

# Formas de utilizar la aplicación VIBIDA

Las posibilidades de utilización de la aplicación desarrollada en el proyecto de investigación VIBIDA (*Visualización multimedia de big data*) son dos:

- Enlazar con la página web del proyecto [VIBIDA](#).
- Reutilizar la aplicación a partir del contenido disponible en el [repositorio de github](#).

## 1. Enlazar con la página web

En esta opción basta con crear un enlace a la web [vibida.unizar.es](http://vibida.unizar.es), pasando como parámetro el código de la entidad local que se desee (ej. <http://vibida.unizar.es/?50000DD00>), de esta forma en el selector de la cabecera del sitio aparecerá la entidad pasada como parámetro.

## 2. Reutilizar la aplicación

Esta posibilidad permite reutilizar la aplicación. En este caso la entidad tiene una instancia de la aplicación que puede utilizar y modificar libremente. En particular, puede:

- Personalizar la página poniendo su escudo, colores institucionales, etc. Únicamente deben conservarse (aunque puede reducirse su tamaño) los logotipos del Ministerio y del Gobierno de Aragón, ya que son quienes han subvencionado el desarrollo.
- Incorporar los datos de la entidad mediante los ficheros LENLOC y PENLOC. Los datos del conjunto de las entidades de España podrán descargarse ya en formato adecuado, del sitio VIBIDA.
- Seleccionar que secciones y que indicadores quiere que se visualicen.

El primer paso a tener en cuenta es la instalación de la aplicación web, para lo cual solo es necesario disponer de un **servidor Apache** con el módulo de **PHP** correspondiente habilitado. Descargando la carpeta **Web** del proyecto y colocándola en el directorio **htdocs** del servidor, se tendrá un duplicado de la aplicación VIBIDA, salvo por la carpeta **data**, que solo posee ficheros de configuración e información general.

Para poder acceder a los datos de administraciones locales desde 2010 hasta el último año con información que ha servido el Ministerio de Hacienda y Administraciones Públicas del Gobierno de España, por el momento **2012**, se ha habilitado una URL donde poder descargar dicha información comprimida en un fichero *zip*. El enlace es el siguiente:

[http://vibida.unizar.es/datos\\_acum\\_periodo/](http://vibida.unizar.es/datos_acum_periodo/).

Dentro del fichero *zip* hay una carpeta **data**, al volcar su contenido en la homónima del servidor la aplicación será funcional. Se seguirá alimentando con datos de entidades públicas la página web, con la limitación del tiempo que tarde el ministerio en publicar nuevas cuentas, en consecuencia, la URL con ficheros *zip* también actualizará su contenido (utilizar siempre el más reciente).

## 2.1. Personalizar la página

Personalizar el aspecto de la aplicación web pasa por modificar ciertos ficheros clave. Si se quiere cambiar la imagen de logo de la cabecera habrá que cambiar la URL de la imagen incluida en el *header* de **index.html**, para modificar sus dimensiones habrá que modificar el contenido de la hoja de estilo **vibida\_style.css** dentro de la subcarpeta **CSS**. Para cambiar el color de las barras horizontales que encabezan cada sección, dentro de **vibida\_style.css** habrá que cambiar el contenido de los selectores CSS: `section div.head_section` (sección principal), `div.collapse_bar` (subsección) y `section div.new_section` (nueva sección, “RELACIÓN ENTRE DOS INDICADORES”). Seguir haciendo cambios en el estilo implicará seguir modificando **vibida\_style.css** a discreción.

Se ha añadido el subdirectorio **cabecera\_customizada** dentro del directorio **Web** del repositorio, que contiene ficheros modificados respecto la versión inicial para que en la cabecera solo aparezca el logo, el título “Datos económico-financieros” y el selector de periodo, de esta manera se consigue una página con la cabecera personalizada para una sola entidad. Dentro de esta carpeta en la subcarpeta **js** se encuentra el fichero `vibidaScript.js` donde habrá que hacer una serie de modificaciones para que se carguen los datos de la entidad seleccionada:

