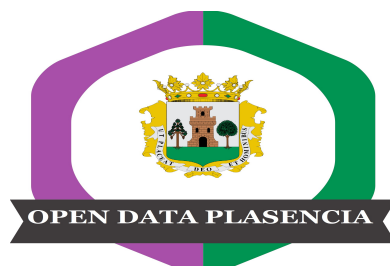


1. DESCRIPCIÓN

OpenData Plasencia es un portal web donde obtener información sobre la ciudad de Plasencia; su gobierno, sus barrios y asociaciones de vecinos e información sobre los organismos sociales, educativos o sanitario de la ciudad.



Es un proyecto realizado con el fin de acercar a la ciudadanía de Plasencia información sobre la misma.

Otro de sus objetivos es dar cobertura a desarrolladores de software que deseen implementar datos de Plasencia en sus proyectos. El portal da la posibilidad al cliente de generar datos de la ciudad en formato Xml, JSon, Html o Csv.

El portal esta realizado con tecnologías actuales e innovadoras como AngularJS con la que se realiza la arquitectura de la web, Css compilado a partir de Sass, con la que se le da estilo a la web y Bootstrap el cual nos permite dar formato tanto a dispositivos móviles como ordenadores.

Dicho portal esta sujeto a una base de datos realizada con MySql la cual se encarga de definir todos los datos que se muestran en la web.

 Inicio Ayuntamiento Organismos Sociales Barrios Sobre...									
Ayuntamiento de Plasencia Emplazado en la plaza mayor de Plasencia, el ayuntamiento es un centro neurálgico de la política placentina y por ende responsable del futuro de los ciudadanos. En esta sección se muestran los datos del gobierno placentino.									
#	NOMBRE	MINISTERIO	DESDE	HASTA	EMAIL	TELÉFONO	FAX	WEB	TWITTER
1	BERNANDO PIZARRO GARCIA	ALCALDIA	2015-05-24		ALCALDIA@AYTOPLASENCIA.ES	927 42 85 00	927 42 85 00		
2	LUIS DÍAZ SÁNCHEZ	JUVENTUD	2015-05-24		CONCEJALIA JUVENTUD@AYTOPLASENCIA.ES	927 42 85 00	927 42 13 04		
3	BERNANDO PIZARRO	EDUCACIÓN Y ENFERMERÍA	2015-05-24		ALCALDIA@AYTOPLASENCIA.ES	27 42 85 00	927 416 473		

Es un portal moderno que utiliza las tecnologías más actuales del momento para ofrecer a sus usuarios una experiencia de uso de última generación.

Esta diseñada para poder verla a través de un navegador web o desde navegadores de dispositivos móviles gracias a la tecnología de Bootstrap con la que se conseguirá un diseño actual para todos los tipos de pantalla.

Todo lo que necesitas para poder obtener los datos del gobierno de Plasencia en formato csv, por ejemplo, es ir a la página <https://museumis.github.io/OpenDataPlasencia/#/> y navegar por ella sin compromiso.

Toda una base de datos rellena con datos obtenidos en la red, webs oficiales de las entidades de las que se compone esta base de datos, dan a esta web un potencial de uso adecuado a los años que corren,

La web es completamente funcional y ya puedes empezar a usarla. Actualmente sólo se puede visionar partes de la web pero es cuestión de tiempo que toda la web esté completamente rellena.

Datos abiertos que podrás disfrutar y aprovechar en tus proyectos.

Para tener una ciudadanía informada sobre algunos aspectos importantes sobre la ciudad en la que viven es necesario agrupar los datos de ella de una manera ordenada y con una búsqueda de información lo más amena y sencilla posible.

Para esto nace OpenData Plasencia para tener a sus ciudadanos informados de todo aquello que les concierne.

1	FERNANDO PIZARRO GARCIA	ALCALDIA	2015-05-24	ALCALDIA@AYTORPLASENCIA.ES	927 42 85 00	927 42 85 00			
2	LUIS DÍAZ SÁNCHEZ	JUVENTUD	2015-05-24	CONCEJALIAJUVENTUD@AYTORPLASENCIA.ES	927 42 85 00	927 42 13 04			
3	FERNANDO PIZARRO	EDUCACIÓN Y PATRIMONIO	2015-05-24	ALCALDIA@AYTORPLASENCIA.ES	27 42 85 00	927 418 473			
4	DAVID DÓNGA ESTÉVEZ	DEPORTES, INNOVACIÓN, COMUNICACIÓN, DISTritos Y FOMENTO	2015-05-24		927 42 85 00	FOMENTO.DEPORTES@OUTLOOK.ES	WWW.DEORTEPLASENCIA.COM	@OITPLASENCIA	HTTP://WWW.FACEBOOK.COM/CONCEJALIA.DEORTEPLASENCIA
5	Mª LUISA BERMEO	CULTURA	2015-05-24	PROCOCCULTURAL@AYTORPLASENCIA.ES	927 42 85 00	927 42 13 04			
6	SONIA GRANDE SÁNCHEZ	RECURSOS HUMANOS, EMPLEO, EMPRESA, INTERIOR Y ADMINISTRACIÓN GENERAL	2015-05-24	ADMONGENERAL@AYTORPLASENCIA.ES	927 42 85 00	927 42 13 04			
7	JOSÉ ANTONIO HERNÁNDEZ	HACIENDA, URBANISMO Y FONDOS EUROPEOS	2015-05-24	HACIENDAYURBANISMO@AYTORPLASENCIA.ES	927 42 85 00	927 42 13 04			
8	FLOR CONEJERO FERNÁNDEZ	SERVICIOS SOCIALES Y MAYORES	2015-05-24	CONCEJALIASOCIALES@PLASENCIA.ES	927 42 85 00	927 42 13 04			
9	LUIS DÍAZ GONZÁLEZ	RELACIONES INSTITUCIONALES	2015-05-24		927 42 85 00	927 42 13 04			
10	Mª TERESA DÍAZ	FAMILIA, IGUALDAD, ACCESIBILIDAD Y DEPENDENCIA	2015-05-24	CONCEJALIAFAMILIA@AYTORPLASENCIA.ES	927 42 85 00	927 42 13 04			
11	LUIS MIGUEL PÉREZ	SERVICIOS MUNICIPALES, SANIDAD Y MEDIO AMBIENTE	2015-05-24	SANIDADMAYSSMM@AYTORPLASENCIA.ES	927 42 85 00	927 42 13 04			

Todo desarrollador de software siempre es gustoso de recibir información adicional para sus proyectos quien sabe que desarrollador puede necesitar datos de Plasencia para implementarlos en su proyecto.

