TRABAJO DE FIN DE MÁSTER

PORTADA

VISTO BUENO DEL TUTOR Y DECLARACIÓN DE BUENA PRÁCTICA ACADÉMICA

SUMARIO O ÍNDICE

RESUMEN

INTRODUCCIÓN (ESTRUCTURA) O PRÓLOGO

ESTADO DE LA CUESTIÓN

OBJETIVOS GENERALES

METODOLOGÍA

PRESENTACIÓN DE LOS RESULTADOS

CONCLUSIONES

REFERENCIAS

ANEXOS

1. Introducción ( + 2 páginas)

El consiguiente trabajo de fin de Máster consiste en el desarrollo, diseño e implementación de un portal web destinado al Libro 1 de la obra Elegías del poeta griego Teognis de Megara.

La creación de este portal web reside en la inexistencia de una sitio web dedicado a la obra de este autor; así como de uno más genérico que recoja su obra dentro del mundo hispanohablante. Ante esta situación, se opta por la creación del mismo con el objetivo de favorecer y promover la investigación en torno a este autor y la poesía griega en general, gracias a la enmarcación de este proyecto en el ámbito de las Humanidades Digitales.

1. Contextualización. Estado de la cuestión (4 páginas)
   1. Contextualización del trabajo

Este trabajo se ha desarrollado como solución a algunas necesidades detectadas tras trabajar en el proyecto ERC POSTDATA (<http://postdata.linhd.es/>).

El proyecto, cuyo acrónimo significa “Poetry Standardization and Linked Open Data” nace en el seno de la UNED (Universidad Nacional de Educación a Distancia) y más concretamente en el LINHD (Laboratorio de Innovación de Humanidades Digitales).

Este tiene por objetivo crear una plataforma digital para la edición de poesía dirigida a diferentes tipos de usuarios: académicos que quieran trabajar en ediciones críticas; usuarios no experimentados que quieran leer, compartir y aprender más sobre tradiciones poéticas; y empresas que podrán utilizar este recurso para diferentes aplicaciones en campos como la educación, el turismo, la psicología u otros propósitos culturales: todo ello con el afán de construir puentes entre la filología y la informática, entre los usuarios y las metodologías.

Uno de los primeros pasos en este proyecto ha consistido en la búsqueda y análisis de las diferentes bases de datos de poesía en el continente europeo, gracias al cual se ha podido determinar el panorama actual en lo que se refiere a la situación de la poesía digitalizada que circula por la web.

* 1. Estado de la cuestión

Tras una búsqueda exhaustiva se ha realizado una clasificación de los proyectos de poesía digitalizada en Europa y que han tenido interés en unirse al proyecto. El conjunto seleccionado se ha dividido atendiendo a la temática de los corpus tratados en sus webs, quedando la clasificación de la siguiente manera:

<http://postdata.linhd.es/partners/>

Bases de datos de poesía de la Edad Media:

* Répertoire métrique de la poésie lyrique occitane des trobadours à leurs héritiers
* Analecta Hymnica Digitalia (<http://webserver.erwin-rauner.de/crophius/Analecta_conspectus.html>)
* ReMeTca: Repertorio Métrico Digital de la Poesía Medieval Castellana (<http://remetca.uned.es>)
* Bibliografia Elettronica dei Trovatori ([www.bedt.it](http://www.bedt.it))
* MedDB Base de datos da Lírica Profana Galego-Portuguesa ([www.cirp/gal.meddb](http://www.cirp/gal.meddb))
* The Oxford Cantigas de Santa Maria Database (<http://csm.mml.ox.ac.uk/>)
* Le Noveau Naetebus (<http://nouveaunaetebus.elte.hu>)
* Cantigas de Santa Maria for singers (<http://www.cantigasdesantamaria.com>)
* Répertoire de la poésie hongroise ancienne (<http://rpha.elte.hu>)
* Lyrik des hohen Mittelalters (<http://lhm-online.de>)
* Corpus Rhythmorum Musicum (<http://corimu.unisi.it>)
* Digital Edition of the Index of Middle English Verse (<http://dimev.net>)
* Skaldic Poetry of the Scandinavian Middle Ages (<http://skaldic.abdn.ac.uk/db.php>)
* Littera – Cantigas Medievais Galego – Portuguesas
* LIO: Lirica Italiana Degli Origini (<http://mirabileweb.it>)
* The last song of trobadours (<http://icalia.es/trobadours/ca>)
* Corpus des Trobadours (<http://trobadors.iec.cat>)

Bases de datos de poesía clásica:

* Pedecerto ([www.pedecerto.eu](http://www.pedecerto.eu))

Bases de datos de poesía moderna:

* Corpus of Spanish Golden-Age Sonnets (<https://github.com/bncolorado/CorpusSonetosSigloDeOro>)
* Sound and Metre in Italian Narrative Verse (<http://www.italianverse.reading.ac.uk>)
* Eighteenh-Century Poetry Archive (ECPA) (<http://eighteenthcenturypoetry.org>)
* Nederlandse Liederenbank ([www.liederenbank.l](http://www.liederenbank.l))
* Versologie ([www.versologie.cz](http://www.versologie.cz))
* Kalevala poetry (<http://dbgw.finlit.fi/skvr/>)
* Métrique en Ligne ([www.crisco.unicaen.fr/velaine](http://www.crisco.unicaen.fr/velaine))
* Russian Academy of Science (<http://ruscorporaru/>)
* Dainu Skapis (<http://dainuskapis.lv>)
* Eesti Regilaulude Andmebaas (<http://www.folklore.ee/regilaul/andmebaas>)

Si bien es cierto, todas estas bases de datos o portales web no son las únicas circulando por la red, sino simplemente aquellas que han mostrado interés en formar parte de un proyecto todavía mayor.

Si analizamos este conjunto de bases de datos, resalta el hecho de que solo una de ellas recoge un corpus de poesía clásica –latina en este caso-, lo cual deja ya entrever la escasez de portales web y bases de datos dedicados a este tipo de poesía.

En relación con la poesía clásica, de textos griegos y latinos, existen otras páginas web o bases de datos:

* PERSEUS (<http://www.perseus.tufts.edu/hopper/>) Es el portal web de textos clásicos más utilizado y completo entre los existentes. Además de recoger algunos corpus de poesía moderna, cuenta especialmente con un apartado destinado a obras griegas y latinas, con un corpus de lo más completo y exhaustivo. En lo que se refiere a estas obras, ofrece las mismas en versión original, así como una traducción al inglés. Cada una de las palabras de los textos originales constituye un hipervínculo a otra página interna de Perseus, donde se ofrece información gramatical de la misma en forma estadística atendiendo a las diferentes opiniones de los usuarios; así como con su localización en el texto.

El portal cuenta además con diferentes comentarios críticos de las obras.

