

Procesado de datos XBRL

El origen de los ficheros CSV y JSON, que sirven como datos de entrada de la aplicación web de **VIBIDA**, proviene de ficheros **XBRL** (*eXtensible Business Reporting Language*), estándar XML para intercambio de información financiera.

Dentro del proyecto se han utilizado datos presupuestarios de Sector Público Local, el Ministerio de Hacienda y Administraciones Públicas del Gobierno de España implementó dos taxonomías XBRL al respecto, **PENLOC**, que contiene los datos sobre los presupuestos de entidades locales, y **LENLOC**, que contiene los correspondientes a las liquidaciones de los mismos. Esta información esta accesible en la siguiente URL:

<http://serviciostelematicos.sgcal.minhap.gob.es/DescargaPresLiqXBRL/>

Para transformar estos datos en los de entrada para la aplicación web se ha utilizado una serie de scripts implementados en **Python**, lenguaje de programación interpretado, que soporta el paradigma de orientado a objetos, programación imperativa y, en menor medida, programación funcional. Los scripts funcionan sobre la versión de **Python 2.7.5**, por lo que es necesario tener instalada dicha versión en la máquina destinada al procesado.

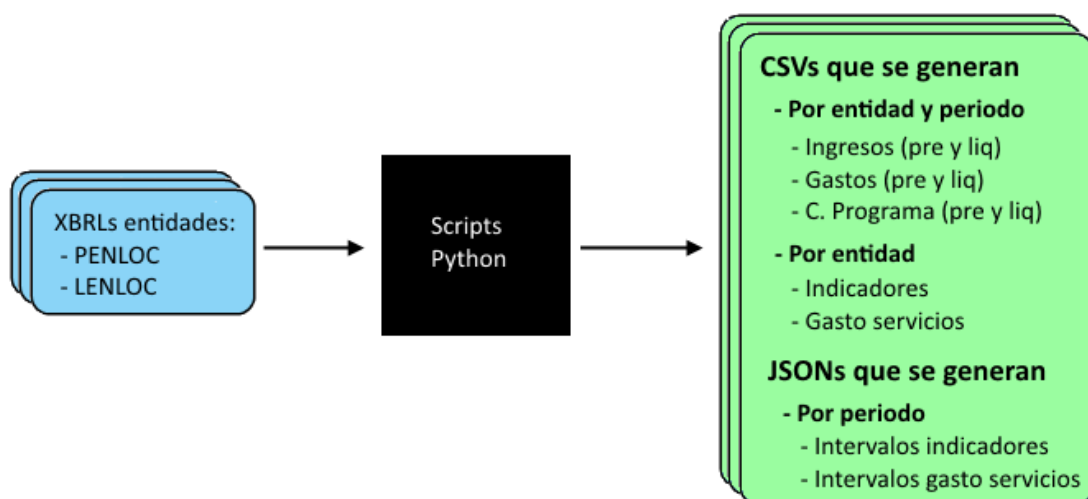
Datos generados

Los ficheros objetivo que alimentan la página son los siguientes:

- Por entidad y periodo
 - **CSVs de Gastos**: contienen información de las cuentas de gastos de la entidad local, tanto de liquidación como de presupuesto para un determinado periodo.
 - **CSVs de Ingresos**: contienen información de las cuentas de ingresos de la entidad local, tanto de liquidación como de presupuesto para un determinado periodo.
 - **CSVs de la clasificación por programa**: contienen información de las cuentas de la clasificación por programa de la entidad local, tanto de liquidación como de presupuesto para un determinado periodo.
- Por entidad
 - **CSVs de indicadores**: contienen información del valor de los diez principales indicadores de la entidad local para diversos periodos incluidos en el mismo fichero.

- **CSVs de gasto de servicios:** contienen información de los gastos de nueve servicios de la entidad local para diversos periodos incluidos en el mismo fichero.
- Por periodo y rango
 - **JSONs de intervalos en indicadores:** contiene información de los valores mínimo, máximo, primer y segundo cuartil, percentiles 1 y 95 y la mediana de los 10 indicadores financieros de las entidades locales para un determinado periodo y rango (municipios o diputaciones).
 - **JSONs de intervalos en gasto de servicios:** contiene información de los valores mínimo, máximo, primer y segundo cuartil, percentiles 1 y 95 y la mediana de los gastos de los 9 servicios de las entidades locales para un determinado periodo y rango (municipios o diputaciones).

