Trabajo Fin de Máster

ARQUEOGRIEGOS

Francisco José Paños Merino

Máster en Ingeniería Informática
Universidad Nacional de Educación a Distancia

Tutor: Antonio Robles Gómez
Cotutora: Ana García Serrano

Octubre 2023

Resumen

Existen trabajos de humanistas que son realizados a título personal, es decir no están encuadrados dentro de algún proyecto profesional o académico financiado, hecho este que no implica necesariamente que sean de menor calidad o que sus contenidos no estén contrastados y puedan tener valor añadido suficiente como para obtener cierto prestigio.

El objetivo de este proyecto es facilitar una aproximación inicial básica y general, centrada fundamentalmente en la información proveniente de textos, para la creación de un entorno informático que provea las herramientas y aplicaciones necesarias para el almacenamiento e indexación de la información recopilada por un humanista y posteriormente habilitar mecanismos para la recuperación de la información mediante consultas que sean eficientes, que produzcan resultados claros y concisos a las preguntas que pretendan resolver esas consultas y que permitan su acceso y visualización.

El caso estudiado en este trabajo abarca información sobre numerosos yacimientos arqueológicos de la Antigua Grecia. Uno de los requisitos es que el sistema informático será accesible tanto para personal del mundo académico, que frecuentemente busca información más detallada, pero también para otro tipo de usuario como pueden ser turistas que busquen información práctica para preparar sus visitas a esos yacimientos.

Para dar cumplimiento a estos requerimientos se valoran varias opciones de tecnologías relacionadas con la gestión y visualización de información y se concluye que para explotar un corpus multimedia digital las bases de datos no son la mejor tecnología de almacenamiento para alojar cadenas de texto tan extensas como en corpus con muchos documentos textuales y, en cambio, las tecnologías como SOLR o bases de datos NoSQL son mucho más apropiadas siendo la eficiencia mejor en términos de velocidad de búsqueda.

En esta memoria se presenta un estado del arte tecnológico, se discutirán las alternativas de diseño e implementación y se muestra el caso de uso desarrollado, denominado Arqueogriegos[1]. El acceso vía web y el software desarrollado es de libre acceso (se ha de solicitar el acceso vía email).

[1] Ana García Serrano, Francisco Paños Merino y César Fernández, “Una mirada digital a los datos: el trabajo de un estudioso de la antigua Grecia” Presentación en el congreso HDH 2023 “Encuentros y transformaciones” Logroño, 18-20 octubre 2023.

Palabras clave: Humanidades Digitales, Apache SOLR, Bases de Datos NoSQL, Motor de Búsqueda, Recuperación de la Información, yacimientos arqueológicos, Antigua Grecia.

Abstract

There are works by humanists that are carried out in a personal capacity, i.e., they are not part of a funded professional or academic project, a fact that does not necessarily imply that they are of lesser quality or that their contents are not contrasted and may have sufficient added value to obtain a certain prestige.

The aim of this project is to provide an initial basic and general approach, centred fundamentally on information from texts, for the creation of a computer environment that provides the necessary tools and applications for the storage and indexing of the information compiled by a humanist and subsequently to enable mechanisms for the retrieval of information by means of efficient queries that produce clear and concise results to the questions that these queries are intended to resolve and that allow access and visualization.

The case studied in this paper covers information on numerous archaeological sites in Ancient Greece. One of the requirements is that the computer system will be accessible both to academics, who often seek more detailed information, but also to other types of users such as tourists seeking practical information to prepare their visits to these sites.

In order to fulfil these requirements, several options of technologies related to information management and visualization are assessed and it is concluded that to exploit a digital multimedia corpus, databases are not the best storage technology to host such large text strings as in corpora with many textual documents and, instead, technologies such as SOLR or NoSQL databases are much more appropriate and the efficiency is better in terms of search speed.

This report presents a technological state of the art, discusses the design and implementation alternatives and shows the use case developed, called Arqueogriegos. The access via web and the developed software is freely available (access has to be requested via email).

Keywords: Digital Humanities, Apache SOLR, NoSQL Databases, Search Engine, Information Retrieval, archaeological sites, Ancient Greece.

Índice General

Contenido

Índice de Figuras

Índice de Tablas

Índice de Diagramas

Acrónimos

ACRONIMO	SIGNIFICADO
Acrónimo	Significado
AAT	Art & Architecture Thesaurus
AEDT	Análisis Estadístico de Datos Textuales
AMP	Apache MySQL PHP
API	Interfaz de Programación de Aplicaciones
AVIP	Plataforma Audiovisual sobre Tecnología IP
CAD	Computer Aided Design
CMD	Command
CMS	Content Management System
CQL	Cassandra Query Language
CRUD	Create Read Update Delete
CSV	Comma Separated Values
CURL	CURL URL Request Library
DIMH	Proyecto El Dibujante Ingeniero Al Servicio De La Monarquía Hispánica
DOC	Extensión de Archivo Microsoft Word Document
DWG	Extensión de Archivo acrónmo de Drawing
ECTS	European Credit Transfer System
FCA	Algoritmo Formal Concept Analysis
GNU	GNU's Not UNIX
GPL3	The GNU General Public License Version 3
HDH	Humanidades Digitales Hispánicas
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IA	Inteligencia Artificial
JPG	Extensión de Archivo Joint Photographic Experts Group
JSON	JavaScript Object Notatio
LINUX	Linus UNIX
MySQL	My Structured Query Language
NoSQL	Not Only SQL
OWL	Web Ontology Language
P2P	Peer to Peer
PDF	Extensión de Archivo Portable Document Format
PHP	Hypert Text-Preprocessor
PLN	Procesamiento del Lenguaje Natural
PNG	Extensión de Archivo Portable Network Graphic
RDF	Resource Description Framework
REST	Representational State Transfer
SIG	Sistema de Información Geográfica