* BIBLIOTHECA AUGUSTANA (<https://www.hs-augsburg.de/~harsch/augustana.html>). Es un portal web de textos clásicos muy completo entre los existentes. Cuenta con un corpus infinito de obras de la época clásica (griegas, latinas, hispánicas, gálicas, itálicas, etc). Se recogen las obras en su lengua original, pero no ofrece traducción de las mismas a una lengua moderna.
* THE LATIN LIBRARY (<http://www.thelatinlibrary.com/>) Es un portal web de textos clásicos latinos, cuyo corpus radica en el conjunto de obras latinas (la mayoría) sin traducción a ninguna lengua moderna.
* WIKISOURCE (<https://es.wikisource.org/>). En algunos casos encontramos aquí las obras completas de clásicos en lenguas modernas.
  1. Problemas detectados

Este análisis deja entrever la inexistencia de un portal web de obras clásicas con su traducción al español, lo cual deja aislada a la comunidad hispanohablante en este ámbito que debe recurrir siempre o casi siempre a trabajos o portales de origen anglosajón para sus investigaciones.

Asimismo y gracias a un análisis exhaustivo de cada una de las bases de datos participantes en el proyecto POSTDATA se detecta una inconsistencia en la conceptualización de los distintos términos en las bases de datos: esto es, los mismos términos tienen nombres diferentes en las distintas bases de datos, cuyos responsables no siguen un patrón a la hora de conceptualizar los elementos. Este análisis fue realizado de forma individual con especial atención a las bases de datos cuyo corpus recogía poesía de índole clásica; y con la ayuda de otros investigadores del proyecto (Helena Bermúdez, Mariana Curado Malta y Elena González-Blanco, investigadora principal del mismo).

Ver anexo (<https://zenodo.org/record/573767#.WYA0B9Pyi3U>)

1. Objetivos generales / Motivaciones (METER MUCHO MÁS)

En el contexto explicado previamente y en el del proyecto POSTDATA, surge la idea de la creación de un portal web dedicado a la obra de Teognis de Megara, poeta griego del siglo VI a.C, por su complejidad lingüística y por su corpus rico en topónimos y nombres propios, lo cual permite una mayor explotación del sitio creado.

Se realiza, por ello, un análisis de necesidades (ver anexo) que permite fijar los objetivos básicos del proyecto, así como el perfil del grupo meta al que va destinado el mismo.

Se establecen los siguientes objetivos:

* Facilitar a los lectores y estudiosos de la obra de Teognis de Megara la comprensión, interacción e investigación de la misma a través de un portal *user-friendly* con múltiples funcionalidades.
* Aportar a la comunidad académica hispanohablante un recurso más en español en relación con el mundo clásico.
* Aportar una edición única de la obra, con elementos inexistentes hasta ahora como: búsqueda de nombres propios que aparecen en la obra con descripción correspondiente, geolocalización actual de los topónimos de la obra, imagen visual de los tipos de verso atendiendo a la métrica de los mismos; así como otros elementos ya conocidos hasta entonces, como breve descripción de los personajes en el propio texto al pasar el ratón por encima de los mismos, o enlace de cada uno de los términos de la obra en griego con su entrada correspondiente en un diccionario de léxico griego antiguo
* Creación de un pequeño recurso poético en la web con una base de datos estándar desde el punto de vista tecnológico, que conceptualiza los términos seleccionados siguiendo el estándar de la Text Encoding Initiative (TEI)

1. METODOLOGÍA/DESARROLLO (15 páginas)

Para la realización de este trabajo se han seguido distintos pasos y utilizado diferentes tecnologías y lenguajes que han permitido el desarrollo e implementación del portal web como se presenta.

A continuación se mencionan de forma esquemática los distintos procedimientos llevados a cabo que, posteriormente, se desarrollarán en profundidad:

1. Proposición del corpus
2. Análisis de necesidades
3. Realización de una encuesta para el análisis de necesidades
4. Selección del corpus
5. Determinación de las funcionalidades de la página web
6. Comienzo del trabajo
7. Recopilación del texto griego digitalizado (del portal web de perseus)
8. Ante la inexistencia del texto español digitalizado + problemas para escaneo OCR + copia manual del texto en un editor de texto plano (SublimeText 3)
9. Creación de los documentos necesarios en XML
10. Etiquetado de los nombres de personas y topónimos siguiendo TEI (Text Encoding Initiative)
11. Búsqueda de los lugares aparentes en la obra en la actualidad y pequeña descripción
12. Geolocalización en Googlemaps de los topónimos (haciendo uso de la herramienta de GoogleMaps para la creación de mapas personalizados)
13. Búsqueda de información de cada uno de los personajes de la obra.
14. Elaboración de una tabla por cada personaje (nombre en griego, nombre en español, breve descripción)
15. Elaboración de un documento en PDF por cada personaje con la tabla anterior
16. Elaboración de documento HTML inicial del texto en griego y del texto en español a través de una hoja de transformaciones XSLT
17. Adaptación manual de los documentos HTML mencionados anteriormente
18. Comienzo de las funcionalidades:
19. Elaboración del primer buscador de nombres de personas en la obra (xPath)
20. Enlace del buscador con documento previo XML– (diferentes problemas)
21. Creación del buscador para los lugares en la obra / funcionamiento (xPath)
22. Creación de un programa en Phyton para enlace de cada término en griego con un diccionario léxico.
23. Creación de los tooltip en el texto en español, con información breve y descriptiva de los personajes principales
24. Contextualización de la página web: vida y obra de Teognis, lengua y métrica. Extracción de la información de XXX
25. Elaboración del HTML con esta información
26. Redacción y elaboración de la guía de usuario para la navegación por la página web
27. Diseño de la página web / Hoja de estilos CSS3 / Importación del framework Bootstrap.

El proyecto comienza con la proposición del corpus seleccionado, así como con la verificación de que el mismo es interesante y de utilidad para la comunidad académica y para los futuros usuarios –en general- del portal web. Tras realizar un análisis de necesidades que sirvió de ayuda para determinar los objetivos del trabajo y con el fin de asegurar la necesidad y utilidad de la creación de un portal de esta índole se realiza una encuesta a alumnos del grado de Filología Clásica de la Universidad Complutense de Madrid, así como del Máster Interuniversitario en Filología Clásica de las Universidades Complutense y Autónoma de Madrid, cuyos perfiles encajan con el grupo meta del proyecto. Esta encuesta es completada por hasta dieciocho alumnos cuyas respuestas determinaron la necesidad de la creación del portal y de una traducción al español de la obra, así como la utilidad de las diferentes funcionalidades de la página que favorecen la interacción del usuario con la obra.

En esta encuesta se preguntó acerca de la necesidad y utilidad de crear un portal web sobre la obra de Teognis.

Tras el análisis de los resultados se determinó que el corpus seleccionado para la página web era el adecuado, por lo que se procedió a la elección de las funcionalidades que tendría la página web. De nuevo, atendiendo a los resultados de la encuesta se fijaron las distintas funcionalidades:

* Buscador de nombres de personajes del texto, con una pequeña descripción de cada uno de ellos.
* Buscador de nombres de lugares del texto, con geolocalización de los mismos a través del framework de GoogleMaps
* Marcador de versos atendiendo a tu tipología métrica (pentámetros, hexámetros o ambos)
* Enlace de cada término en griego con su entrada correspondiente en el diccionario de léxico griego ΛΟΓΕΙΟΝ
* Descripción breve de cada personaje al pasar el ratón por encima en el texto en español

Una vez fijado este marco teórico se procedió a comenzar el proyecto desde un punto de vista tecnológico.