- **Si la entidad es un municipio**, basta con ir a la función **initMun()** y modificar el siguiente código con el identificador de provincia y municipio correspondiente:

```
// - provincia a la que pertenece el municipio a seleccionar
$("#select#provincias option[value='22']").prop("selected", true);
// - Se carga la lista de municipios de la provincia seleccionada
loadMun("22");
// - Se selecciona el municipio
$("#select#municipios option[value='22125AA00']").prop("selected", true);
```

- **Si la entidad es una diputación** habrá que modificar la función **initDip()** con el identificador de diputación y cambiar la función `initHeaderMenu`:

```
//Sección a modificar si la entidad de la cabecera es una diputación:
// - Se selecciona la diputación
$("#select#diputaciones option[value='50000DD00']").prop("selected", true);
```

```

function initHeaderMenu(){
    //$("select#division option[value='MUN']").prop("selected", true);
    $("select#division option[value='DIP']").prop("selected", true);
    //hideDipMenu();
    showDipMenu();
    //showMunMenu();
    hideMunMenu();
    $("select#periodo option[value='2012']").prop("selected", true);
    initDip();
    initMun();
}

```

## 2.2. Pasos a seguir para incorporar nuevos datos de la entidad a la aplicación web a partir de ficheros XBRL

0. Tener instalado Python 2.7.5 en la máquina donde se vayan a tratar los ficheros XBRL.
1. Se supone que la nomenclatura de los ficheros XBRL de liquidación y presupuesto es la siguiente:
  - Liquidacion\_periodo\_Ente\_codigoEntidad.xbrl
  - Presupuesto\_periodo\_Ente\_codigoEntidad.xbrl

Para el caso de la entidad con código 00000AA00 y periodo 2013 sería:

- Liquidacion\_2013\_Ente\_00000AA00.xbrl
  - Presupuesto\_2013\_Ente\_00000AA00.xbrl
2. Crear un mapa CSV nuevo si los ficheros XBRL son de una versión LENLOC o PENLOC diferente a la de los mapas ya creados. Estos ficheros mapa hacen posible extraer la información deseada del fichero XBRL.
  3. Crear ficheros CSV por entidad y periodo con información de gastos, ingresos y clasificación por programa, dependiendo de si el fichero XBRL es de liquidación o de presupuesto se generarán datos distintos. Se utilizarán los scripts **lenloc\_csv\_script\_loop.py** y **penloc\_csv\_script\_loop.py** respectivamente. Ejemplo para una entidad:

```

python lenloc_csv_script_loop.py listaLenloc.txt
mapas_csv\lenloc3-econ-ingr-cuentas-label.csv name-domain\name-
domain-lenloc3.json 1:N

```

```
python penloc_csv_script_loop.py listaPenloc.txt
mapas_csv\penloc2-econ-ingr-cuentas-label.csv name-domain\name-
domain-penloc2.json 1:N
```

Se necesitan un fichero de texto de lista de ficheros XBRL a procesar, un fichero mapa y otro de dominio de nombres e indicar la relación en la definición de contextos (1:1 o 1:N).

4. Crear ficheros CSV por entidad y periodo con información de indicadores financieros y gasto de servicios. Para esta tarea se utilizarán los scripts **buildIndicators.py** y **buildServiceCosts.py** respectivamente. Ejemplo para una entidad (código 00000AA00 y periodo 2013):

```
python buildIndicators.py
Liquidacion_2013_Ente_00000AA00_lenloc3-econ-ingr-cuentas-
label_DATA_2.csv Liquidacion_2013_Ente_00000AA00_lenloc3-econ-
gast-cuentas-label_DATA_2.csv datos-poblacion\pobmun13.json
```

```
python buildServiceCosts.py
Liquidacion_2013_Ente_00000AA00_lenloc3-prog-cuentas-
label_DATA_2.csv datos-poblacion\pobmun13.json
```

Se necesitan los ficheros creados en el paso 3 respetivos a la información de liquidación de la entidad y ficheros JSON con información sobre la población por entidad, en el proyecto esta información se creó a partir de información pública por el INE.