SOLR	Searching On Lucene Replication
SQL	Structured Query Language
UNED	Universidad Nacional de Educación a Distancia
URL	Uniform Resource Locator
XHTML	eXtensible Hyper Text Markup Language
XLS	Extensión de Archivo Microsoft Excel Hoja de Cálculo
XML	Extensión de Archivo eXtensible Markup Language

Capítulo 1: Descripción del Proyecto

En este trabajo se presenta con detalle la metodología, las decisiones y el diseño realizado a partir del trabajo de un estudioso humanista que ha construido un novedoso catálogo de museos y yacimientos de la Antigua Grecia. El marco del trabajo son las Humanidades Digitales en las que se debe concluir sobre la tecnología actual más adecuada para los objetivos planteados por un humanista.

La tecnología básica actual permite la digitalización de trabajos humanistas, utilizando editores de texto, gestores de imágenes u hojas de cálculo para organizar la variada documentación digital generada en sus estudios e investigaciones. Toda esta documentación forma un corpus multimedia. Es la tarea del humanista digital encontrar la mejor solución de carácter informático para explotar todo el conocimiento vertido en el corpus multimedia. Las preguntas que deben responderse son ¿Cuál es la estructura de información y la organización de ficheros del corpus? ¿Qué tipo de visualización es útil para los potenciales usuarios de la información? ¿Cómo se quiere navegar por el corpus multimedia? ¿Qué tipo de búsquedas de información pretenden los expertos de dominio o estudiosos que les sean respondidas?

1.1 Objetivos y detalles

En esta sección se relacionarán los objetivos y un breve detalle de los cinco capítulos que conforman esta memoria.

CAPÍTULO 1 – DESCRIPCIÓN DEL PROYECTO

Objetivos

- Contextualizar la propuesta a desarrollar dentro del área de conocimiento correspondiente, en este caso las Humanidades Digitales
- Indicar las herramientas potenciales para desarrollar la propuesta
- Describir los elementos que conformarán el Corpus
- Planificar los tiempos necesarios para el diseño y la implementación del sistema informático que proveerá la solución de este trabajo
- Presupuestar los costes asociados a ese sistema informático

Detalles

En esta sección se realiza una introducción de las necesidades que pueden tener humanistas que recopilan una cantidad de información considerable en distintos formatos digitales y que pueden sacar beneficio de las tecnologías más actuales para organizar y distribuir esa información de la manera más eficiente posible

Se describe de manera muy breve la composición del Corpus principal de un caso en concreto que es el objeto de este trabajo.

Se detallan una planificación temporal del proyecto y una solución concreta en términos de equipos informáticos y/o servicios que proveerán la plataforma tecnológica base para albergar el caso de uso propuesto en esta memoria, informando de los costes económicos asociados a la creación de esa plataforma.

CAPÍTULO 2 – ESTADO DEL ARTE

Objetivos

- Revisar el estado del arte en Humanidades Digitales
- Revisar técnicas cualitativas y cuantitativas
- Revisar aplicaciones existentes de interés para este área
- Analizar proyectos en humanidades digitales que pudieran tener similitudes con las necesarias de nuestro caso particular
- Analizar tecnologías de almacenamiento y búsqueda
- Revisar aplicación de análisis de datos

Detalles

Para encontrar las tecnologías que mejor puedan resolver las necesidades que plantea el humanista de este proyecto lo primero es revisar el estado del arte en Humanidades Digitales. Esta revisión nos pondrá en la pista tanto de aplicaciones informáticas usadas para proyectos similares como en proyectos ya creados que usan algunas de esas aplicaciones.

Como aplicaciones se hace mención especial a los sistemas de almacenamiento y búsqueda (bases de datos SQL, NoSQL como Cassandra, MongoDB y Lucene-SOLR, almacenamiento basado en ontologías), así como a otras aplicaciones (Voyant Tools, Omeka, Spacy…) que han demostrado ser muy adecuadas para proyectos similares. Existen muchas más de las que se indican en este apartado, pero en la revisión se han analizado, en mayor o menor grado, solo aquellas a las que se les intuía cierta utilidad para este proyecto.

Como proyectos desarrollados se ha analizado con mucho detalles principalmente estos dos:

- DIMH [1] y [2]

- MUSIVARIA [3]

CAPÍTULO 3 – DISEÑO

Objetivos

- Crear Corpus
- Crear Base de Datos SQL con MySQL
- Crear varias Bases de Datos (Colecciones) NoSQL con Apache SOLR
- Insertar información en Bases de Datos y Colecciones

Detalles

En este capítulo se describe el acceso a un sistema de base de datos MySQL, la creación de la estructura de una base de datos MySQL, así como la creación de la base de datos ARQUEOGRIEGOS_SQL.

