



# **PROCESAMIENTO DE INFORMACIÓN EN APLICACIONES TELEMÁTICAS**

**Examen ordinario – 1 de junio  
Segundo parcial**

**Curso 2020/2021**

## 1 Descripción

Se pretende desarrollar una herramienta de búsqueda sobre el [Portal de Datos Abiertos del Ayuntamiento de Madrid](http://datos.madrid.es) (<http://datos.madrid.es>), similar a las desarrolladas en las prácticas, pero partiendo de otro tipo de información.

En el espacio Moodle de la asignatura, en el apartado “Examen Ordinario. 1 de junio”, dispondrá de los ficheros que se nombran en este enunciado y también encontrará la tarea de entrega.

## 2 Realización

Se deberá codificar una aplicación que reciba, a través de la línea de mandatos, los siguientes argumentos:

- (ARG0) Ruta al documento del catálogo, que obligatoriamente será el fichero `catalog-0040-022.json`.
- (ARG1) Código de la categoría de la que se desea información, que obligatoriamente será la `0040-022`.
- (ARG2) Ruta al documento XML de salida.
- (ARG3) Ruta al documento HTML de salida.

La aplicación deberá implementar las funcionalidades de las clases que se describen en los apartados siguientes.

### 2.1 Clase Principal

La clase principal de la aplicación `ExamenJunio` deberá generar un documento de salida XML, que sea válido conforme al documento `ExamenJunioSchema.xsd`, y un documento HTML aplicando una hoja de estilos XSL.

Consideraciones de realización:

- Verificará y validará los argumentos de entrada.
- Obtendrá un mapa con la información pertinente mediante la invocación al método `getResults()` de la clase `JSONParserJunio` (ver apartado 2.2).
- El mapa del apartado anterior se pasará a la clase `ResourcesToXMLJunio`, para que forme un `String` a partir del cual creará un fichero cuyo contenido será el documento XML de salida con el nombre indicado en ARG2 (ver apartado 2.3).
- Finalmente, mediante la clase `XMLTransformer`, y aplicando una hoja de estilos XSL, generará el documento HTML cuyo nombre se indica en el ARG3 (ver apartado 2.4).

## 2.2 Obtención de la información pertinente

La clase `JSONParserJunio`, que debe implementar un `JSONParser` usando el *model Stream*, se usará para extraer la información pertinente del documento indicado en ARG0.

Se considerarán *dataset* pertinentes aquellos cuyas *propiedades* de nombre `concept`, del array `concepts` del *dataset*, tienen algún valor que coincide con el valor de la *propiedad* de nombre `id`, descendientes del array `concepts`, donde la propiedad `code` tiene el valor del código de la categoría pasado en el ARG1.

Cuando un *dataset* es pertinente se extraerá el enlace que está en el valor de la propiedad con nombre `id`.