En el siguiente esquema se puede observar las salidas que se quieren generar:



Datos extra

Además de los ficheros generados a partir de los XBRL, la página necesita los siguientes ficheros:

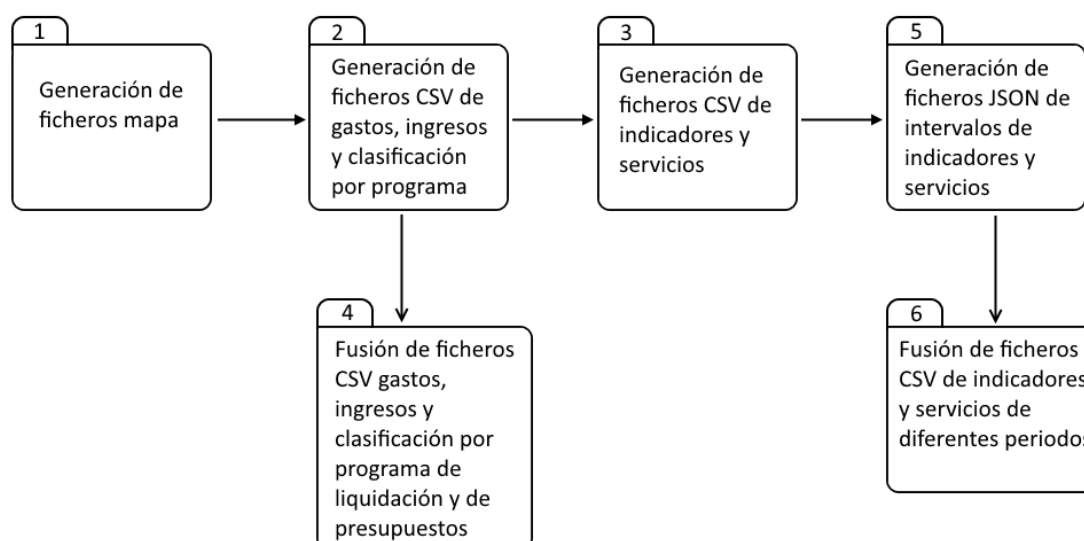
- Tablas *Google Fusion Tables* para alimentar *Google Maps*.
- Tablas *Google BigQuery* que poder consultar.
- Fichero JSON con pares código de municipio y nombre de municipio
- Fichero JSON con pares código de entidad (municipio o diputación) y población, diferenciadas por periodo.

Etapas de procesado

La información pasa por una serie de etapas hasta llegar a los datos utilizados en la aplicación web, son las siguientes:

1. Generación de ficheros mapa
2. Generación de ficheros CSV de gastos, ingresos y clasificación por programa
3. Generación de ficheros CSV de indicadores y servicios
4. Fusión de ficheros CSV gastos, ingresos y clasificación por programa de liquidación y de presupuestos.
5. Generación de ficheros JSON de intervalos de indicadores y servicios
6. Fusión de ficheros CSV de indicadores y servicios de diferentes periodos

En el siguiente diagrama se puede observar las diferentes etapas:



Generación de ficheros mapa

El objetivo de esta etapa es generar una serie de ficheros CSV mapa que permitan buscar información útil al recorrer los ficheros XBRL en la siguiente etapa. Los scripts utilizados para esta tarea son:

- **lenlocGastIngrMap.py:** genera mapa de gastos y mapa de ingresos de liquidaciones, en distintas ejecuciones.
- **penlocGastIngrMap.py:** genera mapa de gastos y mapa de ingresos de presupuestos, en distintas ejecuciones.
- **xbriProMap.py:** genera mapa de clasificación por programa de liquidaciones y mapa de clasificación por programa de presupuestos, en distintas ejecuciones.