En primer lugar se procedió a la obtención de los textos en formato digital: tanto el texto en griego como en castellano. Debido a la fragmentación del Libro II de las Elegías de Teognis, así como a sus fundadas dudas de autoría, se optó por acotar el corpus teognídeo del Libro I del mismo, de mucha mayor extensión, completitud y autoría prácticamente asegurada.

Para la obtención del texto en griego se hizo uso del encontrado en el portal web Perseus.

Por su parte, el texto en español no se encontraba digitalizado ni circulaba por la web en ningún caso. Por ello, se procedió a la transcripción manual del mismo a partir de la edición XXXX, debido a la imposibilidad de utilización de un escáner OCR. Estos textos se escribieron o copiaron en el editor de texto plano Sublime Text .

Una vez se tenían los textos en este editor y de forma digitalizada, se procedió a la elaboración un documento XML con el texto en español, ya que sería aquel en el que se realizarían las búsquedas pertinentes. Se utilizó la codificación UTF-8, capaz de representar cualquier carácter Unicode.

Como se ha comentado previamente, el proyecto cuenta con dos funcionalidades principales en lo que se refiere a búsquedas: un buscador de nombres de personajes y un buscador de topónimos en la obra. Es por esto que se procedió a realizar un etiquetado del XML siguiendo la codificación de la Text Encoding Initiative (TEI) de los nombres de personajes y de lugares. Para ello, se utilizaron las etiquetas persName y placeName junto al atributo @corresp que posteriormente enlazaría estos elementos con los documentos HTML.

<iframe id="textoxml" style="display: none;">

<TEI>

<teiHeader>

<fileDesc>

<titleStmt>

<title>Elegías de Teognis

<ptr target="https://es.wikipedia.org/wiki/Teognis/" />

</title>

<author>Patricia Garrido Teixeira </author>

<respStmt>

<resp>Tutelado por: </resp>

<name>Adrián Menéndez de la Cuesta</name>

<name>Juan Antonio Álvarez-Pedrosa</name>

</respStmt>

</titleStmt>

<publicationStmt>

<p>Trabajo de Fin de Máster de Patricia Garrido.</p>

</publicationStmt>

<sourceDesc>

<p>Elegías de Teognis</p>

</sourceDesc>

</fileDesc>

</teiHeader>

<text>

<body>

<div type="elegías">

<div n="1">

<lg n="1">

<l>Oh soberano, hijo de

<persName corresp="#Leto">Leto</persName>, vástago de

<persName corresp="#Zeus">Zeus</persName>,</l>

<l>de ti nunca me olvidaré al comenzar ni al concluir, sino que siempre el primero,</l>

<l>el último y en medio te cantaré. Mas tú escúchame y concédeme</l>

<l>tu favor. Soberano

<persName corresp="#Apolo">Febo</persName>, cuando te dio a luz la augusta</l>

<l>diosa Leto, sujeta a una palmera con gráciles manos, a ti, el</l>

<l>más bello de los inmortales, junto a una laguna redondeada, la</l>

<l>sagrada

<placeName corresp="#Delos" href="https://www.google.com/maps/d/embed?mid=1RGVkcjs8McbToptYUqY\_cY-VE48">Delos</placeName> se llenó toda de una divina fragancia, y la enorme</l>

<l>tierra sonrió y se gozó el profundo abismo de la mar canosa.</l>

</lg>

Fragmento de código del documento XML principal con el texto en español

Una vez realizado este documento XML con el texto completo en español, se tuvieron que realizar dos documentos más en XML: uno destinado a los nombres de personajes que aparecen en el texto y otro, a los lugares. Por y para ello, se realizó, en primer lugar, un análisis de los diferentes personajes que aparecen en la obra, elaborando así una lista por cada personaje con su nombre en griego, en español y una pequeña descripción del mismo. Este documento, que se estructura de nuevo siguiendo el estándar de la TEI, se conforma de un elemento listPerson con un elemento person para cada personaje, identificado por un #id que después enlazaría con los documentos HTML correspondientes. Dentro de cada elemento person, hay un elemento persName type=”gr” y type=”es” con el nombre del personaje tanto en griego como en español, así como una etiqueta description, para incluir una breve descripción del personaje. Por último, nos encontramos con una etiqueta url donde se recoge una dirección por cada personaje, que referencia un documento en PDF en GoogleDrive con licencia pública con la información recogida del personaje.

<listPerson>

<person id="Eeton">

<persName type="gr">Αἴθων </persName>

<persName type="es">Eetón</persName>

<description>Generalmente se interpreta como nombre de un pueblo histórico o mitológico; también es el nombre que Odiseo se da

a sí mismo en su conversación con Penélope.</description>

<url>https://drive.google.com/file/d/0BzaPEnTHPE2QWFU0U0FhSUY4ZHc/view?usp=sharing</url>

</person>

<person id="Academo">

<persName type="gr">Ἀκάδημος</persName>

<persName type="es">Academo</persName>

<description>Personaje desconocido</description>

<url>https://drive.google.com/file/d/0BzaPEnTHPE2QbEpWdnkwbllIemM/view?usp=sharing</url>

</person>

Fragmento de código del documento XML destinado a los nombres de personajes

Asimismo, para el XML de los nombres de lugares, se realizó una pequeña investigación de los topónimos de la obra, de tal manera que se determinó dónde se encontrarían estos lugares actualmente estableciendo sus coordenadas correspondientes. Se utilizó también el framework de GoogleMaps para la creación de mapas personalizados. Se originó un mapa que contiene todos los lugares aparentes en la obra, así como un mapa individual por cada uno de los lugares de la misma.

Este documento XML se conforma por un elemento raíz listPlace bajo el que se encuentra cada uno de los lugares con la etiqueta place con un identificador para enlazar el documento con los documentos HTML correspondientes; así como con un type como atributo que determina la categoría del lugar atendiendo a su geología. Dentro de la etique place por cada uno de los lugares se encuentra el elemento location, así como un placeName type=”gr” y type=”es”. Además, el elemento geo recoge las coordenadas de la ubicación actual del lugar y el elemento url contiene la dirección del mapa individual por cada lugar creado anteriormente.

<listPlace>

<place xml:id="Delos" type="isla">

<location>

<placeName type="gr">Δῆλος</placeName>

<placeName type="es">Delos</placeName>

<geo>37.398476, 25.268918</geo>

<url>https://www.google.com/maps/d/edit?mid=1RGVkcjs8McbToptYUqY\_cY-VE48&amp;ll=37.39847598254924%2C25.268917999999985&amp;z=18</url>

</location>

</place>

<place xml:id="Troya" type="ciudad">

<location>

<placeName type="gr">Τροία</placeName>

<placeName type="es">Troya</placeName>

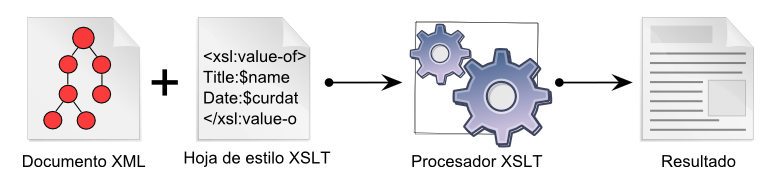
<geo> 39.9575, 26.238889</geo>

</location>

</place>

Fragmento de código del XML de los lugares de la obra

Posteriormente a la elaboración de los documentos XML se dio paso a la creación de los documentos en HTML para los textos en griego y en castellano. Con el fin de reducir la carga y tiempo de trabajo en realizar esto manualmente, se hizo uso de una hoja de transformaciones XSLT, un lenguaje de programación declarativo que permite generar documentos a partir de documentos XML de la siguiente manera:



Fuente de la imagen: <http://www.mclibre.org/consultar/xml/lecciones/xml_xslt.html>

Para lograr el objetivo propuesto, se utilizaron dos hojas XSLT diferentes: una dedicada a la estructura del documento HTML y otra a la numeración de cada uno de los versos del corpus teognídeo.