Para esto también esta OpenData Plasencia todo desarrollador que desee implementar en sus apps información de Plasencia puede obtenerlo a partir de este portal.

Ayuntamiento de plasencia

CSV

Ver en Web

Nombre,Ministerio,Desde,Hasta,Gmail,Teléfono,Fax,Web,Twitter,Facebook

Fernando Pizarro Garcia,Alcaldia,2015-05-24,null,927 42 85 00,alcaldia@aytorplasencia.es,927 42 85 00,null,null,null
 Luis Díaz Sánchez,Juventud,2015-05-24,null,927 42 85 00,concejaliajuventud@aytorplasencia.es,927 42 13 04,null,null,null
 Fernando Pizarro,Educación y Patrimonio,2015-05-24,null,27 42 85 00,alcaldia@aytorplasencia.es,927 418 473,null,null,null
 David Dóniga Estévez,Deportes, Innovación, Comunicación, Distritos y Fomento,2015-05-24,null,927 42 85 00,fomento.deportes@outlook.es,www.deorteplasencia.com,@oitplasencia,http://www.facebook.com/concejalia.deorteplasencia
 Mª Luisa Bermeo,Cultura,2015-05-24,null,927 42 85 00,proccocultural@aytorplasencia.es,927 42 13 04,null,null,null
 Sonia Grande Sánchez,Recursos Humanos, Empleo, Empresa, Interior y Administración General,2015-05-24,null,927 42 85 00,admongeneral@aytorplasencia.es,927 42 13 04,null,null,null
 José Antonio Hernández,Hacienda, Urbanismo y Fondos Europeos,2015-05-24,null,927 42 85 00,haciendayurbanismo@aytorplasencia.es,927 42 13 04,null,null,null
 Flor Conejero Fernández,Servicios Sociales y Mayores,2015-05-24,null,927 42 85 00,concejaliasociales@plasencia.es,927 42 13 04,null,null,null
 Luis Díaz González,Relaciones Institucionales,2015-05-24,null,927 42 85 00,null,927 42 13 04,null,null,null
 Mª Teresa Díaz,Familia, Igualdad, Accesibilidad y Dependencia,2015-05-24,null,927 42 85 00,concejaliafamilia@aytorplasencia.es,927 42 13 04,null,null,null
 Luis Miguel Pérez,Servicios Municipales, Sanidad y Medio Ambiente,2015-05-24,null,927 42 85 00,sanidadmayssm@aytorplasencia.es,927 42 13 04,null,null,null
 Isidro Rodríguez,Turismo,2015-05-24,null,927 42 85 00 / 683313728 /Extensión 5854/5855/5856,concejaliaturismo@plasencia.es,927 42 13 04,null,null,null

3. JUSTIFICACIÓN

Es tiempo de datos, Cáceres y La Oliva de Plasencia ya tienen OpenData. ¿Porqué Plasencia no?

En pleno siglo XXI toda ciudad que desee estar modernizada no debe olvidarse de los datos abiertos, con los que tener bien informados a sus ciudadanos y desarrolladores. Es por ello que este proyecto se realiza.

Un herramienta ambiciosa e innovadora con utilidades beneficiarias para cualquiera que desee utilizarla.

La exploración de nuevas tecnologías de software explica la realización del proyecto con tecnologías de última generación.

La intención de realizar un proyecto personal en el que se desarrolle el front y el backen de la aplicación también justifica la ambiciosa realización de este proyecto.

El estudio de nuevas tecnologías en las prácticas del ciclo profesional que curso también ha hecho que la balanza se incline por estas tecnologías.

4. DESCRIPCIÓN TÉCNICA.

A) INTRODUCCIÓN

El proyecto gira entorno a una base de datos hospedada en 00WebHost su arquitectura esta basada en MySQL. Es la encargada de definir los campos que se muestran en las tablas de la web.

La información de la base de datos se representa como Json utilizando PHP, el cual nos permite mostrar todos los datos en urls definidas en el servidor anteriormente citado.

La arquitectura web hospedada en *GitHub*, AngularJS, se encarga de realizar consultas a dicha URL y procesar los datos de tal forma que se muestran en las tabla de cada apartado del portal web. O en dichos apartados mediante AngularJS y gracias al cliente se muestra dicha consulta en diferentes formatos.



B) HOSPEDAJE

Antes de abordar los detalles del proyecto tenemos que tener en cuenta que actualmente el proyecto se encuentra en la nube y el propósito de la misma es seguir estándolo.

Para ello se han utilizado dos hosting gratuitos que nos permiten tener la base de datos y la web siempre activa.

La base de datos esta hospeda en <https://es.000webhost.com/> y tratada mediante PhpMyAdmin nos permite gestionar los PHP, también alojados en este host, los cuales dan lugar a las url con los que se obtienen los datos en formato JSON

https://ismartinr01.000webhostapp.com/OpenData/Plasencia/Ayuntamiento/Obtener_gobierno.php

Por otra parte la web esta alojada en GitHub. La funcionalidad de poder hacer cambios instantáneos o la propia ventaja de tener la web en la nube sin inconvenientes hacen este host muy atractivo.

La posibilidad de que cualquiera pueda ver el código y participar en el proyecto hace que el espíritu de OpenData quede reflejado al menos desde esta perspectiva.

Toda la documentación del proyecto, tanto base de datos, memoria, php, etc, quedará guardado en dicho repositorio de Git.

<https://github.com/museumis/OpenDataPlasencia>

El motivo por el que la base de datos y la web están alojados en distinto servidores, esta comentado en los anteriores párrafos, la idea de dar la posibilidad de participar en la escritura del código de la web hace de GitHub un hosting muy atractivo. Por otro lado no permite la inserción de base de datos o el mapeo de clases mediante url, es por ello que se acude a 00webhost.

Otro inconveniente de 00WebHost es que el servidor esta inactivo una hora al día (5:00 / 6:00).