5. Fusionar información de liquidación y presupuestos generada en el paso 3, creando 3 ficheros (ingresos, gastos y cl. program.) por entidad. Para esta tarea se utilizarán los scripts **mergeLenlocPenlocCsvs.py** y **mergeLenlocPenlocProgCsvs.py** respectivamente. Ejemplo para una entidad (código 00000AA00 y periodo 2013):

```
python mergeLenlocPenlocCsvs.py
Liquidacion_2013_Ente_00000AA00_lenloc3-econ-ingr-cuentas-
label_DATA_2.csv Presupuesto_2013_Ente_00000AA00_penloc2-econ-
ingr-cuentas-label_DATA_2.csv
```

```
python mergeLenlocPenlocCsvs.py
Liquidacion_2013_Ente_00000AA00_lenloc3-econ-gast-cuentas-
label_DATA_2.csv Presupuesto_2013_Ente_00000AA00_penloc2-econ-
gast-cuentas-label_DATA_2.csv
```

```
python mergeLenlocPenlocProgCsvs.py
Liquidacion_2013_Ente_00000AA00_lenloc3-prog-cuentas-
label_DATA_2.csv Presupuesto_2013_Ente_00000AA00_penloc2-prog-
cuentas-label_DATA_2.csv
```

6. Fusionar información de ingresos y servicios generada en el paso 4, creando un único fichero por entidad donde podrá haber información de varios periodos. Para esta tarea se utilizarán los scripts python

**appendIndicators.py** y **appendServiceCosts.py** respectivamente. Ejemplo para una entidad (código 00000AA00 y periodo 2013):

```
python appendIndicators.py 2013-00000AA00-indi.csv
```

```
python appendServiceCosts.py 2013-00000AA00-serv.csv
```

Estos scripts deben ejecutarse de forma secuencial por cada periodo en un mismo directorio para que se vaya sobreescribiendo la información en el mismo fichero, en el caso del primer periodo introducido, se creará un fichero desde cero.

7. Los ficheros generados en los pasos 5 y 6 deben copiarse en la carpeta **data** de la aplicación web. En el fichero **index.html**, en la definición del **selector de periodo** de la cabecera, debe añadirse una opción para el nuevo periodo. El ejemplo para 2013 sería:

```
...
<span>Periodo</span>
  <div class="menu with-head">
    <select name="" id="periodo">
      <option value="2010">2010</option>
      <option value="2011">2011</option>
      <option value="2012" selected="selected">2012</option>
      <option value="2013">2013</option>
    </select>
  </div>
...
```

Una vez estos pasos han sido realizados, navegar por las secciones de los periodos anteriores a la nueva información añadida no tendrá diferencias, cosa que no pasará al cambiar el selector al nuevo periodo introducido. En el fichero **vibidaScript.js** dentro del directorio **js** de la aplicación web hay definida un variable llamada **period\_limit** que marca el último periodo con información de todos las entidades, si el selector de periodo cambia a un valor superior a esta variable la web tendrá un comportamiento diferente, las secciones “Comparar dos administraciones”, “Ver indicadores o seleccionar una Administración en Google Maps” y “RELACIÓN ENTRE DOS INDICADORES” estarán colapsadas y deshabilitadas, en la sección “LOS 10 PRINCIPALES INDICADORES FINANCIEROS”, los intervalos (mínimo, máximo, mediana,...) utilizados para los gráficos Gauge son los del periodo **period\_limit** y en la sección “GASTOS DE LOS PRINCIPALES SERVICIOS”, sus selectores estarán deshabilitados y el valor promedio será el del periodo **period\_limit**.

Dentro del directorio **Python\_Scripts**, existe el fichero **generarDatos.bat**, que ejecuta, a modo de ejemplo, el procesado del fichero XBRL de una entidad. Este script llama a su vez a **tareasGenerarDatos.bat** con una serie de

parámetros, el primero de ellos es el código identificador de la entidad, el segundo el periodo de la información a procesar, el tercero la versión de LENLOC utilizada, el cuarto la relación entre definición contexto y número de importes del XBRL de liquidación, el quinto la versión de PENLOC utilizada y el sexto la relación entre definición contexto y número de importes del XBRL de presupuesto. Un ejemplo para entidad 00000AA00, con información del periodo 2013, versión 3 de LENLOC, versión 2 de PENLOC y relación 1:N en los XBRL de liquidación y presupuesto, sería:

```
tareasGenerarDatos.bat 00000AA00 2013 3 1:N 2 1:N
```

Es importante que los ficheros XBRL de liquidación y presupuesto estén en el mismo directorio que los scripts y si existiera información de indicadores y servicios previa (Ej: 00000AA00-indi.csv y 00000AA00-serv.csv) también debe colocarse en este directorio para que se les añada un nueva columna de periodo.