También se detalla la instalación del motor de búsqueda y almacenamiento Apache SOLR, se detallan sus modos de funcionamiento (SOLR-CLOUD y USER-MANAGED) y los métodos de inserción de la información y las colecciones (bases de datos) que se crearán para cada uno de esos métodos:

- Método de inserción mediante Apache TIKA -> Colección ARQUEOGRIEGOS_DOC
- Método de inserción mediante descubrimiento -> Colección ARQUEOGRIEGOS_XML
- Método de inserción mediante esquema predefinido -> Colección ARQUEOGRIEGOS_ESQ

CAPÍTULO 4 – PROTOTIPO DESARROLLADO Y PRUEBAS

Objetivos

- Habilitar servidor proveedor de servicio para Apache SOLR
- Habilitar servidor de hosting web para Sistema de Administración de Contenido WORDPRESS
- Habilitar la conexión entre ambos servidores
- Consultar la base de datos SQL y ver rendimientos
- Consultar las colecciones en Apache SOLR y ver rendimientos

- Comparar resultados de las búsquedas

Detalles

En primer lugar, es necesario habilitar los servidores para dar acceso a las personas que tendrán acceso al sistema informático que se implementa en este proyecto. Este sistema tendrá dos servidores principalmente si bien uno de ellos no es estrictamente necesario, estos dos servidores son:

- Servidor Apache SOLR, necesario para almacenar la información y proveer el servicio de búsquedas y consultas
- Servidor Web WORDPRESS, no estrictamente necesario, provee un servicio para usuarios finales presentando la información de una manera más intuitiva.

A continuación, como pruebas del sistema se buscará la respuesta a dos preguntas propuestas por el humanista, estas respuestas se tendrán que encontrar haciendo las búsquedas y consultas a las siguientes colecciones y bases de datos:

- Mediante VOYANT TOOLS sobre los archivos .XML generados previamente
- Consulta a la base de datos relacional ARQUEOGRIEGOS_SQL
- Búsqueda a la Colección ARQUEOGRIEGOS_ESQ
- Búsqueda al sistema conectado SOLR – WORDPRESS

Se analizan los tiempos obtenidos para las consultas y búsquedas y se estiman de manera básica cómo evolucionan los tiempos con la incorporación de más información a las bases de datos y a las colecciones.

Las pruebas se han adaptado para presentar este proyecto como presentación larga en el VI Congreso de la HDH que se celebrara en Logroño del 18 al 20 de octubre de 2023.

CAPÍTULO 5 – CONCLUSIONES Y TRABAJOS FUTUROS

Objetivos

- Obtener las conclusiones a todo lo realizado en este proyecto
- Introducir las mejoras posibles a realizar en trabajos posteriores que amplíen las funcionalidades y prestaciones del sistema desarrollado
- Plantear trabajos futuros a realizar para conseguir un sistema que abarque todas las necesidades especialmente para el tratamiento de más información multimedia (imágenes, vídeos, archivos CAD…)

Detalles

Se resumen los conclusiones obtenidas una vez implementado el sistema informático propuesto para atender las necesidades planteadas por un humanista, indicando las opciones que se han valorado para finalmente usar aquellas aplicaciones que se han considerados más eficientes para la consecución de los objetivos.

Como se indicará a lo largo de esta memoria, se ha intentado ajustar el diseño y la implementación del sistema al objeto de este proyecto, que forma para parte de un Trabajo Fin de Máster de 12 créditos ECTS lo que ha planteado muchas dudas de cuál debería ser la dimensión del proyecto, por lo que finalmente han quedado trabajos pendientes de explorar y a los que encontrar vías de desarrollo, como por ejemplo incorporar otros elementos multimedia al sistema (imágenes, vídeos, archivos CAD…) para su consulta, crear una aplicación que genere archivos .XML de manera automática, y en general otra serie de trabajos que mejorarían de manera importante a este sistema informático.

1.2 Contexto

Para encontrar el que se considera el primer proyecto atribuido a las Humanidades Digitales [4] nos tenemos que remontar al año 1949, y si bien se podrían nombrar algunos proyectos muy puntuales relacionados con las Humanidades Digitales, éstas no han ido progresando con la misma velocidad que otras áreas que sí han ido de la mano de la denominada revolución digital, y cuyo crecimiento ha sido mucho mayor.

La aparición de los ordenadores personales y más recientemente los smartphones han revolucionado la relación de las personas con la información en todos los sentidos y si bien las Humanidades Digitales no están tan consolidadas, podemos encontrar definiciones como la siguiente:

> *Las Humanidades Digitales pueden definirse como el espacio de convergencia entre ciencias de la computación, medio digital y disciplinas humanísticas en la búsqueda de nuevos modelos interpretativos y nuevos paradigmas de conocimiento acordes con las transformaciones operadas en el seno de la sociedad digital* [5]*, y el impacto directo de esta revolución digital en el campo de las Humanidades lo podemos ver reflejado a partir del año 2004 con el libro A Companion to Digital Humanities* [6] *y su posterior revisión de 2015 A New Companion to Digital Humanities* [7].