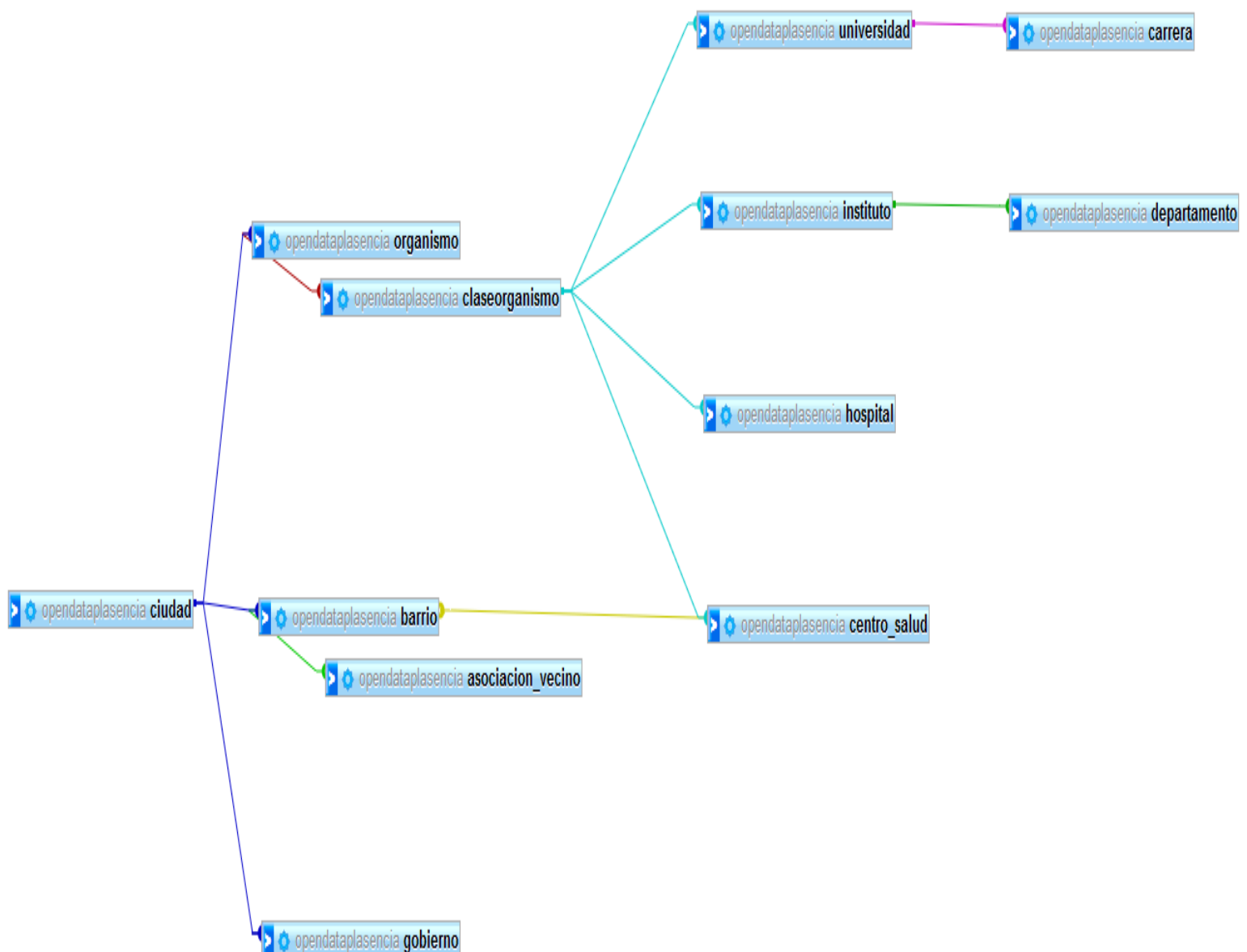
Apuntar que algunos sistemas de redes, como la red del instituto IES Valle del Jerte, bloquean el acceso a este hosting lo que imposibilita una visión adecuada de la web.

C) BASE DE DATOS

Base de datos que almacena información sobre ciudades.

De la ciudad se guarda su nombre, el país al que pertenece, el código postal, su página web oficial, su población, la fecha de su feria y fiestas y un enlace de su localización.

Cada ciudad es gobernada por un gobierno del que se almacenarán los gobernantes; su nombre, sus apellidos y el ministerio que lideran. El gobierno esta dirigido por un alcalde que será miembro del gobierno.



Realizada con MySql la base de datos toma relevancia en el proyecto. Siendo el centro de todo el algoritmo. Un cambio en la base de datos se verá reflejado en tiempo real en el portal web.

La base de datos puede encontrarse en fichero *.txt* lista para importar en los anexos de la documentación.

En la siguiente imagen se muestra el cuerpo de las tablas ciudad y gobierno.

```
CREATE DATABASE CIUDADES2

/*----- TABLAS -----*/

CREATE TABLE CIUDAD(
COD_CIUADAD INT PRIMARY KEY,
NOMBRE VARCHAR(20),
PAIS VARCHAR(20),
PAGINA_WEB VARCHAR(50),
GOOGLE_MAP VARCHAR(50),
POBLACION INT,
FIESTAS_INICIO DATE,
FIESTAS_FIN DATE,
CODIGO_POSTAL VARCHAR(10)
);

SELECT *
FROM CIUDAD

/*-----*/

CREATE TABLE GOBIERNO(
COD_GOBERNANTE INT UNIQUE,
COD_CIUADAD INT,
NOMBRE VARCHAR(50),
MINISTERIO VARCHAR(30),
FECHA_INICIO DATE,
FECHA_FIN DATE,
ALCALDE INT,

CONSTRAINT FK_GOB FOREIGN KEY (COD_CIUADAD) REFERENCES CIUDAD(COD_CIUADAD) ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT FK_ALC FOREIGN KEY (ALCALDE) REFERENCES GOBIERNO(COD_GOBERNANTE) ON DELETE RESTRICT ON UPDATE CASCADE,
CONSTRAINT PK_GOB PRIMARY KEY (COD_GOBERNANTE,COD_CIUADAD)
);
```

Los datos se han insertado con información recabada desde la página oficial del ayuntamiento de Plasencia o las diversas webs de entidades que están involucradas en las tablas de la base de datos, como puede ser información referente a colegios, universidades o organismos de seguridad como la policía, de las cuales se ha podido obtener información desde sus sitios webs oficiales.

En la siguiente imagen se muestran alguna de los *inserts* que se han llevado acabado en la base de datos.