Todos ellos necesitan como parámetros de entrada las definiciones XML de las taxonomías para las distintas cuentas (ingresos, gastos,...). Dado que los XBRL disponibles del ministerio son de las versiones penloc-2010 y lenloc-2010, se han creado los ficheros de mapa con las definiciones de esas versiones, estos se encuentran en el subdirectorío **mapas_csv** del directorío **Python_Scripts** a modo de ejemplo. Si se diera el caso de que se tuvieran que utilizar XBRL de una versión posterior, habría que modificar los anteriores scripts y utilizar las nuevas definiciones de las taxonomías. Cualquier adaptación por versión es independiente del resto de etapas. Las definiciones de LENLOC y PENLOC están disponibles en la siguiente URL:

<http://www.minhap.gob.es/es-ES/Areas%20Tematicas/Administracion%20Electronica/OVEELL/Paginas/LENLOC.aspx>

Generación de ficheros CSV de gastos, ingresos y clasificación por programa

El objetivo de esta etapa es generar una serie de ficheros CSV, de tal manera, que por entidad y periodo se creen dos ficheros con información de ingresos (uno de liquidación y otro de presupuestos), otros dos para gastos y otros dos para clasificación por programa, con lo que queda un total de 6 ficheros por entidad y periodo. Los scripts utilizados para esta tarea son:

- **lenloc_script_loop.py**: genera CSVs de gastos, ingresos y clasificación por programa de liquidaciones, en distintas ejecuciones.
- **penloc_script_loop.py**: genera CSVs de gastos, ingresos y clasificación por programa de presupuestos, en distintas ejecuciones.

Ambos exigen como entrada un fichero mapa de la información a calcular y un fichero de texto con un listado de ficheros XBRL a tratar, se generará un fichero CSV por cada XBRL de la lista. Se podría crear un fichero por lotes compuesto de líneas de ejecución de los scripts con diferentes listas de XBRL y mapas, para crear los CSVs de forma ordenada y jerárquica.

Generación de ficheros CSV de indicadores y servicios

El objetivo de esta etapa es generar una serie de ficheros CSV, de tal manera, que por entidad y periodo, se creen un fichero que contenga información de indicadores financieros y otro de gasto en servicios de la entidad pública. La información sobre indicadores se generará a partir de los ficheros de ingresos y gastos de liquidación de la administración local para un periodo dado y los datos sobre gasto de servicios a partir de los de clasificación por programa de la liquidación, todos creados en la anterior etapa. Los scripts utilizados para esta tarea son:

- **buildIndicators.py:** genera CSVs con los valores de diez indicadores financieros, tiene un comportamiento diferente cuando la entidad es un municipio y otro cuando es una diputación.
- **buildServiceCosts.py:** genera CSVs con el valor de los gastos de nueve servicios, tiene un comportamiento diferente cuando la entidad es un municipio y otro cuando es una diputación.

Ambos exigen como entrada un fichero JSON con el valor de las poblaciones de los municipios para el periodo de cálculo, estos ficheros de poblaciones se crearon a partir de la información publicada por el Instituto Nacional de Estadística. En el subdirectorio **datos-poblacion** del directorio **Python_Scripts** hay ejemplos de los JSON de población a utilizar. Los scripts a partir de la población de los municipios calculan la de las diputaciones sumando el conjunto correspondiente.

Los 10 indicadores financieros calculados son:

- Déficit/Superávit per cápita
- Ahorro neto per cápita
- Ahorro bruto per cápita
- Elasticidad presupuestaria
- Gasto público por habitante
- Gasto de inversión per cápita
- Gasto de inversión directa per cápita
- Presión fiscal
- Autonomía fiscal
- Incremento de deuda per cápita

Los 9 servicios de los que se calcula su gasto son:

- Vivienda y urbanismo
- Saneamiento, abastecimiento y distribución de aguas.
- Recogida, eliminación y tratamiento de residuos.
- Alumbrado público
- Medio ambiente
- Servicios Sociales y promoción social
- Sanidad
- Educación
- Cultura

Modificando estos scripts se pueden añadir nuevos indicadores y servicios a los ficheros CSV que se generen. Se debe usar un fichero por lotes compuesto de líneas de ejecución de los scripts, para crear los CSVs de forma ordenada, jerárquica y automatizada.