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:xs="http://www.w3.org/2001/XMLSchema"

xpath-default-namespace="http://www.tei-c.org/ns/1.0" exclude-result-prefixes="#all"

version="2.0">

<xsl:output method="html" doctype-system="about:legacy-compat"

omit-xml-declaration="yes" />

<xsl:template match="@\*|node()">

<xsl:copy>

<xsl:apply-templates select="@\*|node()"/>

</xsl:copy>

</xsl:template>

<xsl:template match="lg/@n">

<xsl:attribute name="n">

<xsl:value-of select="count(preceding::lg) + 1"/>

</xsl:attribute>

</xsl:template>

<xsl:template match="/">

<html>

<head>

<title>Elegías de Teognis</title>

<meta charset="utf-8" lang="es" />

<link rel="stylesheet" href="../css/bootstrap.css" />

<link rel="shortcut icon" type="png" href=""/>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<script src="../js/bootstrap.min.js"></script>

<script>

$(function () {

$('[data-toggle="tooltip"]').tooltip()

})

</script>

<script>

$(function () {

$('[data-toggle="popover"]').popover()

})

</script>

</head>

Fragmento de la hoja XSLT para la estructuración del documento HTML

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:xs="http://www.w3.org/2001/XMLSchema"

xpath-default-namespace="http://www.tei-c.org/ns/1.0" exclude-result-prefixes="#all"

version="2.0">

<xsl:output method="xml"/>

<xsl:template match="@\*|node()">

<xsl:copy>

<xsl:apply-templates select="@\*|node()"/>

</xsl:copy>

</xsl:template>

<xsl:template match="lg/@n">

<xsl:attribute name="n">

<xsl:value-of select="count(preceding::lg) + 1"/>

</xsl:attribute>

</xsl:template>

</xsl:stylesheet>

Código de la hoja XSLT para la numeración de los versos

Estas hojas de transformación dieron como resultado dos documentos HTML: uno destinado al texto en griego y otro destinado al texto en español. Debido a la falta de conocimientos en este lenguaje de programación, las hojas de estilo sirvieron como base de estos documentos, pero se tuvo que modificar manualmente parte de estos HTML con el fin de adaptarlos a las necesidades establecidas.

En estos documentos se da un identificador a cada uno de los nombres de personajes y de lugares, con el fin de enlazar estos con los documentos xml correspondientes a través del atributo @corresp

IMPORTACIÓ DEL MARCO DE BOOTSTRAP

En cada documento HTML se creó un buscador para realizar las búsquedas de nombres de personajes en ambos textos: griego y español. Estos buscadores se hicieron de forma manual utilizando HTML, así como el lenguaje de programación JavaScript y XPath, lenguaje que permite construir expresiones que recorren y procesan un documento XML; esto es: buscar en él.

Este paso fue uno de los más problemáticos en el desarrollo del trabajo, ya que la idea inicial para el funcionamiento de los buscadores no salió adelante y se optó por otra solución. En un primer lugar, se quería almacenar, por un lado, los documentos xml y por otro, los documentos HTML. El funcionamiento del buscador consistiría en los siguientes pasos:

El usuario selecciona el nombre del personaje que desea buscar (entrada)

A través de un código en javascript y xpath se recoge la información introducida por el usuario.

Se busca la información del personaje seleccionado en el documento XML correspondiente (xml de los nombres de los personajes)

Una vez recogidos los datos necesarios, se pinta a través de una función en Javascript una tabla con estos mismos en la página web que está viendo el usuario.

Sin embargo y como se ha mencionado anteriormente, este método no funcionó. Las búsquedas no se realizaban con normalidad ya que se repetía constantemente un mismo problema: aquello que el usuario introducía en el buscador (entrada) no podía enlazar directamente con el documento XML correspondiente ya que, como informaban los distintos navegadores al intentarlo, parecía violarse la jerarquía de permisos, lo que intentaba romper la seguridad de los mismos.

Tanto para el texto en español como en griego se utilizó el siguiente código en el cuerpo del documento HTML, cambiando simplemente la lengua de los valores.

<p> Buscador de nombres del texto </p>

<div id="custom-search-input">

<div class="input-group col-md-3">

<select style="width:220px;border:1px solid #04467E;background-color:#0000; color:#2D4167;font-size:14px" id="buscapalabra"

placeholder="buscar">

<option value="Academo"> Academo</option>

<option value="Agamenon"> Agamenón</option>

<option value="Alcatoo">Alcatoo</option>

<option value="Apolo">Apolo</option>

<option value="Arguiris"> Arguiris</option>

<option value="Artemis"> Ártemis</option>

<option value="Asclepiadas"> Asclepíadas</option>

<option value="Boreas">Bóreas</option

(…)

</select>

<span class="input-group-btn">

<button class="btn btn-info btn-xs" onclick="buscador()">

<i class="glyphicon glyphicon-search"></i>

</button>

</span>

</div>

</div>

Código del *body* de los documentos HTML para el buscador de personajes

En la cabecera de los documentos HTML se incluyó la siguiente función para el correcto funcionamiento del buscador:

<script type="text/javascript">

function buscador() {

var textoxml = document.getElementById('textoxml').textContent;

var nombresxml = document.getElementById('nombresxml').textContent;

parser = new DOMParser();

xmlDoc1 = parser.parseFromString(textoxml, "text/xml");

xmlDoc2 = parser.parseFromString(nombresxml, "text/xml");

// Obtengo la palabra a buscar

var nom = document.getElementById('buscapalabra').value;

// Consultas xPath

st1 = "//persName[@corresp='#" + nom + "']";

st2 = "//person[@id='" + nom + "']//\*";

var result1 = xmlDoc1.evaluate(st1, xmlDoc1, null, XPathResult.ITERATOR, null);

var result2 = xmlDoc2.evaluate(st2, xmlDoc2, null, XPathResult.ITERATOR, null);

try {

// Se van a iterar los dos nodos

var node1 = result1.iterateNext();

var node2 = result2.iterateNext();

// En infoNode1 guardo la información que obtenga del textoxml

var infoNode1;

// En el array datos guardo la información que obtenga del nombresxml

var i = 0;

var datos = [];

while (node1) {

infoNode1 = node1.textContent.toString();

node1 = result1.iterateNext();

}

while (node2) {

// Almaceno en el array datos, los campos necesarios para mostrar la tabla

datos[i] = node2.textContent.toString();

node2 = result2.iterateNext();

i++;

