) aquí

Tipo de fuente: ☒ Paraguay CKAN Harvester ⓘ

Frecuencia de actualización:

Configuración:

Organización:

Borrar
Guardar

Figura 8. Fuente de Recolección del Tribunal Superior de Justicia Electoral

4.2.3 Se ejecuta el proceso de recolección, resultando en la siguiente lista de datasets catalogados de manera privada, a la espera de la aprobación del administrador:

<p>tsje</p> <p>Datos Abiertos del Tribunal Superior de Justicia Electoral</p> <p>leer más</p> <hr/> <p>Conjuntos de datos</p> <p>8</p>	<div> Conjuntos de datos Acerca de Administrador </div> <div> <div>PRIVADO</div> Elecciones Generales 2013 <p>Resultados de las Elecciones Generales 2013</p> <div>CSV XLS</div> </div> <hr/> <div> <div>PRIVADO</div> Otras Elecciones Municipales <p>Otras Elecciones Municipales</p> <div>CSV</div> </div> <hr/> <div> <div>PRIVADO</div> Otras Elecciones Departamentales <div>CSV</div> </div> <hr/> <div> <div>PRIVADO</div> Elecciones Municipales 2010 <div>CSV XLS</div> </div> <hr/> <div> <div>PRIVADO</div> Elecciones Municipales 2006 <p>Elecciones Municipales 2006</p> <div>CSV XLS</div> </div> <hr/> <div> <div>PRIVADO</div> Elecciones Generales 2008 <p>Elecciones Generales 2008</p> </div>
--	---



Figura 9. Lista de datasets importados

- 4.2.5** El administrador decide realizar algunos cambios sobre el dataset “Elecciones 2013” antes de su aprobación. En particular modifica los campos de descripción, licencia y le añade un número de versión al dataset:


🏠 / Organizaciones / Tribunal Superior de Justicia ... / Elecciones Generales 2013 / **Editar**

Elecciones Generales 2013

Seguidores
0




 Editar metadatos  Recursos  Ver conjunto de datos

Título: Elecciones Generales 2013

* URL: www.datos.gov.py/dataset/elecciones-generales-2013 

Descripción: Resultados de las Elecciones Generales 2013

Puede utilizar [formato de marcado aquí](#)

Etiquetas:  2013  elecciones generales  federada

Licencia: Creative Commons Attribu...  Definiciones de licencias e información adicional puede ser encontrada en opendefinition.org

Organización: tribunal-superior-de-justici...

Visibilidad: Privado

Fuente: <http://ejemplo.com/dataset.json>

Versión: 1.0

Figura 10. Edición de Dataset "Elecciones 2013"

4.2.6 Posteriormente, continúa trabajando con el catálogo modificando y aprobando otros datasets. La pregunta que se plantea es la siguiente: ¿Cómo puede saber el administrador los cambios realizados sobre el dataset “Elecciones 2013” de modo a, por ejemplo, comunicarlos al administrador del catálogo del Tribunal Superior de Justicia Electoral?

Para llevar esto a cabo, el administrador puede ingresar a la vista de control de versiones del dataset, que se encuentra en la url:

<http://www.datos.gov.py/dataset/history/elecciones-generales-2013>

Elecciones Generales 2013

Seguidores
0

 Seguir

 **Organización**



Tribunal Superior de Justicia Electoral (TSJE)

No existe una descripción para esta organización

 Administrar

 Conjunto de datos
  Grupos
  Flujo de Actividad
  Relacionados

	Revisión	Marca de tiempo	Autor	Mensaje de registro
	aa9360...	22 Septiembre, 2014, 11:40	 senatics	
	808014...	12 Septiembre, 2014, 17:07	 senatics	REST API: Create object elecciones-generales-2013


 Comparar

Figura 11. Historial de Versiones del Dataset "Elecciones 2013"

4.2.7 Esta vista presenta un resumen del historial de cambios realizados sobre el dataset en cuestión. El administrador puede seleccionar las versiones de su interés, y presionar el botón “Comparar”.





Elecciones Generales 2013

 Seguidores
0
[Seguir](#)

Organización

Tribunal Superior de Justicia Electoral (TSJE)
No existe una descripción para esta organización

Administrar

Conjunto de datos Grupos Flujo de Actividad Relacionados

	Revisión	Marca de tiempo	Autor	Mensaje de registro
	aa9360...	22 Septiembre, 2014, 11:40	 senatics	
	808014...	12 Septiembre, 2014, 17:07	 senatics	REST API: Create object elecciones-generales-2013




Figura 12. Comparar Versiones de "Elecciones 2013"

4.2.8 La vista de comparación presenta entre versiones del dataset presenta un encabezado, donde se identifican las versiones comparadas.

Diferencias en las revisiones - - Elecciones Generales 2013

From: 808014a7-532c-430c-b198-49f54e990485 - 12 Septiembre, 2014, 17:07

To: aa9360ac-cfc6-48af-94d5-bc16fcfc003 - 22 Septiembre, 2014, 11:40

Figura 13. Encabezado de Vista de Comparación de Versiones

Además, se incluye una tabla cuyas filas exponen los cambios realizados para cada uno de los atributos del dataset.

license_id	- notspecified + cc-by
metadata_modified	- 2014-09-12 17:07:36.705966 + 2014-09-22 11:40:11.985692
notes	- Elecciones Generales 2013 + Resultados de las Elecciones Generales 2013
url	- None +
version	- None + 1.0

Figura 14. Tabla de Cambios entre Versiones del Dataset

De este modo, el administrador del Catálogo Nacional de Datos Abiertos Gubernamentales puede controlar de manera detallada los cambios realizados sobre los datasets, así como la fecha y el autor de los mismos.