/*-----*/

INSERT INTO GOBIERNO

```
VALUES(1,1,'Fernando Pizarro García','Alcaldia','2015-5-24',null,1),
(2,1,'Luis Díaz Sánchez','Juventud','2015-5-24',null,1),
(3,1,'Fernando Pizarro','Educación','2015-5-24',null,1),
(4,1,'David Dóniga Estévez','Innovación','2015-5-24',null,1),
(5,1,'Ma Luisa Bermejo','Cultura','2015-5-24',null,1)
;
```

SELECT *

FROM GOBIERNO

/*-----*/

INSERT INTO ORGANISMO

```
VALUES(1,1,'Parque Comarcal de bomberos de Plasencia','Polígono Industrial. C/ Juan de la Cierva s/n','080/927 410 080/927 427 063','Seguridad'),
(2,1,'Jefatura de la Policía Local','Avenida de las Acacias, s/n (La Mazuela)','092/927 42 13 15/927 42 65 64','Seguridad'),
(3,1,'C.C.E.I.P.S San Calixto','Av. Virgen del Puerto','927 41 13 47','Educación'),
(4,1,'C.E.I.P. Miralvalle','Calle Cayetano García Martín','927 01 78 00','Educación'),
(5,1,'Colegio Público El Pilar','Av. Extremadura','927 41 13 47','Educación'),
(6,1,'C.E.I.P. Alfonso VIII','Calle Cristo de las Batallas','927 01 77 96','Educación'),
(7,1,'CEIP Inés Suárez','Parque de la Coronación','927 01 77 84','Educación'),
(8,1,'IES Valle del Jerte','Calle Pedro y Francisco González','927 01 77 74','Educación'),
(9,1,'I.E.S. Gabriel y Galán','España, Av. Cañada Real','927 01 77 88','Educación'),
(10,1,'Instituto Pérez Comendador','Av. Virgen del Puerto','927 01 77 32','Educación'),
(11,1,'I.E.S. Sierra de Santa Bárbara','Calle Aldehuela del Jerte','927 42 46 26','Educación'),
(12,1,'Centro Universitario De Plasencia','Av. Virgen del Puerto','927 42 77 00','Educación'),
(13,1,'U.N.E.D. Plasencia','Plaza Sta. Ana','927 42 05 20','Educación'),
(14,1,'Hospital Virgen del puerto','Paraje Valcorchero','927 42 83 00','Sanidad'),
(15,1,'Luis de Toro','Calle Luis de Toro','CitaPrevia:927 423 436/Urgencias:927 423 380','Sanidad'),
(16,1,'San Miguel','Calle Antonio Vargas y Laguna','Urgencias: 927 407 522','Sanidad'),
(17,1,'La Data','Calle Cañada Real','CitaPrevia:927 428 421/Urgencias:927 428 420','Sanidad'),
(18,1,'Salud Mental','Avda.de la Salle','927 458 036','Sanidad'),
(19,1,'Conductas Adictivas','Avda.de la Salle','927 458 036','Sanidad')
```

La base de datos se ha escrito línea a línea y se ha importado y tratado desde *phpMyAdmin*.

D) PHP

Mediante PHP creamos una conexión entre la base de datos y la web.

El PHP accediendo a la base de datos genera una serie de urls las cuales muestran los datos de la base de datos en formato JSON y XML.

```
1 <?php
2 /**
3  * Obtiene todas los gobernantes de la base de datos
4  */
5
6 require 'Gobierno.php';
7
8 if ($_SERVER['REQUEST_METHOD'] == 'GET') {
9
10     // Manejar petición GET
11     $gobierno = Gobierno::getAll(); //Ejecuta el metodo getAll() de la clase Gobierno
12
13     if ($gobierno) {
14
15         $datos["estado"] = 1;
16         $datos["gobierno"] = $gobierno;
17
18         // print xmlrpc_encode($datos); //Estructura de resultado en XML
19         print json_encode($datos); //Estructura de resultado en JSON
20
21     } else {
22         print json_encode(array(
23             "estado" => 2,
24             "mensaje" => "Ha ocurrido un error"
25         ));
26     }
27 }
28
29
```

La clase encode da la posibilidad de hacer request mapping de una consulta y mostrarla en formato JSON.

Esta clase también permite devolver el mapeo en XML, como se observa en el la línea comentada número 18. Cabe señalar que este método de encode es experimental, por lo que no está implementado en la arquitectura PHP de forma definitiva.

La documentación de dicha función se puede encontrar en <http://php.net/manual/es/function.xmlrpc-encode.php>

La generación de la URL viene dada por PHP mediante la funcionalidad encode en la dirección web del hosting donde se encuentra el fichero que realiza la llamada a esta función. La dirección del directorio del fichero será la url donde se generen las respuestas a las consultas.

Las demás clases que podemos encontrar en el apartado de PHP comprenden a la conexión.

```
require_once 'Mysql_login.php'; //Este archivo Database requiere el mysql_login

class Database //No hace falta modificar nada. Es un archivo de conexión con la base
{
    /**
     * @Única instancia de la clase
     */
    private static $db = null;

    /**
     * Instancia de PDO
     */
    private static $pdo;
```

```
1 <?php
2 /**
3  * Provee las constantes para conectarse a la base de datos
4  * Mysql.
5  */
6  header("Access-Control-Allow-Origin: *");
7  header("Content-Type: application/json; charset=UTF-8");
8
9  define("HOSTNAME", "localhost");// Nombre del host. Siempre va a ser
   localhost ya que lo normal es alojar los php en la misma máquina donde
   esta la base de datos
10  define("DATABASE", "id4499805_opendataplasencia");// Nombre de la base
   de datos. Me lo da "Hostinger" o la cuenta de hosting que haya
   contratado.
11  define("USERNAME", "id4499805_si");// Nombre del usuario. Me lo da
   "Hostinger" o la cuenta de hosting que haya contratado.
```

También hay clases generales de cada tabla, en el caso de la siguiente imagen la tabla gobierno tiene en php una clase la cual determina las llamadas de ejecución a la clase donde se encuentra la clase *encode* que se vió anteriormente.

```
class Gobierno {
    function __construct() {
    }

    /**
     * Retorna en la fila especificada de la tabla 'Gobierno'
     *
     * @param $id Identificador del registro. No utilizado aquí
     * @return array Datos del registro
     */
    public static function getAll() { //Me devuelve todos los registros de la tabla Gobierno
        $consulta = "SELECT * FROM gobierno";
        try {
            // Preparar sentencia
            $comando = Database::getInstance()->getDb()->prepare($consulta);
            // Ejecutar sentencia preparada
            $comando->execute();

            return $comando->fetchAll(PDO::FETCH_ASSOC);
        } catch (PDOException $e) {
            return false;
        }
    }
}
```


Esta clase mediante *\$comando* y el *getAll()* ejecuta las llamadas a la clase de encode.