En la siguiente figura se muestra un ejemplo de un *dataset* pertinente para el código de categoría 0040-022:



```

{
  "id": "https://datos.madrid.es/egob/kos/entidadesYorganismos/CentrosAtencionSocial",
  "properties": {
    "code": "0040-022",
    "label": "Centros de atención social"
  }
},
{
  "id": "https://datos.madrid.es/egob/kos/entidadesYorganismos/CentrosAtencionSocial/AlberguesSocialesCentrosAcogida",
  "properties": {
    "code": "0001-0040FL22",
    "label": "Albergues sociales y centros de acogida"
  }
}
},
"datasets": [
  {
    "id": "https://datos.madrid.es/egob/catalogo/209094-0-centros-servicios-sociales.xml",
    "title": "Centros de Servicios Sociales Municipales",
    "properties": {
      "description": "Listado de Centros de Servicios Sociales Municipales, donde se ofrecen los siguientes servicios: Información, orientación y asesora",
      "publisher": "http://datos.gob.es/recurso/sector-publico/org/Organismo/L01280796",
      "theme": "http://datos.gob.es/kos/sector-publico/sector/sociedad-bienestar",
      "concepts": [
        { "concept": "https://datos.madrid.es/egob/kos/entidadesYorganismos/CentrosAtencionSocial/CentrosServiciosSociales" },
        { "concept": "https://datos.madrid.es/egob/kos/entidadesYorganismos/CentrosAtencionSocial" }
      ]
    }
  },
  {
    "id": "https://datos.madrid.es/egob/catalogo/201544-0-centros-salud.xml",
    "title": "Madrid Salud. Relación de centros municipales",
    "properties": {
      "description": "Centros de Salud Municipales Centros de Atención a Drogodependencias Centros especializados Centros monográficos Datos de ubicación",
      "publisher": "http://datos.gob.es/recurso/sector-publico/org/Organismo/A13002908",
      "theme": "http://datos.gob.es/kos/sector-publico/sector/salud",
      "concepts": [
        { "concept": "https://datos.madrid.es/egob/kos/entidadesYorganismos/CentrosAtencionMedica/CentrosAsistenciaDrogodependientes" },
        { "concept": "https://datos.madrid.es/egob/kos/entidadesYorganismos/AyuntamientoMadrid/OrganismosEmpresasMunicipales" },
        { "concept": "https://datos.madrid.es/egob/kos/entidadesYorganismos/CentrosAtencionMedica/CentrosPrevencionEnfermedades" },
        { "concept": "https://datos.madrid.es/egob/kos/entidadesYorganismos/CentrosAtencionSocial/CentrosRehabilitacionPsicosocial" },
        { "concept": "https://datos.madrid.es/egob/kos/entidadesYorganismos/AyuntamientoMadrid/OtrasEntidadesAdministrativasAyto" },
        { "concept": "https://datos.madrid.es/egob/kos/entidadesYorganismos/ResidenciasAnimales" }
      ]
    }
  }
]
}

```

Dataset pertinente para el código 0040-022

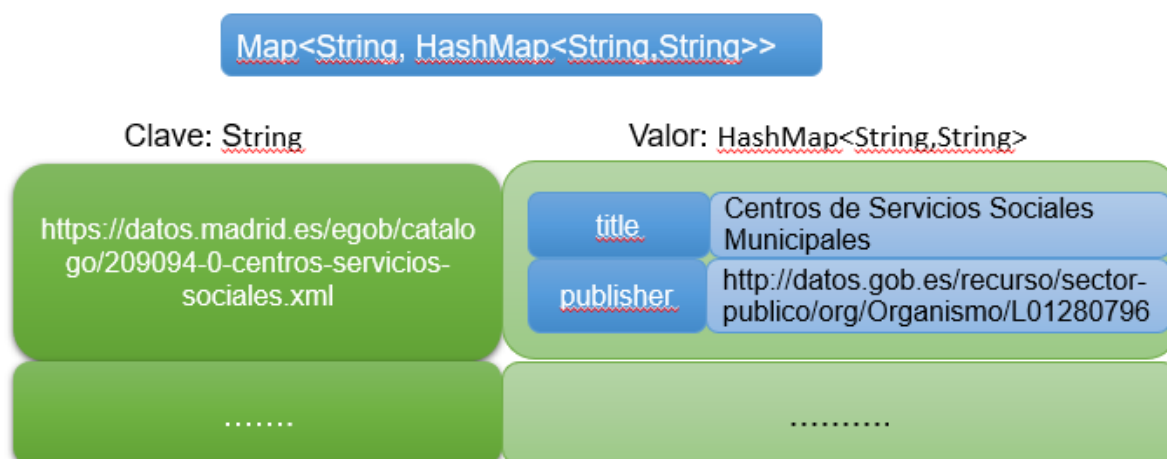
De cada *dataset* pertinente, se extraerá el valor de las propiedades de nombre `title` y `publisher`.

Con los datos pertinentes se generará un documento de salida XML, que sea válido conforme al documento `ExamenJunioSchema.xsd`. No es necesario implementar el proceso de validación.

