

G

MANUALES Y
GUIAS

Repositorios

2025

GeoREBIUN: LAS UNIVERSIDADES EN EL TERRITORIO. GEOLOCALIZACIÓN DE LA PRODUCCIÓN ACADÉMICA



crue

Universidades
Españolas

I+D+i

Red de Bibliotecas
REBIUN

GeoREBIUN: LAS UNIVERSIDADES EN EL TERRITORIO. GEOLOCALIZACIÓN DE LA PRODUCCIÓN ACADÉMICA

URI: <https://hdl.handle.net/20.500.11967/1461>

Documento elaborado por:

Fran Máñez (UPC)

Rebeca Marín (UC3M)

Paco Martínez (UPV)

Conchi Méndez (USC)

M^a Ángeles Molina (UM)

Lidón Paris (UJI)

Carmen Posse (USC)

Ana Poveda (UC3M)

Jordi Prats (UPC)

Toni Prieto (UPC)

Vicente Sabater (UPCT)

José Antonio Vidal (UPCT)

REBIUN Plan Estratégico 2024-2027.

Grupo de Trabajo de Repositorios

Documento bajo licencia Creative Commons

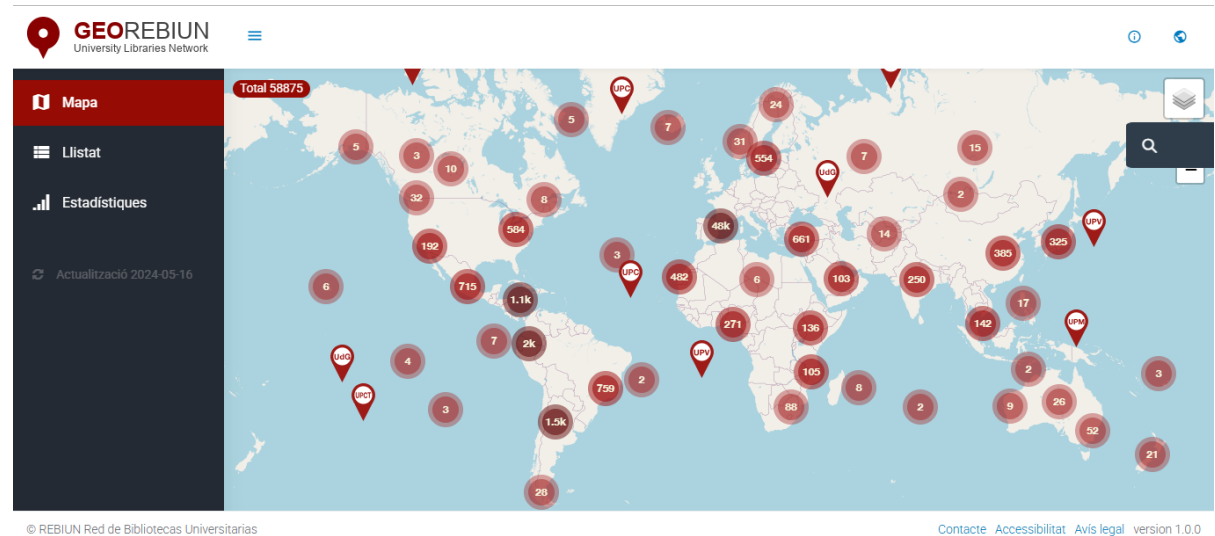
<https://creativecommons.org/licenses/by/4.0/>



Contenido

1.- Introducción.....	3
2.- Geolocalización de documentos.....	5
2.1.- Metadato y formato.....	5
2.2.- Implementación del selector de coordenadas en DSpace.....	5
2.2.1.- Versión 7.x.....	6
2.2.2.- Versiones anteriores (5.x / 6.x).....	7
2.3.- Implementación del selector en otro software.....	8
3.- Mostrar un mapa en pantalla de ítem con puntos geolocalizados en DSpace.....	9
3.1.- Versiones de DSpace.....	9
4.- Incorporación de registros al portal GeoREBIUN.....	10
4.1.- Template para datacite:geoLocations (v.4).....	10
4.2.- Definir un set con registros geolocalizados.....	12

1.- Introducción



<https://geo.rebiun.org/>

GeOREBIUN es el nombre que recibe el portal web que tiene como misión principal intentar mostrar la actividad que las universidades tienen en distintos puntos del territorio, a partir de la geolocalización de los documentos depositados en sus repositorios institucionales.