}

// Creación de la tabla con la información

var loadTable = '<table width="1000" border="2" margin="100"><tr><td colspan="2" rowspan="1">' + "Información de" + " " + infoNode1 + '</td>';

for (var i = 0; i < datos.length - 1; i++) {

loadTable += '</tr><tr><td width="118">' + datos[i] + '</td>';

}

loadTable += '</tr></table>';

loadTable += "<a href='" + datos[datos.length - 1] + "'>Documento con la información</a>";

//Cargo la tabla

document.getElementById('elementoTabla').innerHTML = "";

$('#elementoTabla').append(loadTable.valueOf());

document.getElementById('elementoTabla').style.visibility = 'visible';

}

catch (e) {

alert('No se encuentra la palabra');

dump('Error: Document tree modified during iteration ' + e);

}

}

</script>

De esta forma, el buscador funciona de la siguiente manera:

1. El usuario selecciona el personaje sobre el cual desea obtener información
2. El usuario hace click en el icono de búsqueda, que llamará a la función buscador( )
3. A través de esta función, se coge el contenido de los documentos xml correspondientes (xml con el texto completo y xml de los personajes de la obra), los cuales se encuentran insertados de forma oculta en el documento HTML
4. Mediante el identificador del buscador en el cuerpo del html ‘buscapalabra’, se ha recogido la palabra que se desea buscar
5. En la función buscador( ) se obtiene la palabra a buscar recogiendo su valor.
6. A continuación se declaran y se ejecutan las consultas en XPath
7. Los datos recogidos por las consultas se guardan en variables a través de las distintas iteraciones en un array, con el fin de poder sacarlos por pantalla posteriormente.
8. Por último, se crea una tabla a través de javascript que va colocando cada uno de los campos recogidas en una celda. Esta información se pinta en un div escondido en el documento html cuyo código es el siguiente.

<div id="elementoTabla" style="visibility: hidden;">

</div>

1. Así, se conforma una tabla de cuatro que saca por pantalla la siguiente información:
   1. Sobre qué personaje es la información
   2. Nombre del personaje en griego
   3. Nombre del personaje en español
   4. Breve descripción del personaje

Tras el correcto funcionamiento de este tipo de buscador, se pasó a desarrollar la página destinada a los lugares de la obra. Para ello se elaboró manualmente un documento HTML, en el cual se incluyó un segundo buscador. En esta página obtenemos, en un principio, la vista de un mapa que recoge todos los lugares que aparecen en la obra, el cual se inserta a través de un iframe que recoge la url del mapa, publicado en la web de forma pública para cualquier usuario.

El funcionamiento de este es similar al anterior, ya que, de nuevo, el documento xml correspondiente (documento del texto completo) tuvo que ser incrustado en el documento HTML de forma oculta.

El buscador del cuerpo del documento en HTML funciona, como el anterior, con un select que permite seleccionar el nombre deseado por el usuario:

<p> Buscador de nombres antiguos de lugares en la obra </p>

<div class="input-group col-md-3 col-sm-12 col-xs-12">

<select id="buscalugar" placeholder="Buscar lugares">

<option value="vacío"> - - - </option>

<option value="Delos">Delos / Δήλος</option>

<option value="Troya"> Troya / Τροία</option>

<option value="Tebas"> Tebas / Θήβη</option>

<option value="Esmirna">Esmirna / Σμύρνα</option>

<option value="Colofon"> Colofón / Κολοφών</option>

<option value="Eurotas"> Eurotas / Εὐρώτας</option>

<option value="Lacedemonia"> Lacedemonia / Λακεδαίμων</option>

(…)

</select>

<span class="input-group-btn">

<button class="btn btn-info btn-xs" onclick="buscaLugares()">

<i class="glyphicon glyphicon-search"></i>

</button>

<a href="location.html">

<button class="btn btn-primary btn-xs" type="reset">

<i class="glyphicon glyphicon-repeat"></i>

</button></a>

</span>

</div>

En la cabecera del documento HTML se utiliza la siguiente función:

<script type="text/javascript">

function buscaLugares() {

var textoxml = document.getElementById('textoxml').textContent;

parser = new DOMParser();

xmlDoc1 = parser.parseFromString(textoxml, "text/xml");

var nom = document.getElementById('buscalugar').value;

st3 = "//placeName[@corresp='#" + nom + "']//@href";

var result3 = xmlDoc1.evaluate(st3, xmlDoc1, null, XPathResult.ITERATOR, null);

try {

var node3 = result3.iterateNext();

var infoNode3;

while (node3) {

infoNode3 = node3.textContent.toString();

node3 = result3.iterateNext();

}

// Se pone la url en el iframe para que se carge correctamente

$('#showPlace').prop('src', infoNode3.toString());

document.getElementById("showPlace").style.display = "block";

}

catch (e) {

alert('Elige una opción válida');

dump('Error: Document tree modified during iteration ' + e);

}

}

</script>

Así pues, el buscador de topónimos en la obra funciona de la siguiente manera:

1. El usuario selecciona el lugar que desea buscar
2. El usuario pincha en el icono de búsqueda y se llama a la función buscaLugares ( )
3. En esta función se busca en el contenido del documento XML necesario.
4. Cuando el usuario ha hecho la búsqueda se ha recogido el valor de la búsqueda en su identificador y se toma el valor introducido por el mismo
5. A continuación se declaran y se ejecutan las consultas en XPath
6. Los datos recogidos por las consultas se guardan en variables a través de las distintas iteraciones en un array, con el fin de poder sacarlos por pantalla posteriormente.
7. Por último, se saca en el iframe del mapa inicial, la url correspondiente al lugar seleccionado tras realizar la búsqueda en el documento XML, que se establecía a través del atributo href.
8. El usuario puede recargar el mapa inicial pulsando en el icono correspondiente que hará que se recargue la página inicial

A continuación, se fueron aumentando las funcionalidades y lo siguiente fue crear un enlace para cada uno de los términos del texto en griego, de tal manera que enlazaran directamente con su entrada correspondiente en el diccionario de léxico griego ΛΟΓΕΙΟΝ. Para ello, se creó un pequeño programa en Phyton que produjera de forma automática cada uno de estos enlaces en el texto en el documento HTML.