**Consideraciones de realización:**

- El acceso a la información extraída por la clase `JSONParserJunio` se realizará mediante la invocación al método `Map <String, HashMap<String, String>> getResults()`.
- El mapa devuelto por `getResults()` contiene la siguiente información:
  - La **clave** de cada entrada será el valor de las propiedades de nombre `id` pertinentes del array `datasets`.
  - El **valor** de cada entrada será un `HashMap` con dos entradas:
    - Las **claves** serán `"title"` y `"publisher"`.
    - Los **valores** serán los correspondientes al valor de las propiedades de nombre `title` y `publisher`.

En la siguiente figura se muestra un ejemplo de como sería una entrada del mapa para un `dataset` pertinente para el código de categoría `0040-022`:



## 2.3 Generación del documento XML de salida

La clase `ResultsToXML`, a partir de la estructura de información disponible en el mapa `<String, HashMap<String, String>>`, generará un `String` cuyo contenido pueda ser posteriormente escrito en un fichero desde el programa principal.

**Consideraciones de realización:**

- El acceso a la información generada en la clase `ResultsToXML` se realizará mediante el método `String resultsToXML (Map <String, HashMap<String, String>>)`.
- El parámetro de tipo `Map <String, HashMap<String, String>>`, es el que se obtuvo al invocar el método `getResults()`.

## 2.4 Aplicación de hoja de estilos para crear un HTML de salida

La clase `XMLTransformer`, utilizando como fuente el documento XML generado, y una hoja de estilos que debe llamarse `HojaEstilos.xsl`, generará un documento HTML en el que se deberá visualizar, ordenados por el contenido del elemento `<publisher>`, el `<title>` de cada uno de los `<resource>` al que pertenece ese elemento `<publisher>`, tal y como se observa en la siguiente figura:

### Resultado

1 - <http://datos.gob.es/recurso/sector-publico/org/Organismo/L01280796>

Centros asociados:

- 1 - Centros de Servicios Sociales Municipales
- 2 - Puntos de atención a mujeres

2 - <http://datos.gob.es/recurso/sector-publico/org/Organismo/A13002908>

Centros asociados:

- 1 - Sedes. Centros de Atención Médica
- 2 - Sedes. Centros de Atención Social

## 3 Entrega

Como resultado de su examen, debe subir al espacio '[Entrega del examen](#)', un fichero comprimido en formato `.zip` (no se admitirá otro formato) que incluya, exclusivamente, los ficheros `.java` y `.xsl` realizados, y los ficheros XML y HTML obtenidos al ejecutar la aplicación (únicamente si ha conseguido que sean generados por la aplicación).

La entrega **se calificará con un 0** si:

- o Añade al fichero `.zip` el proyecto de eclipse. Este fichero debe contener únicamente ficheros fuente de java y los ficheros XML, XSL y HTML.
- o No ha escrito en cada fichero java su nombre y DNI, como comentario al inicio.
- o Se detecta que la entrega tiene un porcentaje de similitudes, respecto a entregas de otros alumnos, lo suficientemente grandes como para deducir que es una copia.

No entregue en ningún caso código procedente de las prácticas que ha realizado que no sea relevante para el examen.

## 4 Consejos de realización

Seguir estos consejos no es obligatorio, pero le ayudará a realizar este examen:

- 
- Haga un proyecto eclipse nuevo usando los ficheros java de plantilla que puede bajarse de moodle.
  - No copie y pegue todo el código usado en clases similares de las prácticas creyendo que así puede ahorrar tiempo porque justo le sucederá lo contrario. Es mejor que escriba lo que sea necesario para este examen, y en todo caso reutilice pequeños fragmentos de código de las prácticas.
  - Si su aplicación no ha sido capaz de generar el documento XML de salida, puede utilizar el que se proporciona como ejemplo en Moodle para comprobar el funcionamiento de la hoja de estilos XSL.