Como se ha indicado, en esta memoria se desarrolla un proceso completo de Humanidades Digitales para una necesidad concreta, un humanista que ha realizado personalmente un extenso y completo estudio para la construcción de un catálogo universal de museos y yacimientos arqueológicos de la antigua Grecia [8]. El estudioso que lo ha

elaborado ha planteado una serie de requisitos que deberán ser atendidos con la creación de un sistema informático, o aplicación o conjunto de herramientas, que de manera resumida son los siguientes:

- REQUISITO 1: Uso de la información para el mundo universitario pudiéndose usar los datos del estudio como "cantera" de materiales aptos para una función didáctica que incluyera una recuperación de la información ágil y eficiente.
- REQUISITO2: Uso de la información para el más prosaico mundo turístico pudiéndose usar los datos como guía de viaje para los visitantes de un yacimiento arqueológico o museo de los recogidos en el estudio.

Para estos dos requisitos de diseño del sistema informático a crear, se necesita conocer qué tipo peticiones de información o preguntas quiere realizar el humanista. Nos indicó que desea acceder a la información a través de las siguientes preguntas tipo:

- Pregunta 1: ¿Cuál es el conjunto de templos dedicados a una determinada divinidad del Panteón Olímpico (Zeus, Hera, Poseidón Apolo, Artemisa, Afrodita, Ares, Hermes, Deméter, Dioniso o Hefestos), que hay en una determinada región de las que está dividido el estudio (Ática, Atenas Tracia…)
- Pregunta 2: Igual que la anterior, pero con relación a los templos de tipo dórico o jónico, independientemente de la divinidad a la que esté dedicado.
- Pregunta 3: ¿Cuál es el conjunto de teatros que hay en una determinada región de las que está dividido el estudio?
- Pregunta 4: Conjunto de un determinado tipo de cerámica (imágenes) que hay en el total del estudio (Ánfora, Kylix, Oinochoe, Crátera, Aribalo, Pixide…).
- Pregunta 5: Estatuas (imágenes) de una determinada divinidad del Panteón Olímpico que hay en el total del estudio.
- Pregunta 6: Cerámica micénica que hay en los museos de una determinada región.
- Pregunta 7: Tumbas micénicas que hay en una determinada región.
- Pregunta 8: Conjunto de yacimientos relacionados con una determinada divinidad o personaje mitológico.

Por ello el objetivo general planteado a partir de ellas es la realización de un sistema de recuperación de información para la búsqueda de respuestas y acceso a los documentos que las contienen. Además, para el segundo requisito se podrá crear una aplicación, preferiblemente de tipo web para, que muestre la información recopilada por el estudioso de la manera más amigable para un usuario estándar.

Tras una revisión del estado del arte sobre las Humanidades Digitales, y con conocimiento profundo del autor sobre bases de datos y aplicaciones web, se discutirán una serie de herramientas y tecnologías para seleccionar las más oportunas para el diseño de la aplicación que permita la gestión integral de ese catálogo, haciendo especial énfasis en el diseño del modelo de almacenamiento de la información que provea el humanista. Estas herramientas y tecnologías que se usarán darán forma, denominado Proyecto ARQUEOGRIEGOS.

1.2 Descripción del catálogo: corpus

El catálogo universal del que se dispone para este proyecto versa sobre yacimientos arqueológicos de la antigua Grecia. Inicialmente, se incluirán en el corpus de prueba datos sobre dos regiones, Ática y Tracia, formando un sub-corpus de trabajo de forma que, una vez probada la viabilidad de la propuesta, se puedan incluir más regiones. Finalmente se harán unas pruebas con más regiones.

Antes de continuar, se define qué es un corpus según el diccionario de la Real Academia Española: *un corpus es un conjunto lo más extenso y ordenado posible de datos o textos científicos, literarios… que pueden servir de base a una investigación*.

Como objetos del corpus digital original se distinguirán principalmente cuatro:

- Documentos de texto en formato .doc
- Documentos de hoja de cálculo en formato .xls
- Archivos de imagen en formato .jpg
- Archivos de CAD en formato .dwg

En este trabajo fin de máster de 12 créditos ECTS, el proyecto inicial se centrará de manera principal en la gestión de la información textual del estudio, es decir de la información contenida en los archivos en formato de texto .DOC y en las hojas de cálculo en formato .XLS, si bien también para cumplir con el requisito 2 se hará una primera aproximación muy básica a la gestión de documentos de tipo grafico o JPG.

1.4 Planificación y Presupuesto

El sistema informático a desarrollar en este proyecto requerirá una serie de recursos como aplicaciones y dispositivos informáticos que cubrirán las necesidades para que el sistema pueda llevarse a cabo.

Los plazos temporales y los costes económicos de ejecución de este apartado se refieren a lo que se necesitaría tanto en tiempo como en dinero para la implementación de la solución que se ofrece en esta memoria. De este modo, los plazos de ejecución, de los tres escenarios que se plantearán, dados en semanas en esta memoria no se refieren al tiempo que ha llevado elaborar esta memoria y todo lo que ella conlleva como estudio del arte, adecuación de la información entregada por el humanista, prueba de aplicaciones hasta determinar cuáles son las más idóneos para nuestra propuesta, instalación e implementación de la solución… que sería mucho más tiempo (algo más de 300 horas) si no que se refieren exclusivamente a los tiempos en semanas (indicando que establecemos la equivalencia de 1 semana = 4 horas) que se necesitarían para implementar cada uno de los sistemas informáticos propuestos en cada uno de los tres escenarios, siendo el escenario 1 una implementación que podría hacer un usuario avanzado y siendo los escenarios 2 y 3 contratados a un empresa de terceros que pudiera llevarlos a cabo.