# -\*- coding: utf-8 -\*-

"""

Created on Mon Jun 19 11:19:39 2017

"""

def generar\_links\_plano():

f = open('texto.txt', 'r')

f2 = open('texto\_links.txt', 'w')

ls = f.readlines()

texto = ""

for linea in ls:

palabras = linea.split()

for pal in palabras:

if pal[-1] == ",":

pal = pal[:-1]

texto = texto + "<a href=\"http://logeion.uchicago.edu/index.html#" + pal + "\">" + pal + "</a>" + ", "

else:

texto = texto + "<a href=\"http://logeion.uchicago.edu/index.html#" + pal + "\">" + pal + "</a>" + " "

texto = texto + "\n"

f2.write(texto)

f.close()

f2.close()

def generar\_links\_htmlv2():

f = open('teognisgriego.html', 'r', encoding='utf-8')

f2 = open('teognisgriego\_links.html', 'w', encoding='utf-8')

ls = f.readlines()

for linea in ls:

if linea.find("<p>") != -1:

mi\_pal = ""

link = False

for letra in linea:

if letra == "<":

link = False

if mi\_pal != "":

mi\_pal = "<a href=\"http://logeion.uchicago.edu/index.html#" + mi\_pal + "\">" + mi\_pal + "</a>"

f2.write(mi\_pal)

mi\_pal = ""

f2.write(letra)

elif letra == " " or letra == "," or letra == "." or letra == "\n":

if mi\_pal != "":

mi\_pal = "<a href=\"http://logeion.uchicago.edu/index.html#" + mi\_pal + "\">" + mi\_pal + "</a>"

f2.write(mi\_pal)

mi\_pal = ""

f2.write(letra)

elif link:

mi\_pal += letra

elif letra == ">":

link = True

f2.write(letra)

else:

f2.write(letra)

else:

f2.write(linea)

f.close()

f2.close()

def generar\_links\_html():

f = open('teognisgriego.html', 'r')

f2 = open('texto\_links.txt', 'w')

ls = f.readlines()

for linea in ls:

mi\_pal = ""

link = False

for letra in linea:

if letra == "<":

link = False

if mi\_pal != "":

mi\_pal = "<a href=\"http://logeion.uchicago.edu/index.html#" + mi\_pal + "\">" + mi\_pal + "</a>"

f2.write(mi\_pal)

mi\_pal = ""

f2.write(letra)

elif letra == " " or letra == "," or letra == "." or letra == "\n":

if mi\_pal != "":

mi\_pal = "<a href=\"http://logeion.uchicago.edu/index.html#" + mi\_pal + "\">" + mi\_pal + "</a>"

f2.write(mi\_pal)

mi\_pal = ""

f2.write(letra)

elif link:

mi\_pal += letra

elif letra == ">":

link = True

f2.write(letra)

else:

f2.write(letra)

f.close()

f2.close()

Código en Python para la creación de links de forma automática

Este código fue desarrollado en el editor Anaconda, una distribución libre de código abierto para lenguajes de programación como Python o R para procesamiento de datos a gran escala.

A través de este código se abre el fichero de texto correspondiente y se inserta en cada una de las palabras un enlace. Esto se hace a través del elemento <a> que se abre y se cierra ante cada término y con el atributo <href> para incluir el enlace correspondiente.

Para la formación del enlace correspondiente, se realizó un estudio de cómo funcionaban los mismos en la página web del diccionario. Cuando se clicaba sobre un término en griego, el enlace resultante se conformaba de la dirección del sitio web (<http://logeion.uchicago.edu/index.html>) junto a /’la palabra’ seleccionada.

Es por esto que a través de Phyton se creaban los enlaces de la siguiente manera:

"<a href=\"http://logeion.uchicago.edu/index.html#" + mi\_pal + "\">" + mi\_pal + "</a>"

Una vez desarrollada esta funcionalidad se pasó al siguiente proceso, consistente en la creación de tooltips (o descripciones emergentes) para el texto de la obra en castellano, particularmente para los nombres de personajes.

El fin de adjuntar al texto esta funcionalidad, no fue más que la de ofrecer al usuario una descripción breve de cada personaje a medida que va leyendo el texto, sin la necesidad de tener que recurrir al buscador –en la parte superior de la página- para una mera contextualización del fragmento de su lectura. La información aquí proporcionada no es especialmente extensa, sino que contiene el texto necesario para situar al lector en el marco correcto.

Para la realización de esta funcionalidad se hizo uso del siguiente código: <a href="#" data-toggle="tooltip" title=""> en el documento HTML y del siguiente en javascript en la cabecera del mismo: $(document).ready(function () { $('[data-toggle="tooltip"]').tooltip();});

La primera línea de código se insertó directamente en el documento HTML para el texto en español en cada uno de los personajes. De esta manera, cada una de las palabras objeto de esta funcionalidad son un enlace que en lugar de redirigir al usuario a un sitio externo, crea un pequeño cuadro emergente sobre la palabra con la información que se introduce en el atributo tittle. Además, el puntero cambia cuando se sitúa sobre estas palabras, de tal manera que esto supone una ayuda al usuario, que cuando pase el puntero por encima, se percatará de que algo es diferente.

Con este procedimiento, se finalizó el desarrollo e implementación de las distintas funcionalidades de la página web y se dio pie, en consecuencia, a la maquetación y diseño de la página web; así como a la búsqueda y redacción de la contextualización de la obra y a la creación de una guía de navegación para el usuario.

Para la contextualización de la página web, se recurrió a la edición XXXX de XXXX y se optó por dividir la información en: vida y obra de Teognis de Megara; lengua; y métrica. La información aportada es breve, seria y concisa; lo cual permite al usuario situarse en un marco teórico y lingüístico sin caer en la pesadez de un texto demasiado extenso con poca información útil.

Asimismo, la guía para el usuario ofrece una explicación básica de navegación y de las funcionalidades de la página web, clasificada según las distintas páginas de las que consta este portal.

A continuación, se pasó a la inclusión en el sitio web de los documentos HTML con estas informaciones.