E) HTML

Este proyecto tiene una arquitectura basada en AngularJs lo que hace de html un simple esclavo de dicha tecnología. Es por ello que toda la estructura de HTML se manifiesta y sitúa para la utilización de dicho framework.

Esto hace que la utilización de html sea muy simple y reducida. Haciendo que simples partes de código estén separados en el proyecto y se muestren en conjuntos gracias a las llamadas realizadas por las funcionalidades de Angular.

Se puede poner el simple ejemplo del *index*. Este se compone de pocos elementos html pero es la única página que se muestra en toda la web, tal como se describe en el apartado de *routing* del siguiente apartado de la descripción técnica.

```
30
31 <body >
32   <div ng-include src="'ciudad/view/menuPrincipal.html'"></div>
33
34   <div ng-view></div>
35
36   <div ng-include src="'ciudad/view/footer.html'"></div>
37
38
39   <script src="ciudad/script/app.js"></script>
40   <script src="ciudad/script/controllers/controller.js"></script>
41   <script src="ciudad/script/routines/routines.js"></script>
42   <script src="ciudad/plascencia/ayuntamiento/controller.js"></script>
43
44   </body>
45 </html>
46
```

Este es el body del index del portal web. Tan solo un par de *div* señalando a otras partes del directorio donde se almacenan elementos html.

Es de esta manera como toda la web esta compartimentada en pequeños trozos que se cargan en el index mediante la etiqueta de Angular *ng-view* junto con un menu y un pie de página que no cambiará en toda la web.

En la siguiente imagen se muestra el fichero que carga el pie de página de la web.

```
1 <!--
2 Created on : 15-abr-2018
3 Author : Ismael Martin Ramirez [imartinr01@informatica.iesvalledeljerteplascencia.es]
4 -->
5 <hr>
6 <footer class="container">
7 <address title="Contacto" class="footer">
8 Ismael Martín | Desarrollador de Apps Multiplataforma | 2018
9 </address>
10 </footer>
11 |
```

Aunque la estructura la pone AngularJS la semántica html se ha tenido muy en cuenta a la hora de realizar el portal web. Lo cortés no quita lo valiente. Esta es la arquitectura HTML si juntas todas las piezas.

```
1 <!--
2 Created on : 15-abr-2018
3 Author : Ismael Martin Ramirez [imartinr01@informatica.iesvalledeljerteplascencia.es]
4 -->
5 <!DOCTYPE html>
6 <html >
7 <head></head>
8
9 <body >
10 <nav></nav>
11
12 <header><h1></h1><h2></h2></header>
13
14 <section>
15 <table></table>
16 </section>
17
18 <section>
19 <div></div>
20 </section>
21
22 <footer></footer>
23
24 <script></script>
25
26 </body>
27 </html>|
```


F) ANGULAR JS

- Introducción

Angular es un framework de JavaScript mantenido por Google, el cual premia con opciones de SEO a las páginas que implementan este framework.

Con AngularJs se conseguirá una web fácil de manejar y compartimentada en secciones o “*controllers*” que hará el trabajo del desarrollador más sencillo, teniendo todo localizado y con la capacidad de enmendar errores de una manera más optima.

Dichos controllers serán los encargados, por separado, de dar funcionalidad a las distintas partes de la web. También controlarán las llamadas de muestro de las piezas HTML que tiene el proyecto.

Las etiquetas Angular en los elementos HTML son los encargados de activar los controllers.

- Routing

Se denomina *Routing* a la técnica utilizada en Angular para tener toda una web alojada en un solo fichero .html.

En este caso se ha visto anteriormente el body de la web con tres div. Uno de ellos es el encargado de realizar routing junto con el controller que le sirve el muestreo mediante el scripts de routing.

```
<body >  
  <div ng-include src="'ciudad/view/menuPrincipal.html'"></div>  
  
  <div ng-view></div>  
  
  <div ng-include src="'ciudad/view/footer.html'"></div>
```

```
<!--Configuracion de ngRoute -> Rautins-->
app.config(['$routeProvider',
  function($routeProvider){
    $routeProvider.
      when('/opendata/plasencia',{
        templateUrl:'ciudad/plasencia/plasencia.html',
      }).
      when('/opendata/plasencia/gobierno',{
        templateUrl:'ciudad/plasencia/ayuntamiento/gobierno.html',
      }).
      when('/opendata/plasencia/barrio',{
        templateUrl:'ciudad/plasencia/barrio.html',
      }).
      when('/opendata/plasencia/organismosSociales',{
        templateUrl:'ciudad/plasencia/organismo.html',
      }).
      when('/opendata/plasencia/gobierno/xml',{
        templateUrl:'ciudad/plasencia/ayuntamiento/export/xml.html',
      }).
      when('/opendata/plasencia/gobierno/json',{
        templateUrl:'ciudad/plasencia/ayuntamiento/export/json.html',
      }).
      when('/opendata/plasencia/gobierno/csv',{
        templateUrl:'ciudad/plasencia/ayuntamiento/export/csv.html',
      }).
      when('/opendata/plasencia/gobierno/html',{
        templateUrl:'ciudad/plasencia/ayuntamiento/export/html.html',
      }).
      otherwise({
        redirectTo:'/',
        templateUrl:'ciudad/plasencia/inicio/cabecera.html',
      })
  })
})
```

Como se puede observar en la imagen que están todas las url del proyecto junto con el fichero ("pieza html") que deben cargar cuando dichas urls sean llamadas. Las url serán llamadas mediante una etiqueta `` y el controller cargará en la etiqueta `ng-view` del index ese fichero .html.

Por consiguiente toda la web es el index en el que se cargan piezas html. La web esta alojada en index y cargar secciones independientes como se muestra en la siguiente imagen.

```
https://museumis.github.io/OpenDataPlasencia/#!/
https://museumis.github.io/OpenDataPlasencia/#!/opendata/plasencia/gobierno
https://museumis.github.io/OpenDataPlasencia/#!/opendata/plasencia/organismosSociales
```

Esto hace que la página sea rápida y con una utilización de urls muy limpia y estable.