Un factor muy importante para determinar la dimensión de estos recursos será la finalidad que el humanista interesado en adquirir el sistema informático que se desarrolla en este proyecto le dé al suyo en términos por ejemplo de la audiencia a la que irá destinado, con esto se quiere decir que un humanista puede querer tener las prestaciones de nuestro sistema informático únicamente a título personal, o quizá quiera habilitar el sistema para una comunidad científica, o quizá ofrecerlo como un servicio profesional.

Como podría haber multitud de escenarios en esta memoria se van a detallar los costes económicos y las planificaciones que se asocian a tres escenarios distintos, que son los siguientes:

- Escenario 1, humanista que implementa el sistema de manera local para uso a título personal
- Escenario 2, humanista implementa el sistema para una comunidad (científica, cultural, de ámbito general…)
- Escenario 3, humanista implementa el sistema con fines profesionales para ofrecer el sistema como servicio

En todos los casos, se tendrán que presupuestar tres recursos de manera principal:

- Recursos Hardware, equipos informáticos, como servidores, ordenadores personales…, que albergarán las aplicaciones a implementar.
- Recursos Software, aplicaciones informáticas que se instalarán en los servidores u ordenadores personales.
- Recursos Humanos, habilitar los recursos anteriores requiere de una pericia y conocimiento de personal cualificado para llevar a cabo todo lo descrito en este proyecto.

Cabe destacar que los costes económicos se darán únicamente de los trabajos realizados con tareas de sistemas de información, evidentemente la labor de investigación, recopilación de información y preparación de esta, se supone que ya ha sido realizada por el humanista y por lo tanto no se tendrán en cuenta en los importes que se detallen en este apartado.

A la hora de determinar los costes de horas de personal cualificado se han tenido en cuenta dos perfiles:

- Administrador de sistemas estándar, profesional en administración de sistemas con conocimientos generales de administración de servidores y sistemas web.
  Se establece su coste horario en 25,00 €.
- Administrador de sistemas especialista en Apache SOLR, para la correcta configuración de un servidor Apache SOLR avanzado son necesarios conocimientos concretos de esta tecnología.
  Se establece su coste horario en 50,00 €.

ESCENARIO 1

Humanista que implementa el sistema de manera local para uso a título personal, los costes serán mínimos, el ámbito de acceso al sistema será muy limitado y las prestaciones del sistema serán muy básicas, pero eso sí serán suficiente.

ESCENARIO 1	IMPORTE 1	HORAS	PRECIO HORA	IMPORTE 2	SUMA
RECURSOS HARDWARE					
Ordenador personal	700,00 €			- €	700,00 €
RECURSOS SOFTWARE					
Licencia Windows 10 Home	145,00 €			- €	145,00 €
Apache SOLR	- €			- €	- €
RECURSOS HUMANOS					
Adaptación información a formato .XML	- €	8		- €	- €
Instalación - Configuración Apache SOLR	- €	4		- €	- €
Inserción documentos en Apache SOLR	- €	4		- €	- €
Pruebas del Sistema	- €	4		- €	- €
				TOTAL	**845,00 €**

Tabla 1 - Presupuesto Escenario 1

Los tiempos de ejecución del proyecto se han amplificado mucho, en caso de necesidad estos tiempos se pueden reducir de manera considerable.

ESCENARIO 1	SEMANAS						
TAREA	INICIO	FIN	1	2	3	4	5
FASE ADAPTACIÓN INFORMACIÓN	Semana 1	Semana 2					
Adaptación información a .XML	Semana 1	Semana 1					
Creación Esquema Colección	Semana 2	Semana 2					
FASE IMPLEMENTACION	Semana 3	Semana 4					
Instalación Configuración Apache SOLR	Semana 3	Semana 3					
Inserción Documentos .XML en SOLR	Semana 4	Semana 4					
FASE PRUEBAS	Semana 5	Semana 5					
Consultas pruebas	Semana 5	Semana 5					

Tabla 2 - Planificación Escenario 1

ESCENARIO 2

Humanista implementa el sistema para una comunidad (científica, cultural, de ámbito general…), las implicaciones de implementación son mayores pues se recomienda habilitar una página web con WORDPRESS para una mayor accesibilidad a la información.

ESCENARIO 2	IMPORTE 1	HORAS	PRECIO HORA	IMPORTE 2	SUMA
RECURSOS HARDWARE					
Servidor Apache SOLR *	119,88 €			- €	119,88 €
Servidor Web WORDPRESS *	47,88 €			- €	47,88 €
				SUBTOTAL 1	**119,88 €**
RECURSOS SOFTWARE					
Instalación - Configuración Apache SOLR	- €			- €	- €
Instalación WORDPRESS	- €			- €	- €
				SUBTOTAL 2	**- €**
RECURSOS HUMANOS					
Adaptación información a formato .XML	- €	8	25,00 €	200,00 €	200,00 €
Instalación - Configuración Apache SOLR	- €	4	25,00 €	100,00 €	100,00 €
Inserción documentos en Apache SOLR	- €	4	25,00 €	100,00 €	100,00 €
Instalación WORDPRESS	- €	4	25,00 €	100,00 €	100,00 €
Conexión SOLR-WORDPRESS	- €	4	25,00 €	100,00 €	100,00 €
Pruebas del Sistema	- €	4	25,00 €	100,00 €	100,00 €
				SUBTOTAL 3	**700,00 €**
				TOTAL 1 AÑO	**819,88 €**
				TOTAL 5 AÑOS	**1.299,40 €**

* Servicios con cuota anual, se estima el coste para 1 año - Tarifas proveedor HOSTINGER

Tabla 3 - Presupuesto Escenario 2

Los tiempos de ejecución del proyecto se han amplificado algo, pero no tanto como en el escenario anterior, en caso de necesidad estos tiempos se pueden reducir de manera ligeramente.