Si posteriormente se quisiera añadir nueva información de las entidades de toda España que actualice el ministerio, se podrá acudir al directorio que contendrá los datos ya tratados, como ya se comentó en la instalación, [http://vibida.unizar.es/datos\\_acum\\_periodo/](http://vibida.unizar.es/datos_acum_periodo/), descargar la información del nuevo periodo, sobrescribir la carpeta **data** de la aplicación web y cambiar la variable **period\_limit** del script **vibidaScript.js**. Se debe tener cuidado de no eliminar ficheros de entidades con datos más actualizados que los que se quiere cargar, como puede ser el caso de que se suba información de una entidad hasta el año 2014 teniendo datos del resto de municipios de 2012 y posteriormente se publiquen datos de 2013 y se quiera actualizar la web, habrá que evitar que se destruya los ficheros relativos a esa entidad.

### 2.3. Seleccionar que secciones y que indicadores son visibles

En el directorio **Web** del repositorio en el subdirectorio **Data** se encuentran los ficheros **display\_config.json** y **display\_config\_default.json**. Modificar **display\_config.json** hará que se oculten o muestren secciones, indicadores y servicios y el fichero **display\_config\_default.json** sirve de referencia para el usuario de tal manera que su contenido es la versión de **display\_config.json** con todas las secciones y componentes visibles.

La estructura del fichero JSON es la siguiente:

```
{
  "section0": {"name":"Mapa","options":[1,2,3,4,5,6,7,8,9,10]},
  "section1":
  {"name":"Indicadores","slides":[["indicador1","indicador2","indicador3"],["indicador4","indicador5","indicador6"],["indicador7","indicador8","indicador9","indicador10"]]},
  "section2": {"name":"Ingresos y gastos","slides":[["treemap"],["hbars"]]},
```

```

"section3": {"name":"Comparar
indicadores","slides":[[{"indicador1","indicador2","indicador3"},[{"indicador4",
"indicador5","indicador6"},[{"indicador7","indicador8","indicador9","indicador1
0"}]]},
"section4":
{"name":"Servicios","slides":[[{"servicio1","servicio2","servicio3"},[{"servicio
4","servicio5","servicio6"},[{"servicio7","servicio8","servicio9"}]]},
"section5": {"name":"Relacion","options":[1,2,3,4,5,6,7,8,9,10,11,12,13]}
}

```

El objeto tiene tantas propiedades como secciones, siendo “section0” la sección “Ver indicadores o seleccionar una Administración en Google Maps”, “section1” la sección “LOS 10 PRINCIPALES INDICADORES FINANCIEROS”, “section2” la sección “INGRESOS Y GASTOS”, “section3” la sección “Comparar dos administraciones”, “section4” la sección “GASTOS DE LOS PRINCIPALES SERVICIOS” y “section5” la sección “RELACIÓN ENTRE DOS INDICADORES”.

Todas las secciones a su vez tienen una propiedad “name” con su nombre resumido. Las secciones 0 y 5 poseen la propiedad “options” un *array* de enteros, cada número representa una de las opciones del selector de las secciones. Las secciones 1, 2, 3 y 4 tienen la propiedad “slides”, *array* de *arrays* con cadenas de texto que representan los *dashboards* dentro de las secciones.

Para hacer desaparecer una sección completa basta igualar la propiedad “sectionX” a null (ej: “section1”: null), si se quiere ocultar una de las opciones de los selectores de las secciones 0 y 5 basta con borrar el número (ej: “section5”: {"name":"Relacion","options":[1,2,3]}, esto ocultaría todas las opciones menos las 3 primeras), por último si se quieren ocultar *dashboard* de indicadores y gasto de servicios de las secciones 1, 2, 3 y 4, habría que borrar las cadenas de texto de los indicadores/servicios del campo “slides” (ej: “section1”: {"name": "Indicadores", "slides":[[{"indicador2","indicador3"},[{"indicador4","indicador5"},[{"indicador7","indicador9","indicador10"}]]}, para ocultar los indicadores 1, 6 y 8).