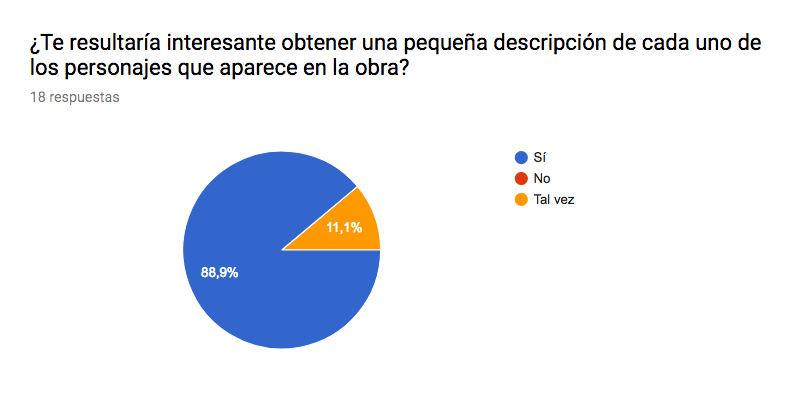
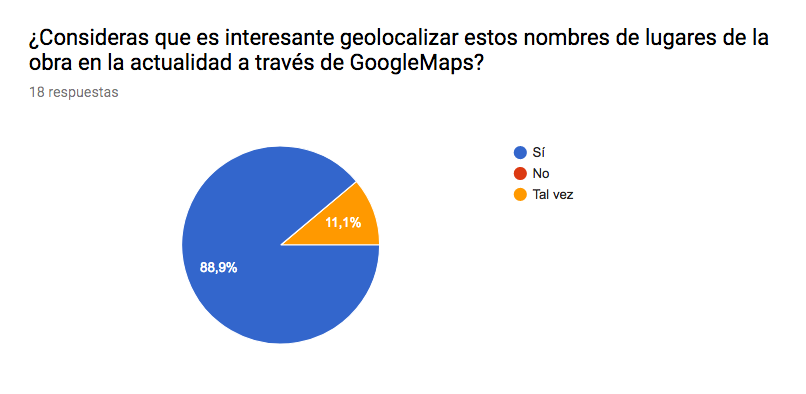
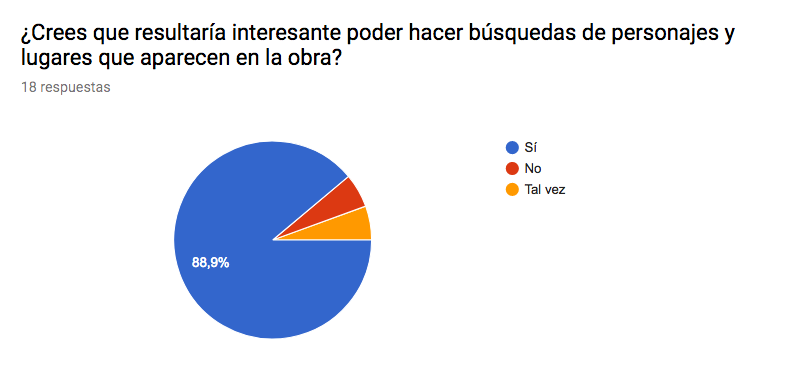
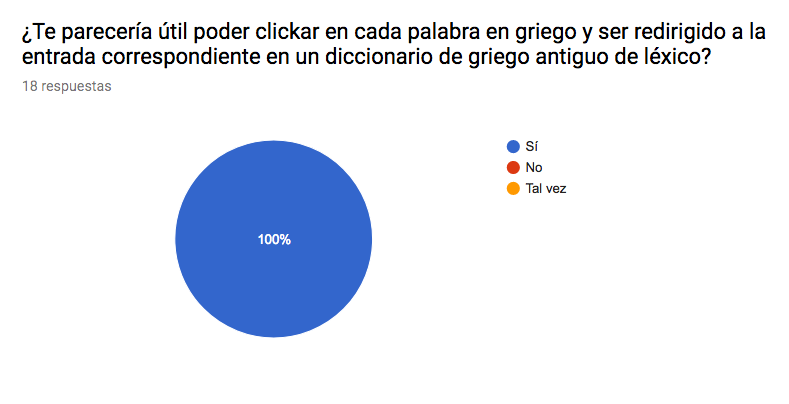
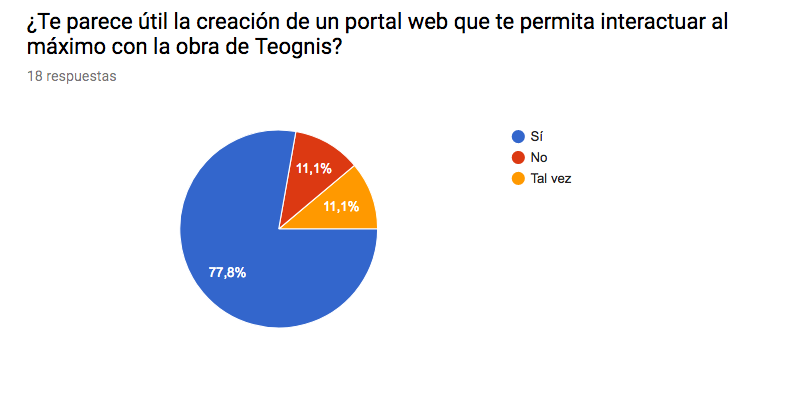
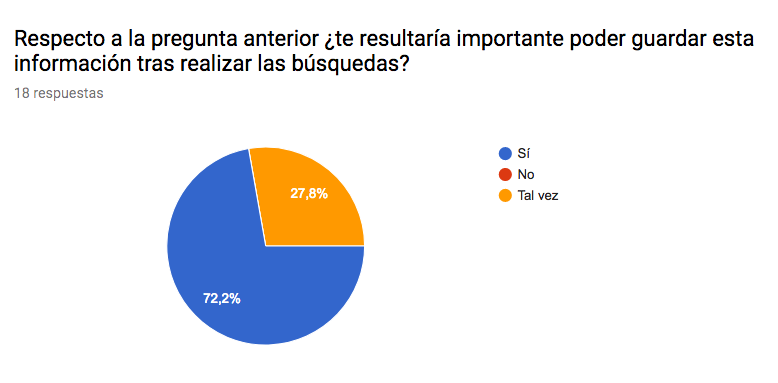
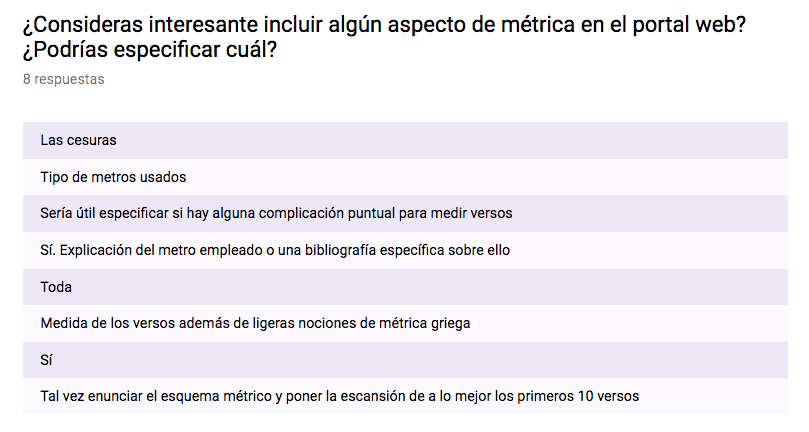
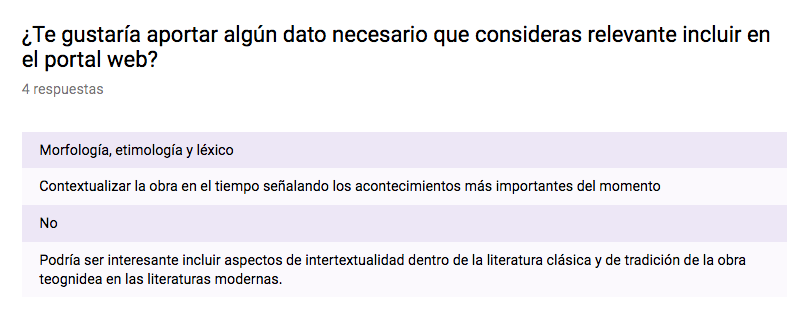
Por último, se dio paso a la maquetación del sitio web. Esto se hizo a través de la modificación de la propia css predefinida por Bootstrap, así como con la creación de una css personalizada fuera de esta carpeta, con distintos aspectos que no recogía la primera.