ESCENARIO 2	SEMANAS								
TAREA	INICIO	FIN	1	2	3	4	5	6	7
FASE ADAPTACIÓN INFORMACIÓN	Semana 1	Semana 2							
Adaptación información a .XML	Semana 1	Semana 1							
Creación Esquema Colección	Semana 2	Semana 2							
FASE IMPLEMENTACION	Semana 3	Semana 6							
Instalación Configuración Apache SOLR	Semana 3	Semana 3							
Inserción Documentos .XML en SOLR	Semana 4	Semana 4							
Instalación WORDPRESS	Semana 5	Semana 5							
Conexión SOLR - WORDPRESS	Semana 6	Semana 6							
FASE PRUEBAS	Semana 7	Semana 7							
Consultas pruebas	Semana 7	Semana 7							

Tabla 4 - Planificación Escenario 2

ESCENARIO 3

Humanista implementa el sistema con fines profesionales para ofrecer el sistema como servicio. Este servicio puede darse a varios clientes con el mismo servidor de SOLR y servicio de hosting, pero en cualquier caso los costes indicados se aplicarían por cada cliente.

Este escenario también se puede ofrecer a un posible cliente para hacer la implementación de manera exclusiva para ese posible cliente.

ESCENARIO 3	IMPORTE 1	HORAS	PRECIO HORA	IMPORTE 2	SUMA
RECURSOS HARDWARE					
Servidor Apache SOLR ***	444,00 €			- €	444,00 €
Servidor Web WORDPRESS **	119,88 €			- €	119,88 €
				SUBTOTAL 1	**444,00 €**
RECURSOS SOFTWARE					
Apache SOLR	- €			- €	- €
WORDPRESS	- €			- €	- €
				SUBTOTAL 2	**- €**
RECURSOS HUMANOS					
Adaptación información a formato .XML	- €	8	25,00 €	200,00 €	200,00 €
Instalación - Configuración Apache SOLR	- €	4	50,00 €	200,00 €	200,00 €
Inserción documentos en Apache SOLR	- €	4	25,00 €	100,00 €	100,00 €
Instalación WORDPRESS	- €	4	25,00 €	100,00 €	100,00 €
Conexión SOLR-WORDPRESS	- €	4	25,00 €	100,00 €	100,00 €
Pruebas del Sistema	- €	8	25,00 €	200,00 €	200,00 €
				SUBTOTAL 3	**900,00 €**
				TOTAL 1 AÑO	**1.344,00 €**
				TOTAL 5 AÑOS	**3.120,00 €**

** Servicio con cuota anual, se estima el coste para 1 año - Tarifas proveedor HOSTINGER

*** Servicio con cuota anual, se estima el coste para 1 año - Tarifas proveedor SEARCHSTAX

Tabla 5 - Presupuesto Escenario 3

Los tiempos de ejecución del proyecto en este caso están mucho más ajustados y bajar esos tiempos sería menos indicado.

ESCENARIO 3			SEMANAS									
TAREA	INICIO	FIN	1	2	3	4	5	6	7	8	9	10
FASE ADAPTACIÓN INFORMACIÓN	Semana 1	Semana 2										
Adaptación información a .XML	Semana 1	Semana 1										
Creación Esquema Colección	Semana 2	Semana 2										
FASE IMPLEMENTACION	Semana 3	Semana 8										
Instalación Configuración Apache SOLR	Semana 3	Semana 4										
Inserción Documentos .XML en SOLR	Semana 5	Semana 5										
Instalación WORDPRESS	Semana 6	Semana 6										
Conexión SOLR - WORDPRESS	Semana 7	Semana 8										
FASE PRUEBAS	Semana 9	Semana 10										
Consultas pruebas	Semana 9	Semana 10										

Tabla 6 - Planificación Escenario 3

Capítulo 2: Estado del arte

Para situar el panorama actual de las Humanidades Digitales conviene tener en cuenta una serie de tecnologías que se han ido desarrollando a lo largo de los últimos treinta años, aunque continuamente surgen nuevas tecnologías que mejoran a las anteriores o que abren nuevas rutas a la consecución de los objetivos planteados en los diversos proyectos de las Humanidades Digitales, tecnologías que se pueden enmarcar dentro de las áreas de la Inteligencia Artificial IA, el Procesamiento del Lenguaje Natural PLN y el Análisis de Datos [9].

Puede observarse que el campo de las Humanidades Digitales adolece de una definición metodológica que aporte información sobre los principales pasos o etapas a realizar para alcanzar un diseño inicial que cumpla con los requisitos esperados por un humanista profesional o investigador que aporta la información de un dominio.