- Carga de PHP

Otra cualidad de Angular es la posibilidad de obtener en una variable una respuesta `http.get` la cual se utilizará para la gestión de la web.

```
/*Metodo que gestiona la conexion http y las variable que se mostraran en la seccion html exports*/
$scope.cargarJsonFromUrl = function() {
  //Metodo que se conecta a la base de datos.
  $http.get('https://imartinr01.000webhostapp.com/OpenData/Plasencia/Ayuntamiento/Obtener_gobierno.php').then(function (flujo) {
    //Gestion de las variables en la web. La tabla y la seccion export.
    $scope.generarTabla(flujo)
    $scope.dataXml = $scope.convertirAXml(flujo)
    $scope.dataJson = $scope.convertirAJson(flujo)
    $scope.dataCsv = $scope.convertirACsv(flujo)
    $scope.dataHtml = $scope.convertirAHtml(flujo)
  })
}
```

La función *cargarJsonFromUrl* es la encargada de hacer la petición `get` al servidor mediante la función `$http.get('url')` que responderá con otra función que a partir del flujo de datos obtenido cargará la tabla que se muestra en la web y guardará en variables distintas los formatos que se muestran en la web.

La siguiente imagen muestra la “pieza html” de angular que muestra la variable `$scope.dataCsv` en la web, en el apartado de gobierno; ver datos en csv.

```
<p ng-controller="gobierno">{{dataCsv}}</p>
```

Esta variable esta dentro del controller de gobierno.

- Funciones de export

Ya que Angular es JavaScript existe la posibilidad de hacer que el cliente genere código a partir del existente o en otras palabras tiene la posibilidad de darle funcionalidad a la página en el momento que se esta usando.

Gracias a esto el portal web, junto al cliente, tiene la posibilidad de generar xml, csv, html a partir del JSON obtenido en la respuesta del servidor.

Se vió anteriormente como se guardaban en variables los distintos formatos de muestra, estas variables se iniciaban con una llamada al método que genera dichos formatos.

```
//Funcion que recibe la cadena json obtenida en la conexion http y la convierte a csv
$scope.convertirACsv = function(flujo){

    var csv = ""

    for (var i=0; i<$scope.titulos.length; i++) {

        if(i==$scope.titulos.length-1){
            csv += $scope.titulos[i]
        }else{
            csv += $scope.titulos[i] + ","
        }

    }

    csv+="\n"
    for (var i=0; i<flujo.data.gobierno.length; i++) {

        if(i==flujo.data.gobierno.length-1){
            csv += flujo.data.gobierno[i].NOMBRE + "," +
                flujo.data.gobierno[i].MINISTERIO + "," +
                flujo.data.gobierno[i].FECHA_INICIO + "," +
                flujo.data.gobierno[i].FECHA_FIN + "," +
                flujo.data.gobierno[i].TELEFONO + "," +
                flujo.data.gobierno[i].CORREOELECTRONICO + "," +
                flujo.data.gobierno[i].FAX + "," +
                flujo.data.gobierno[i].WEB + "," +
                flujo.data.gobierno[i].TWITTER + "," +
                flujo.data.gobierno[i].FACEBOOK

        }else{
            csv += flujo.data.gobierno[i].NOMBRE + "," +
                flujo.data.gobierno[i].MINISTERIO + "," +
                flujo.data.gobierno[i].FECHA_INICIO + "," +
                flujo.data.gobierno[i].FECHA_FIN + "," +
                flujo.data.gobierno[i].TELEFONO + "," +
                flujo.data.gobierno[i].CORREOELECTRONICO + "," +
                flujo.data.gobierno[i].FAX + "," +
                flujo.data.gobierno[i].WEB + "," +
                flujo.data.gobierno[i].TWITTER + "," +
                flujo.data.gobierno[i].FACEBOOK

        }

        csv+="\n"

    }

    return csv;
}
```

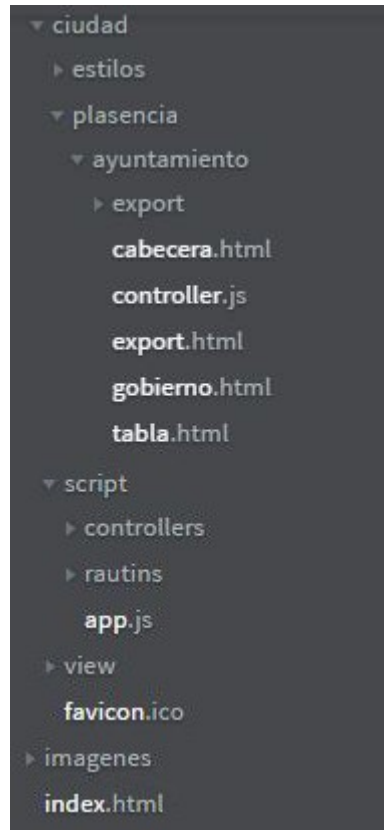
- Directorio

El directorio del proyecto esta estructurado siguiendo un modelo complementario con AngularJS.

En el directorio principal está el index, las imágenes y la carpeta ciudad que alberga todo el código de web. Dentro de ella se encuentra las carpetas; scripts, que afecta a todos los controlles generales de Angular como la forma de iniciar o el controller de routing, la carpeta estilo, con el fichero sass y el css compilado a partir de este y la carpeta plasencia.

La carpeta plasencia tiene distribuidas en carpetas todas las partes de la web. Estas carpetas contienen, por ejemplo en la carpeta

ayuntamiento, sus “piezas html” de angular, su controller y una carpeta donde se alojan las web de los formatos *export*



En *view* se guarda el menu principal y el footer presentes en toda la web.

G) CSS COMPILADO CON SASS

Sass es un herramienta que nos permite crear archivos Css de una manera mas programable. Nos permite guardar variables, no repetir etiquetas hijas de manera constante teniendo la forma de anidar elementos hijos sin salir de la etiqueta a la que se le esta dando estilo, entre otras funcionalidad.

Mediante NodeJS se compila este fichero que genera el Css listo para funcionar.

El proyecto no ha recibido mucho estilo gracias a la utilización de Bootstrap, pero algunos retoques han sido necesarios.

La siguiente imagen se muestra como se retoca el estilo del menú principal.

```
.navbar
border-radius: 4px 4px 0px 0px
background-color: $colorPrimary
a
    color: $colorText
.navbar-toggler-icon
    background: url("../imagenes/menuIcon.jpg")
li
    border-radius: 4px 4px 4px 4px
li:hover
    background-color: $colorPrimaryDark
```

Una vez compilado el fichero Css del menú principal queda de esta manera.

```
.navbar {
border-radius: 4px 4px 0px 0px;
background-color: #f8f8f8; }
.navbar a {
color: #272c31; }
.navbar .navbar-toggler-icon {
background: url("../imagenes/menuIcon.jpg"); }
.navbar li {
border-radius: 4px 4px 4px 4px; }
.navbar li:hover {
background-color: #babbbf; }
```

Se observa en esta imagen la repetición contante de la etiqueta .navbar, problema solventado con la utilización de SASS.