En este sentido, cabe destacar que para el desarrollo del proyecto se han definido tres objetivos principales:

- El desarrollo de una plataforma web que permita cosechar los registros previamente geolocalizados en los repositorios institucionales de las universidades españolas, para mostrarlos en un mapa mundial.
- Definir los estándares de metadatos que se van a utilizar para poder servir los registros al portal.
- Ofrecer herramientas que faciliten a los administradores y gestores de los repositorios la geolocalización de los registros en sus repositorios.

Aparte del mapa mundial con los documentos geolocalizados, GeOREBIUN ofrece otras funcionalidades que permiten filtrar sus recursos u obtener datos estadísticos, como pueden ser:

- Buscador global de contenidos, a partir de los campos de autor, título y materias.
- Filtrado de contenidos por facetas, tales como el nombre de la universidad, el tipo de documento, la fecha de publicación o las condiciones de acceso al documento.
- Vista de los contenidos a través de un listado, ordenable según distintos criterios.
- Estadísticas sobre el número de documentos geolocalizados por cada una de

las instituciones participantes.

Para la definición del modelo de metadatos que utiliza GeoREBIUN se han tenido en cuenta directrices muy usadas en el ámbito de los repositorios institucionales, como pueden ser las *OpenAIRE Guidelines for Literature, institutional, and thematic Repositories*,¹ tanto las versiones 3 como la última, la 4. Esto garantiza que el trabajo de geolocalización realizado por las universidades, más allá del proyecto GeoREBIUN, puede ser de utilidad para otros proyectos y compatible con otras iniciativas.

El presente documento es de carácter muy técnico y ofrece directrices a los gestores y administradores de los repositorios institucionales (principalmente para aquellos repositorios desarrollados con la plataforma DSpace²) para que puedan geolocalizar sus documentos e incorporarlos a GeoREBIUN.

En primer lugar, se aborda la implementación de un seleccionador de coordenadas que facilita la geolocalización de documentos desde el entorno de edición de registros del repositorio. Después se ofrece información sobre cómo visualizar la geolocalización desde la pantalla del ítem o registro. El proyecto no se ha centrado únicamente en la implementación del portal GeoREBIUN, sino que permite también enriquecer las plataformas de las propias universidades. Seguidamente, se documentan los esquemas de metadatos y vocabularios soportados por el proyecto, con el fin de poderlos facilitar a GeoREBIUN. Para finalizar se documenta cómo se debe configurar el servidor OAI-PMH para que los metadatos sean cosechables.

Como ya se ha comentado, la documentación se centra en su implementación con el software DSpace, aunque este no debe ser exclusivo. El uso de directrices y protocolos altamente usados garantiza que cualquier universidad interesada en el proyecto pueda participar en el proyecto.

¹ <https://guidelines.openaire.eu/en/latest/literature/index.html>

² <https://dspace.org/>

2.- Geolocalización de documentos

2.1.- Metadato y formato

Para la geolocalización de documentos en repositorios DSpace, se utiliza el metadato Dublin Core “dc.coverage.spatial” con el siguiente formato:

```
east=LONGITUD; north=LATITUD; name=NOMBRE
```

Por ejemplo, un documento con dos puntos geolocalizados incluirá los metadatos en este formato:

```
dc.coverage.spatial= east=13.883333300000004; north=-14.3166667;  
name=Quilengues, Angola
```

```
dc.coverage.spatial= east=14.448020000000042; north=-11.71077;  
name=Seles, Angola
```

Se puede ver el ejemplo en:
<https://upcommons.upc.edu/entities/publication/8a2e9acf-0b2c-41f2-98bc-a61da407e726/full>

2.2.- Implementación del selector de coordenadas en DSpace

Para facilitar la inclusión de estos metadatos se han preparado cambios para diferentes versiones de DSpace que incluyen un selector de coordenadas utilizando librerías de Google Maps.