Fusión de ficheros CSV gastos, ingresos y clasificación por programa de liquidación y de presupuestos.

El objetivo de esta etapa es fusionar ficheros CSV generados en la etapa “Generación de ficheros CSV de gastos, ingresos y clasificación por programa”, quedando finalmente por entidad y periodo un fichero de ingresos, con información de liquidación y/o presupuesto, y los mismo para gastos y clasificación por programa, resultando al final 3 ficheros por entidad y periodo. Los scripts utilizados para esta tarea son:

- **mergeLenlocPenlocCsvs.py:** dado un fichero de liquidación (gastos o ingresos) y un fichero de presupuesto (gastos o ingresos) de una misma entidad y periodo se genera un CSVs con información de liquidación y presupuesto.
- **mergeLenlocPenlocProgCsvs.py:** dado un fichero de liquidación (clasificación por programa) y un fichero de presupuesto (clasificación por programa) de una misma entidad y periodo se genera un CSVs con información de liquidación y presupuesto.

En estos script tiene varias opciones de parámetros: se pueden introducir ficheros de liquidación y presupuesto y por tanto quedaran fusionados; se puede añadir un fichero de liquidación y se generará un fichero con el campo de presupuesto vacío; se puede añadir un fichero de presupuesto y se generará un fichero con el campo de liquidación vacío.

Se debe usar un fichero por lotes compuesto de líneas de ejecución de los scripts para crear los CSVs de forma ordenada, jerárquica y automatizada prestando atención a que habrá líneas con uno o dos parámetros, ya que puede haber entidades con información de liquidación y presupuesto o que solo disponga de un tipos de información, habrá que hacer un estudio previo.

Generación de ficheros JSON de intervalos de indicadores y servicios

El objetivo de esta etapa es generar una serie de ficheros JSON, que para cada periodo, se cree un fichero con los intervalos de los indicadores financieros y otro con los intervalos de gasto en servicios. Ambos contendrán el valor mínimo, máximo, primer cuartil y tercer cuartil, percentiles 1 y 95 y la mediana. Los scripts utilizados para esta tarea son:

- **buildPeriodIndicatorsJSON.py:** genera un fichero JSON con los valores de intervalos para indicadores en un periodo, antes descritos.

- **buildPeriodServiceCostsJSON.py:** genera un fichero JSON con los valores de intervalos para gasto de servicios en un periodo, antes descritos.

Ambos tendrán como parámetro un fichero de texto con una lista de ficheros CSV de indicadores o servicios según corresponda, siendo el total generado en el mismo periodo. Los ficheros incluidos en las listas se crearon en la etapa “Generación de ficheros CSV de indicadores y servicios”.

Fusión de ficheros CSV de indicadores y servicios de diferentes periodos

El objetivo de esta etapa es fusionar el contenido de los ficheros de indicadores y servicios generados en la etapa “Generación de ficheros CSV de indicadores y servicios”, generando un único fichero CSV por entidad, el cual incluirá columnas con los valores de periodos de los que se tenga información. Los scripts utilizados para esta tarea son:

- **appendIndicators.py:** dado un fichero de indicadores para una entidad en un periodo determinado, se genera un fichero con información de indicadores para ese año, si no existía ya un fichero para esa entidad con información de anteriores periodos, si ya existía simplemente se añade una nueva columna.
- **appendServiceCosts.py:** dado un fichero de gasto de servicios para una entidad en un periodo determinado, se genera un fichero con información de gasto de servicios para ese año, si no existía ya un fichero para esa entidad con información de anteriores periodos, si ya existía simplemente se añade una nueva columna.

Ambos scripts deben ejecutarse dentro de un mismo directorio ya que la primera vez que se ejecutan crean el fichero para la entidad, pero como vaya lanzándose con información de nuevos periodos se ira sobre-escribiendo el fichero antes mencionado. Se debe usar un fichero por lotes compuesto de líneas de ejecución de los scripts, para crear los CSVs de cada entidad a partir de la información de los distintos periodos que se tenga para la misma, de forma ordenada, jerárquica y automatizada.