Como las técnicas más utilizadas actualmente pueden organizarse en dos grandes grupos, uno cualitativo en el que el almacenamiento de la información se suele realizar en una base de datos y otro grupo, el de las técnicas cuantitativas, en el que se utilizan principalmente sistemas de almacenamiento no SQL, una primera decisión es esta, qué técnicas serán las más prometedoras. Entre las técnicas cualitativas para el modelado y la organización de los datos, destacamos (no de forma exhaustiva) el etiquetado y anotación de los datos e información, el análisis del discurso (periodísticos, bibliográficos…), la extracción de información (*web scraping, web harvesting*), la interpretación ontológica, la geolocalización con sistemas de información geográfica (SIG) o la visualización de datos e información con infografías, mapas ilustrados, gráficas, mapas animados, árboles genealógicos, grafos… Entre las técnicas cuantitativas, tan en auge actualmente para la gestión de información, destacamos las que se enmarcan en el área de la recuperación de información y la minería de textos para la clasificación de textos, la identificación de temas o aspectos abstractos (*topic modeling* o *Formal Concept Analysi*s), la identificación de entidades nombradas, el análisis de sentimientos y otras tareas relacionadas con información textual. Estas técnicas utilizan redes de neuronas artificiales, métodos de análisis estadísticos o probabilísticos en general o técnicas de aprendizaje automático profundo. En lo que sigue hay un apartado con más información al respecto.

A partir de las técnicas mencionadas, se pueden desarrollar sistemas que realicen tareas complejas para conformar un Proyecto, como el que se presenta en esta memoria, sobre una colección multimedia de documentos o “Catálogo universal de museos y

yacimientos arqueológicos de la Antigua Grecia" que contiene una descripción de 154 museos y 277 yacimientos arqueológicos.

Afrontar una colección de estas características exige decidir en la etapa de pre-proceso, sobre la necesidad de una transcripción o no, si realizar o no cambios de formato o qué tareas básicas sobre el texto o las imágenes realizar, como el etiquetado automático lingüístico (etiquetado POS) de los textos, la anotación de imágenes o planos o expertos o usuarios en general, la normalización no lingüística de las palabras (truncamiento o *steamming*) u otros. Esta es una segunda fase de decisión en el diseño del sistema.

A continuación, comienza la etapa relacionada con la extracción de datos o información contenida en la colección almacenada de documentos y que será utilizada en análisis posteriores. Para esta etapa ya se dispone de tecnologías sobre corpus textuales que facilitan la realización de ciertas tareas más o menos complejas para alcanzar el objetivo principal definido. Sin ánimo de ser exhaustivos, posteriormente se dedica un apartado donde se citan algunas herramientas básicas.

Además, como antecedentes del trabajo concreto que se presenta en la memoria, se describen brevemente dos proyectos: El proyecto DIMH ([6], [8], [9]) de carácter multidisciplinar, incorpora un corpus digital compuesto por una colección de mapas, planos y dibujos (en formato de imagen) y fichas con texto semiestructurado. El segundo proyecto también muy relacionado con este trabajo es "Iconografía musivaria en la Península Ibérica en época romana: investigación y difusión desde el campo de las Humanidades Digitales" [10] que consiste en el diseño de una herramienta informática que permite la gestión de la información recopilada sobre los mosaicos romanos encontrados en la península ibérica (https://musivariahd.com/).

2.1 Técnicas, tareas y recursos

Esta presentación de la tecnología la organizamos sobre la base de técnicas, tareas y recursos que forman parte de un proyecto [10].

- Técnicas, o conjunto de procedimientos que se usan en un arte, en una ciencia o en una actividad determinada, en especial cuando se adquieren por medio de su práctica y requieren habilidad.

En esta definición podemos incluir muchos procedimientos como pueden ser la búsqueda basadas en bolsas de palabras, clasificación basada en ontologías, análisis de frecuencias (Herramienta VOYANT TOOLS), o análisis de estilo métricos usando Deep Learning.

Las Técnicas las podemos clasificar en dos grandes grupos:

1.- Técnicas cualitativas, como modelado y organización de los datos, etiquetado y anotación (individual o colectiva), análisis de texto cualitativo, discurso e imágenes (periodísticos, bibliográficos…), extracción de información (web scraping, web harvesting), clasificación de textos e interpretación ontológica, localización con sistemas de información geográfica (SIG), visualización con infografías, mapas ilustrados, gráficas, mapas animados, árboles genealógicos, árboles de problemas y soluciones, grafos…

2.- Técnicas cuantitativas, como recuperación de información, encuestas y su análisis (estudios de audiencias…), análisis estadísticos o probabilísticos, minería de textos, identificación de temas o aspectos abstractos (topic modeling, Latent Dirichlet Allocation (LDA), Formal Concept Analysis (FCA) y otros), redes de neuronas artificiales, análisis de sentimientos (basada en un diccionario y otras), aprendizaje automático o profundo, procesamiento automático del lenguaje Natural (PLN, tokenización, steamming, etiquetado POS), identificación de entidades nombradas,

- Tareas que se realizana partir de las técnicas antes descritas, como por ejemplo una búsqueda de información a partir de la petición de un usuario (buscadores de Internet como Google o Bing), la clasificación de documentos, la recuperación de imágenes similares a partir de otra imagen ya dada, la simplificación de textos y otras.

A partir de la agrupación de un número mayor o menor de tareas relacionadas en torno a una temática o finalidad concreto podemos conformar un proyecto por completo.