En este documento se referencian los cambios para versiones 7.x. Para información más actualizada sobre estos cambios se puede visitar el repositorio GitHub:
<https://github.com/toniprieto/documentacion-georebiun>

Cada uno de los cambios está disponible en los últimos *commits* de una rama de desarrollo en Github. Se pueden aplicar estos *commits* directamente accediendo a la URL del *commit* y añadiendo ".diff" al final, y aplicando los cambios via *patch* desde la carpeta raíz del código fuente. Por ejemplo:

```
wget https://github.com/toniprieto/dspace-  
angular/commit/5316ae8c01c9449b85931f747cc41b0c5173e545.diff patch -p1  
< 5316ae8c01c9449b85931f747cc41b0c5173e545.diff
```

Para poderlo utilizar es necesario registrar una *ApiKey* de Google Maps siguiendo los pasos descritos en:
<https://developers.google.com/maps/documentation/javascript/get-api-key?hl=ES>

NOTA: desde hace unos años Google requiere añadir una cuenta de facturación al solicitar el alta de la *ApiKey*. Aunque por el uso que se da al selector es extraño superar el número de consultas gratuitas, para más seguridad se pueden configurar límites y restricciones de uso para no acarrear gastos.

A continuación, se incluyen enlaces y comentarios para la implementación en diferentes versiones de DSpace:

2.2.1.- Versión 7.x

En versiones 7.x y posteriores los cambios se han desarrollado en el código de `dspace-angular`.

Incluye nuevas opciones de configuración a añadir a los archivos `config.*.yml` para incluir la `ApiKey` de Google, el metadato usado y el inicio del mapa:

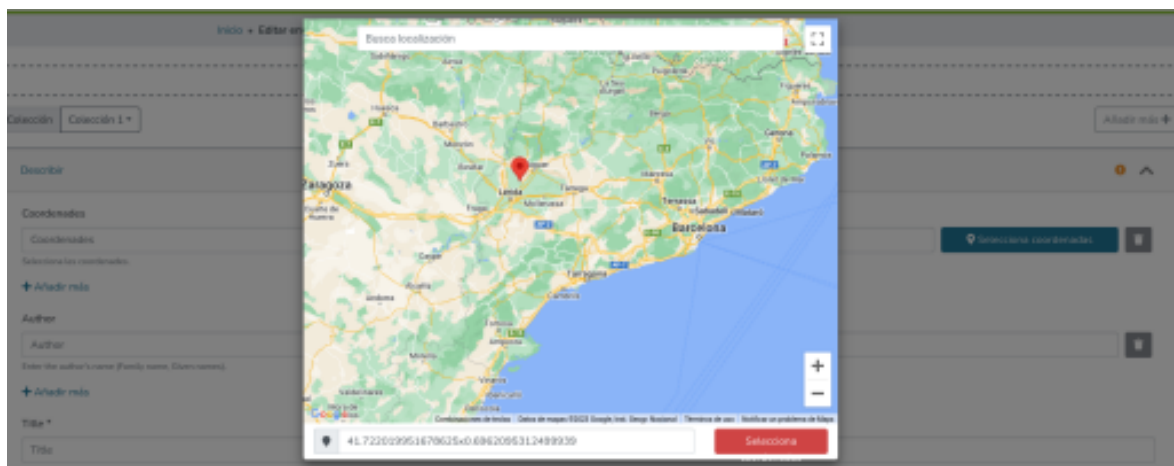
```
LocationPicker:  
  googleApiKey: YOUR-GOOGLEMAP-API-KEY  
  metadata: dc.coverage.spatial  
  
  zoom: 8  
  
  center:  
    Lat: 41.48378  
    Lng: 1.11369
```

Aunque es posible configurar el metadato utilizado, en una parte del código el metadato está añadido directamente y se debería modificar si se cambia (en `src/app/shared/form/builder/parsers/onebox-field parser.ts`).