H) BOOTSTRAP

Bootstrap es una herramienta que nos permite dar estilo a elementos de la web con diseños predeterminados en dicho framework. Estos diseños de bootstrap están pensados para dispositivos móviles o navegadores de escritorio. Por ello con la utilización de bootstrap conseguimos un portal *Responsive Design*.

- Diseño

El diseño de la web se ha encauzado por Bootstrap tanto cabeceras, navs, estilos en tablas llevan estilos predeterminados de la

biblioteca de bootstrap lo que hace de ello una apariencia muy vistosa y agradable.

La siguiente imagen muestra como el header esta etiquetado con la etiqueta *jumbotron*, esta funcionalidad de bootstrap da un grandioso estilo al encabezado sin darle estilo de una manera predeterminada.

```
<header class="jumbotron">
  <div class="container">
    <h2>Ayuntamiento de Plasencia</h2>
    <p>Emplazado en la plaza mayor de Pl
      futuro de los ciudadanos. En esta se
    </div>
  </header>
```



Existen muchas clase de jumbotron, el necesario para este web es el más sencillo el que sólo implementa la clase *jumbotron* de la biblioteca de bootstrap, este viene con fondo predeterminado gris, y no era el adecuado para la web, también cuenta con un *border-radius* el cual no necesitamos.

Esto puede suceder en cualquier elemento que se desee implementar en la web mediante bootstrap. Es aquí donde entra en juego Sass y Css ya que gracias a ellos podemos corregir diseños predeterminados de bootstrap.

```
.jumbotron
  background-color: $colorPrimary
  border-radius: none
```

Otro ejemplo de diseño es como se da estilo a un simple *a* haciendo que parezca un botón.

```
<div class="col-sm-4">  
  <a href="#!/opendata/plasencia/gobierno/json" target="_blank" class="btn btn-info btn-lg col-xs-12">  
    <span class="glyphicon glyphicon-new-window"></span> Ver en Web  
  </a>  
</div>
```

Gracias a la *class="btn btn-info btn-log"* le damos estilo al hipervínculo, haciéndole parecer un botón.

Esto puede ocasionar pérdida de semántica en la arquitectura html pero siempre será un *<a>* con forma de botón.

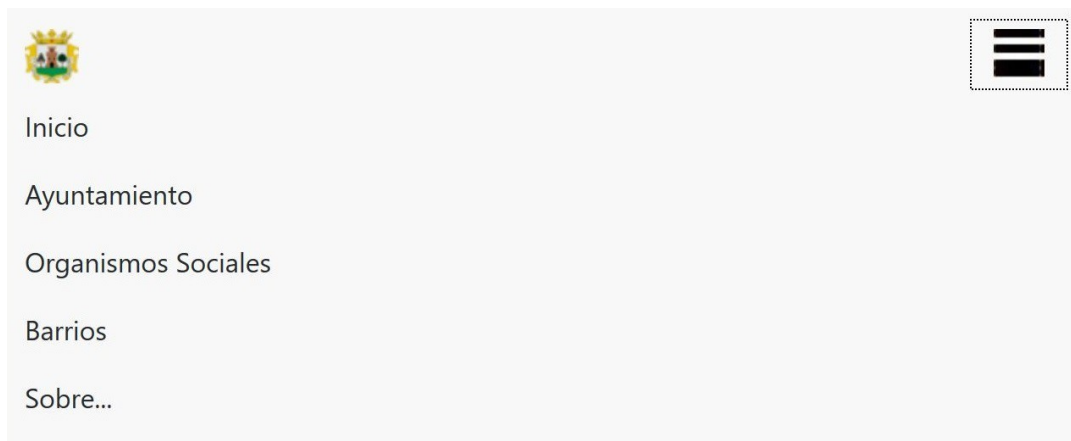
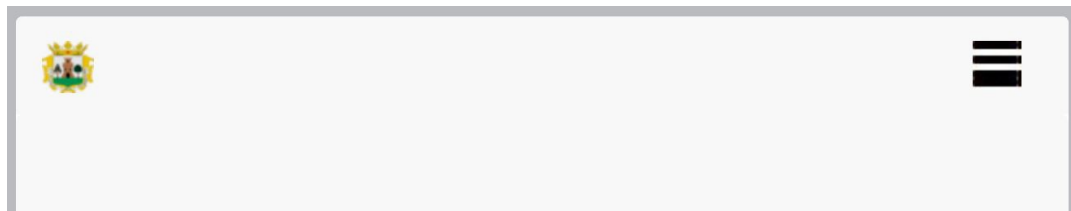
- Responsive Design

Gracias a las funciones predeterminadas de este framework podemos ver como el diseño del portal web cuadra para cualquier formato o pixelado en el que se desee ver.

En la siguiente imagen se puede observar el nav del menú principal en diseño web.



En esta otra se observa el nav del menú principal para la versión móvil. Es un nav que se convierte en lista con los elementos que contiene el nav. En la imagen se aprecia tanto en su estado oculto como plegado.



Este patrón de diseño lo siguen los elementos que implementan clases de bootstrap. Pero la colocación de elementos a la hora de insertarlos en bloques puede generar algún problema de diseño. Por ejemplo un *div* con varios *p* deben tratarse de una manera determinada.

La manera de tratar este problema es con la utilizando del sistema de rejilla.

Teniendo en cuenta cualquier pantalla o pixelado, el sistema de rejillas divide la pantalla en doce secciones, el desarrollador es el encargado de asignar que elementos deben ocupar doce secciones de la rejilla o que tres elementos deben ocupar cuatro secciones de la rejilla cada uno, en cada columna del elemento principal, tomando como columna una sección de un *div* al que se le pueden implementar columnas.

Es recomendado hacer todo el diseño de un elemento en una sola columna, para evitar saltos indeseados de elementos en los distintos tamaños de pantalla.