PRESENTACIÓN DE LOS RESULTADOS

CONCLUSIONES + LÍNEAS DE FUTURO

Conclusiones relacionadas con el máster en Letras Digitales

REFERENCIAS

* [**DH Poetry Modelling: a Quest for Philological and Technical Standardization**](https://zenodo.org/record/60762)  
  *Rio Riande, Gimena del; Martínez Cantón, Clara Isabel; González-Blanco García, Elena*  
  *Jagiellonian University & Pedagogical University (Cracovia) 2016*  
  *doi:10.5281/zenodo.60765, doi:10.5281/zenodo.60762*
* [**Postdata: Towards publishing European Poetry as Linked Open Data**](http://e-spacio.uned.es/fez/view/bibliuned:363-Egonzalez9)  
  *Curado Malta, Mariana; Centenera, Paloma; González-Blanco García, Elena2016*
* [**POSTDATA – Towards publishing European poetry as linked open data**](http://hdl.handle.net/10400.22/8564)  
  *Curado Malta, Mariana; Gonzalez-Blanco, Elena*  
  *DCMI 2016*
* [**Un nuevo camino hacia las humanidades digitales: el Laboratorio de Innovación en Humanidades Digitales de la UNED (LINHD)**](http://e-spacio.uned.es/fez/view/bibliuned:352-Egonzalez21)  
  *González-Blanco García, Elena*  
  *Universidad Nacional de Educación a Distancia (España). Centro de Investigación de Semiótica Literaria, Teatral y Nuevas Tecnologías Signa: Revista de la Asociación Española de Semiótica, 25, 79-93 2016*
* [**Métrica, clasificación y tecnología en los repertorios métricos**](http://e-spacio.uned.es/fez/view/bibliuned:signa-2016-25-7050)  
  *Rodríguez Gómez, José Luis*  
  *Universidad Nacional de Educación a Distancia (España). Centro de Investigación de Semiótica Literaria, Teatral y Nuevas Tecnologías Signa: Revista de la Asociación Española de Semiótica, 25, 137-167 2016*
* [**TEIScribe: A graphic tool for composing and testing TEI documents in the context of the EVI-LINHD environment**](http://e-spacio.uned.es/fez/view/bibliuned:363-Egonzalez10)  
  *González-Blanco García, Elena2016*
* [**La codificación informática del sistema poético medieval castell ano, problemas y propuestas en la elaboración de un repertorio métrico digital: ReMetCa**](http://hdl.handle.net/10171/35718)  
  *González-Blanco-García, E. (Elena); Martínez-Cantón, C. I. (Clara Isabel); Martos-Pérez, M. D. (María Dolores); Río-Riande, M.G. (María Gimena) del*  
  *Servicio de Publicaciones de la Universidad de Navarra 2014*
* [**EVILINHD, a Virtual Research Environment open and collaborative for DH Scholars**](https://zenodo.org/record/574101)  
  *Elena González-Blanco2017*  
  *doi:10.5281/zenodo.574101*
* [**Researchers’ perceptions of DH trends and topics in the English and Spanish-speaking community. DayofDH data as a case study**](http://e-spacio.uned.es/fez/view/bibliuned:363-Egonzalez7)  
  *Robles-Gómez, Antonio; González-Blanco García, Elena; Ros, Salvador; Rio Riande, Gimena del; Hernández, Roberto; Tobarra, Llanos; Caminero, Agustín; Pastor, Rafael*  
  *Jagiellonian University & Pedagogical University (Cracovia) 2016*
* [**From syllables, lines and stanzas to linked open data: standardization, interoperability and multilingual challenges for digital humanities**](http://e-spacio.uned.es/fez/view/bibliuned:363-Egonzalez11)  
  *González-Blanco García, Elena; Manailescu, Mara; Ros, Salvador2016*
* [**La edición digital de textos literarios: planteamientos y perspectivas de futuro**](http://e-spacio.uned.es/fez/view/bibliuned:352-Egonzalez22)  
  *González-Blanco García, Elena*  
  *Universidad de Navarra RILCE: Revista de filología hispánica, 33 (1), 239-258 2017*
* [**Spanish CLARIN K-Centre**](http://hdl.handle.net/10045/57765)  
  *Bel, Núria; González-Blanco García, Elena; Iruskieta, Mikel*  
  *Sociedad Española para el Procesamiento del Lenguaje Natural 2016*
* [**Boundary Land: Diversity as a defining feature of the Digital Humanities**](http://e-spacio.uned.es/fez/view/bibliuned:363-Egonzalez5)  
  *O'Donnell, Daniel Paul; Bordalejo, Barbara; Murray Ray, Padmini; Rio Riande, Gimena del; González-Blanco García, Elena*  
  *Jagiellonian University & Pedagogical University (Cracovia) 2016*
* [**EVI-LINHD. A Virtual Research Environment for the Spanish-speaking Community**](http://e-spacio.uned.es/fez/view/bibliuned:363-Egonzalez6)  
  *Rio Riande, Gimena del; González-Blanco García, Elena; Martínez Cantón, Clara Isabel; Escribano, J*  
  *Jagiellonian University & Pedagogical University (Cracovia) 2016*
* [**Linked open data to represent multilingual poetry collections. A proposal to solve interoperability issues between poetic repertoires**](https://zenodo.org/record/60767)  
  *González-Blanco, Elena; Del Rio Riande, Gimena; Martinez Cantón,Clara2016*  
  *doi:10.5281/zenodo.60767*
* [**Digital repertoires of poetry metrics: towards a Linked Open Data ecosystem**](http://e-spacio.uned.es/fez/view/bibliuned:363-Egonzalez12)  
  *Curado Malta, Mariana; González-Blanco García, Elena; Martínez Cantón, Clara Isabel; Rio Riande, Gimena del2016*
* [**TEIScribe: A graphic tool for composing and testing TEI documents in the context of the EVI-LINHD environment**](https://zenodo.org/record/161379)  
  *González Blanco García, Elena; Robles-Gómez, Antonio; Ros, Salvador; Del Rio Riande, Gimena; Hernández, Roberto; Urizar, Miguel; Martínez Cantón, Clara; Pastor, Rafael; Cano, Jesús; Caminero, Agustín*  
  *Austrian Centre for Digital Humanities 2016*  
  *doi:10.5281/zenodo.161379*
* [**Analyzing poetry databases to develop a metadata application profile. Why each language uses a different way of modelling?**](https://zenodo.org/record/573767)  
  *Patricia Garrido2017*  
  *doi:10.5281/zenodo.573767*
* [**Linked open data to represent multilingual poetry collections. A proposal to solve interoperability issues between poetic repertoires**](https://zenodo.org/record/60768)  
  *González-Blanco, Elena; Del Rio Riande, Gimena; Martinez Cantón,Clara2016*  
  *doi:10.5281/zenodo.60768*
* [**Domain Model V0.1 of the metadata application profile for European Poetry**](https://zenodo.org/record/437827)  
  *Curado Malta, Mariana; Bermudez Sabel, Helena; Centenera, Paloma2017*  
  *doi:10.5281/zenodo.437827*
* 
* 
* 
* 
* 
* 
* 
* Figura 7. Encuesta AN
* 
* Figura 8. Encuesta AN
* 
* 

ANEXOS

Ver anexo (<https://zenodo.org/record/573767#.WYA0B9Pyi3U>)

Fuente imágenes:

<https://2.bp.blogspot.com/-iTzaFgaxnRY/V6YfdbaTb5I/AAAAAAAAMKA/vJDzdnf5cEc1UNaU3lUscXu_BUcSDIXhQCLcB/s640/Vida.png>

<http://concepto.de/wp-content/uploads/2014/10/que-es-lenguaje-e1414454982880.jpg>

<http://serveiestacio.com/wp-content/uploads/2017/05/cinta.jpg>