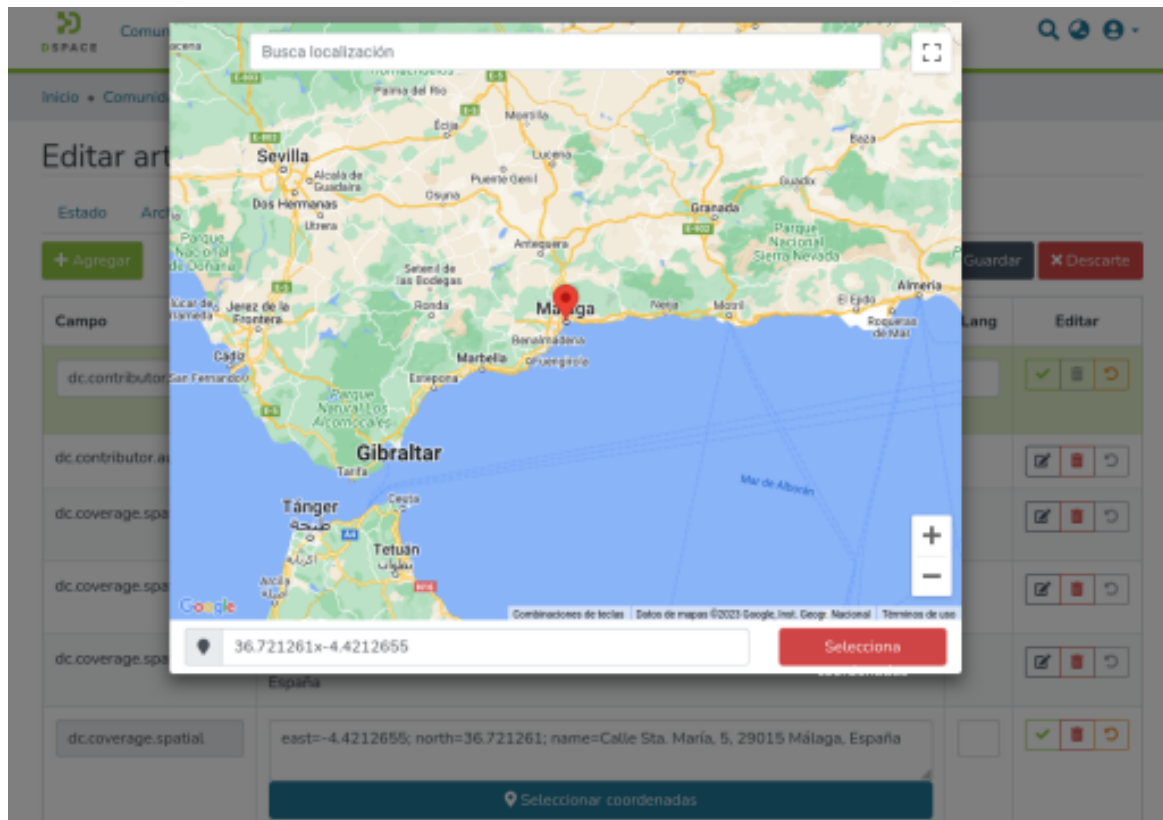
Una vez aplicado el cambio es necesario lanzar `yarn install` para descargar las nuevas librerías utilizadas.

El selector se incorpora tanto al formulario de entrada como al de edición para el campo “`dc.coverage.spatial`”. Este es el resultado esperado:

Entrada:



Edición:



Versiones:

7.6.x:

<https://github.com/toniprieto/dspace-angular/commits/selector-coordenadas-7.6>

Esta versión no incluye los cambios para añadir nuevos textos en los archivos i18n. Consultar información primer *commit* para más información.