En este portal web se trata con mucho énfasis el sistema de rejillas en la parte donde el cliente tiene las opciones de export. Donde se muestran dos títulos un hipervínculo y un elemento pre.

```
<!--Seccion que se oculta y muestra al pulsar sobre las opciones la lista anterior-->
<!--XML-->
<div ng-if="xmlAyuntamiento" class="contenedorExport">
<div class="container">
  <div class="row">
    <div class="col-sm-4"><h3>Ayuntamiento de Plasencia</h3></div>
    <div class="col-sm-4"><h4>XML</h4></div>
    <div class="col-sm-4">
      <a href="#!/opendata/plasencia/gobierno/xml" target="_blank" class="btn btn-info btn-lg col-xs-12">...</a>
    </div>
    <pre class="col-xs-12">{{dataXml}}</pre>
  </div>
</div>
...

```

Pues bien gracias a la `class="container"` señalamos ese elemento div como contenedor de una rejilla. Con la clase `class="row"` señalaremos una fila y dentro de ella mediante `class="col-sm-4"` se otorgara a esos elementos cuatro espacios de rejilla y doce espacios para el elemento *pre*.

Esto hace que el diseño de la web sea equilibrado para todo tipo de pantallas. Este procedimiento se ha llevado acabo en todos los lugares donde se ha precisado una corrección de elementos.

5. MEDIOS UTILIZADOS

A) HOSPEDAJE

Los medios utilizados para el hospeda de la web son los siguientes.

Para mantener la base de datos activa se ha necesitado de un navegador web. Y en los inicios de la creación de dicha base de datos se necesitó XAMPP ya que se requería de un servidor localhost idóneo para las oportunas comprobaciones.

Para la sustentación de la web en GitHub se ha trabajado con GitHub Desktop. Con ello se obtiene un manejo fácil de subida de archivos al repositorio.



B) BASE DE DATOS

La base de datos se desarrolló en el entorno de trabajo PostgreSQL. Y actualmente se mantiene desde PhpMyAdmin.




C) PHP

En la creación y desarrollo de los archivos PHP se ha utilizado SublimeText.



Una vez en producción el hosting de 000webHost no da la opción de utilizar un entorno personalizado para el servidor, esta ha sido la manera para retocar algunos archivos PHP una vez la página era funcional.

 000webhost imartinr01 > public_h... > OpenData				
	Name ▼	Size	Date	Permissions
▼ public_html	Database.php	1.9 kB	2018-04-14 19:12:00	-rw-r--r--
> img	Gobierno.php	0.9 kB	2018-04-14 19:13:00	-rw-r--r--
▼ OpenData	Mysql_login.php	0.7 kB	2018-04-14 20:10:00	-rw-r--r--
▼ Plascencia	Obtener_gobierno.php	0.9 kB	2018-05-17 19:29:00	-rw-r--r--
▼ Ayuntamiento				
> opina				
> tmp				

Para la creación y mantenimiento de la arquitectura Angular se ha utilizado Brackets IDE que nos permite tener la web alojada en un host local.

E) **BOOTSTRAP**

Bootstrap necesita de un servidor alojado en un host para poder funcionar. Requiere de funcionalidades que solo pueden ser funcionales si están alojadas en la nube.

Es por esta razón que Brackets es un IDE idóneo para la realización de proyectos con Bootstrap.

Un IDE sencillo y comodo que te permite trabajar con las últimas tecnologías.



Es necesario que para poder ejecutar nuestra web desde un entorno de desarrollo debe implementar la opción de alojar la web en un host temporal.

Si deseas ejecutar este proyecto se recomienda hacerlo con Brackets desde la vista dinámica.

De ejecutarlo en un entorno que no tenga esta utilidad el proyecto no se ejecutará de forma adecuada.

F) **CSS COMPILADO CON SASS**

El desarrollo de los ficheros Sass es algo complejo ya que requiere de NodeJS una tecnología que implementa comandos de usos que se ejecutan atrás de la consola de comandos.



Tanto el fichero Sass o el Css se encuentran en el directorio del proyecto alojados con demás archivos que son necesarios para la compilación.

El modo de trabajo es realizar cambios en el fichero Sass y compilar de forma automática mediante comandos en la consola donde se ejecutará la compilación del fichero Css, se adaptará al fichero Sass para dar cobertura a los navegadores web.

```
ruby 2.5.1p57 (2018-03-29 revision 63029) [x64-mingw32]

C:\Users\Bri>cd C:\Users\Bri\Desktop\2ºDAM\6 - ProyectoFinCiclo\ODPlasencia\OpenDataPlasencia\ciudad\estilos

C:\Users\Bri\Desktop\2ºDAM\6 - ProyectoFinCiclo\ODPlasencia\OpenDataPlasencia\ciudad\estilos>sass generalcsssass.sass:generalcss.css

C:\Users\Bri\Desktop\2ºDAM\6 - ProyectoFinCiclo\ODPlasencia\OpenDataPlasencia\ciudad\estilos>
```

Se trata de ir a la ruta de los fichero e indicar que fichero es el Sass y a cual debe compilar.

El proyecto esta disponible en:

<https://github.com/museumis/OpenDataPlasencia>

Para una correcta visión de la web será necesario estar en una red libre de cortafuegos ya que el host de la base de datos es bloqueada en algunas redes.

Para ejecutar el proyecto desde un IDE será necesario que el ide tenga la opción de ejecutarlo en un servidor de prueba. Tal como se explica anteriormente con Brackets puede ejecutarse sin problemas.

El proyecto no está totalmente acabado a la hora de la entrega, pero si contiene todo lo suficiente para ser funcional. Las implementaciones que quedan pendiente son referidas al contenido de la web, no a la lógica de la web.

Desde un principio se ha hecho el proyecto con vista a realizar una implementación que corresponda de igual manera para todas las partes del proyecto. Es por eso que el actual código puede dar cabida a completar la web sin necesidad de cambios excesivos. Puedes tomar el código realizado para la parte del gobierno, y realizar la sección de asociaciones de vecinos sin mayor contratiempo que cambiar los textos o retocar los controllers de cada sección.

- (Video2Brain) AngularJS de 0 a 100
- (Video2Brain) CSS con LESS & Sass
- (Video2Brain) Desarrollo web con Bootstrap 3
- <https://www.w3schools.com/angular/default.asp>
- <https://www.w3schools.com/bootstrap/default.asp>
- <https://www.w3schools.com/php/default.asp>
- <https://nodejs.org/es/docs/>
- <http://php.net/manual/es/function.xmlrpc-encode.php>
- <https://es.stackoverflow.com/>
- <https://mediaqueri.es/>