7.5:

<https://github.com/toniprieto/dspace-angular/commits/selector-coordenadas-7.5.7.4>:

<https://github.com/toniprieto/dspace-angular/commits/selector-coordenadas-7.4-v1>

2.2.2.- Versiones anteriores (5.x / 6.x)

Existen también cambios para versiones anteriores que utilizan XMLUI para la versión 5.x, y que se debería poder aplicar también en versiones 6.x:

https://github.com/toniprieto/DSpace/commits/5_x-selecciona-coordenadas/

(incluye también cambios para mostrar los puntos geolocalizados en el registro de ítem)

En este caso se debe sustituir el <GOOGLE-API-KEY> por una clave Google propia, en la siguiente línea:

<https://github.com/toniprieto/Dspace/blob/d76a3f99a1876226fe11ffd5afb7694bf1dc1d2e/dspace-xmlui/src/main/webapp/static/coordenadas.html#L351>

2.3.- Implementación del selector en otro software

En la carpeta html del repositorio Github [Documentación GeoRebiun](#) se incluyen dos archivos HTML de ejemplo para implementar un selector de coordenadas en otros software diferentes, usando el mismo método implementado en DSpace:

- Formulario inicial: [form.html](#)
- Pantalla con selector de coordenadas: [coordenadas.html](#)

Como en el caso anterior, es necesario sustituir el `<GOOGLE-API-KEY>` por una clave Google propia en el archivo [coordenadas.html](#)

3.- Mostrar un mapa en pantalla de ítem con puntos geolocalizados en DSpace

También se han desarrollado cambios para poder mostrar los puntos geolocalizados en la pantalla principal de un registro de DSpace. En este caso, se utiliza la librería Leaflet³ para mostrar un mapa con los puntos seleccionados:

Localización

- 📍 Quilengues, Angola
- 📍 Seles, Angola



3.1.- Versiones de DSpace

7.6.x

<https://github.com/toniprieto/dspace-angular/commits/mostrar-mapa-coordenadas-7x/>

Las instrucciones para aplicar el cambio están en la descripción del primer *commit*. Las instrucciones incluyen un paso para copiar imágenes de Leaflet para mostrarlas correctamente.

5.x (6.x)

<https://github.com/toniprieto/DSpace/commit/4a06de211f9430b0860ad89f22542ba6f77f29c> f Es un *commit* dentro de la rama que implementa el selector para la versión 5.x

³ <https://leafletjs.com/>

4.- Incorporación de registros al portal GeoREBIUN

Para poder recolectar los registros con puntos geolocalizados para el proyecto GeoREBIUN el repositorio debe servirlos a través de su servidor OAI-PMH con un punto de acceso dedicado o mediante un set específico que incluya todos estos documentos. El formato del servidor OAI-PMH debe seguir las especificaciones *OpenAIRE Guidelines for Literature, institutional, and thematic Repositories*,⁴ pudiendo utilizar tanto la versión 3 como la 4.

A continuación se detallan, para cada una de las versiones, los metadatos básicos que deben servirse en el servidor OAI-PMH:

Versión 3

- Esquema de metadatos “oai_dc”.
- dc.title
- dc.creator
- dc.rights (según el vocabulario Driver <https://wiki.surfnet.nl/display/standards/info-eu-repo/#infoeu-repo-AccessRights>)
- dc.subject
- dc.date
- dc.type (según el vocabulario Driver <https://wiki.surfnet.nl/display/standards/info-eu-repo/#infoeu-repo-Publicationtypes>)
- dc.coverage (siguiendo el formato descrito en el punto 2.1)

Versión 4

- Esquema de metadatos “oai_openaire”.
- datacite:title
- datacite:creator
- datacite:rights (según el vocabulario COAR https://vocabularies.coar-repositories.org/documentation/access_rights/)
- datacite:subject
- datacite:date
- Oaire:resourceType (según el vocabulario COAR https://vocabularies.coar-repositories.org/documentation/resource_types/)
- Datacite:geoLocation (siguiendo las indicaciones de las *Guidelines* para la geolocalización de puntos).

4.1.- Template para datacite:geoLocations (v.4)

Para mostrar los puntos geolocalizados en el formato oai_openaire de la versión 4 de las *Guidelines*, usando el metadato <datacite:geoLocations>, se puede utilizar esta template:

⁴ <https://guidelines.openaire.eu/en/latest/literature/index.html>

```

<xsl:template name="GeoLocation_R">
  <xsl:if
test="doc:metadata/doc:element[@name='dc']/doc:element[@name='cov
erage']/doc:element[@name='spatial']/doc:element/doc:field[@name=
'value']">
    <datacite:geoLocations>
      <xsl:for-each

select="doc:metadata/doc:element[@name='dc']/doc:element[@name='c
overage']/doc:element[@name='spatial']/doc:element/doc:field[@nam
e='value']">
        <xsl:variable name="Locationname">
          <xsl:choose>
            <xsl:when test="contains(substring-
after(.,'name='),' ;')">
              <xsl:value-of select="substring-
before(substring-after(.,'name='),' ;')"/>
            </xsl:when>
            <xsl:otherwise>
              <xsl:value-of select="substring-
after(.,'name=')"/>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:variable>
        <xsl:variable name="Longitude">
          <xsl:choose>
            <xsl:when test="contains(substring-
after(.,'east='),' ;')">
              <xsl:value-of select="substring-
before(substring-after(.,'east='),' ;')"/>
            </xsl:when>
            <xsl:otherwise>
              <xsl:value-of select="substring-
after(.,'east=')"/>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:variable>
        <xsl:variable name="Latitude">

```

```

        <xsl:choose>
            <xsl:when test="contains(substring-
after(., 'north= '), ';')">
                <xsl:value-of select="substring-
before(substring-after(., 'north= '), ';')"/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="substring-
after(., 'north= ')/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <datacite:geoLocation>
        <datacite:geoLocationPoint>
            <datacite:pointLongitude><xsl:value-of
select="{$Longitude}"/></datacite:pointLongitude>
            <datacite:pointLatitude><xsl:value-of
select="{$Latitude}"/></datacite:pointLatitude>
        </datacite:geoLocationPoint>
        <datacite:geoLocationPlace>
            <xsl:value-of select="{$Locationname}"/>
        </datacite:geoLocationPlace>
    </datacite:geoLocation>
</xsl:for-each>
</datacite:geoLocations>
</xsl:if>
</xsl:template>

```

4.2.- Definir un set con registros geolocalizados

Se puede crear un set “geo” en el servidor OAI_PMH con los registros geolocalizados en DSpace con la siguiente configuración.

1. Crear una condición (dc.coverage.spatial que empiece por east=)

```

<CustomCondition id="metadatosGeoCondition">
    <Class>org.dspace.xoai.filter.DSpaceAtLeastOneMetadataFilter</Cla

```

```

SS>
  <Configuration>
    <string name="field">dc.coverage.spatial</string>
    <string name="operator">starts_with</string>
    <string name="value">east=</string>
  </Configuration>
</CustomCondition>

```

2. Definir un filtro:

```

<Filter id="geoFilter">
  <Definition>
    <And>
      <LeftCondition>
        <Or>
          <LeftCondition>
            <Custom ref="itemAccessCondition"/>
          </LeftCondition>
          <RightCondition>
            <Custom ref="itemWithdrawnCondition"/>
          </RightCondition>
        </Or>
      </LeftCondition>
      <RightCondition>
        <Custom ref="metadatosGeoCondition"/>
      </RightCondition>
    </And>
  </Definition>
</Filter>

```

3. Declarar el set con el filtro creado:

```

<Set id="geo">
  <Spec>Geo</Spec>
  <Name>Geo set</Name>

```

```
<Filter ref="geoFilter" />  
</Set>
```

 **crue** Universidades
Españolas

I+D+i

Red de Bibliotecas
REBIUN