- Recursos, son los instrumentos necesarios para llegar a cabo las tareas que conforman un proyecto concreto. Distinguimos dos tipos de recursos principalmente (1) las herramientas tecnológicas (analizadores lingüísticos, contadores de palabras, identificadores de entidades nombradas, bases de datos, bibliotecas digitales y otros) y (2) las herramientas colaborativas de gestión de proyectos o educativas como son las aulas virtuales tipo TEAMS o la plataforma AVIP-UNED.

Para este proyecto se consideran de especial interés las siguientes aplicaciones o recursos.

- VOYANT TOOLS http://voyant-tools.org, se trata de una aplicación web (código libre en abierto bajo licencia pública general GNU) para el análisis de textos digitales, en concreto análisis de frecuencias.

- SPACY https://spacy.io, es una librería de software para procesamiento de lenguajes naturales programado en Python, es software libre con licencia MIT.

- OMEKA https://omeka.org, es una plataforma, con licencia PL-3.0, para publicar y compartir colecciones digitales enriquecidas con materiales multimedia. Tiene dos versiones Omeka Classic para proyectos individuales (como es este caso) y Omeka S para proyectos compartidos.

- NETLINE https://www.netline.com, es una herramienta de representación espacio temporal para la creación de mapas anotados a partir de colecciones OMEKA, tiene licencia GPL 3.0, diseñado como una colección de plugins para OMEKA.

- TROPY https://www.tropy.org, es una aplicación de escritorio, gratuita y de código abierto con licencia AGPL, para gestionar documentos gráficos (JPG, PNG, SVG, WEBP…) y añadir plantillas de metadatos personalizadas en proyectos de investigación agrupando una colección en un único archivo para una mejor organización. Ofrece posibilidades de conectividad con otras herramientas como OMEKA S o ZOTERO.

- TEITOK http://www.teitok.org, es una plataforma para ver, crear y editar corpus tanto con anotaciones lingüísticas como con marcado de texto enriquecido siguiendo el estándar TEI. Muy extendida en proyectos de Humanidades Digitales. Con diseño modular, cada módulo procesa un corpus distinto (manuscritos, audios…), visualización de árboles de dependencia y geolocalización de manuscritos.

- TAGTOG https://www.tagtog.com, es una herramienta web de anotación que permite entrenar modelos de Inteligencia Artificial para la obtención de información relevante. La anotación puede ser tanto manual como automática para encontrar entidades, desambiguar, clasificar textos o establecer relaciones entre entidades. Está disponible tanto en aplicación web como en aplicación de escritorio

- ZOTERO https://www.zotero.org, es un gestor de referencias bibliográficas que permite crear, almacenar, organizar, compartir e insertar esas referencias. Es software de código abierto, multiplataforma y con complementos para navegadores de Internet.

No todas ellas se van a utilizar en el caso de estudio de Arqueogriegos, pero sí se han estudiado y evaluado.

2.2. Casos de Uso

Una vez exploradas algunas de las tecnologías que pueden ser usadas para este trabajo se ha procedido a revisar una serie de proyectos llevados a cabo por diversas instituciones y empresas que puedan guardar más o menos similitudes con el proyecto ARQUEOGRIEGOS para fijar cuales de esas tecnologías formarán parte del mismo. En concreto el proyecto DIMH y el proyecto MUSIVARIA, para documentar las tecnologías que se han usado, valorar si los objetivos iniciales se cumplen en el desarrollo del diseño de la solución propuesta y comprobar si ese el resultado final es adecuado a nuestro proyecto.

Si bien el análisis de un tercer proyecto BURCKHARDTSOURCE.ORG parecía muy interesante y fue recomendado de manera especial por la tutora de este proyecto, durante la mayor parte del periodo (abril de 2023 a julio de 2023) de desarrollo de este proyecto iniciado en febrero de 2023 no estaba disponible la web y por lo tanto se tuvo que descartar ese estudio. En una última comprobación realizada el 25 de julio de 2023, la web ya vuelve a estar disponible, pero ya resulta inútil su análisis puesto que ya se determinó qué tecnologías aplicar a este proyecto.

2.2.1. DIMH

Este proyecto [9] se puede considerar como pionero [2] en el campo de las Humanidades Digitales, de carácter multidisciplinar y ha servido como punto de partida para diversas investigaciones. Se trabaja con un corpus digital compuesto por una colección de mapas, planos y dibujos (en formato de imagen), en concreto se trata de un total de 7792 fichas con información textual.

La información de las 7792 fichas originalmente se encuentra codificada con metadatos en formato RDF-FC y es descargada en un total de 8 ficheros [1], en el corpus digitalgenerado para el proyecto se crea un archivo XML para cada una de las fichas, que posteriormente son enriquecidos con contenido textual. La información se almacenó en una página web con un SPARQL Endpoint para la realización de las consultas (tecnología de la web semántica).

El análisis formal de conceptos FCA es una técnica de organización y modelado que detecta relaciones entre contenidos y posteriormente los organiza en base a esas relaciones. En la aplicación de esta técnica al corpus DIMH se realizan las siguientes etapas:

- Selección y extracción de información, se realiza sobre el conjunto de objetos (fichas) que comparten un conjunto concreto de atributos (campos). Para el modelado se seleccionan los campos que se consideran más oportunos, y se realizan otras tareas de tipo lingüístico como eliminar palabras vacías.

- Creación del contexto formal, mediante matriz de adyacencia (aparición de un atributo en un objeto).
- Reducción del contexto formal, el contexto formal generado inicialmente puede contener información redundante o poco significativa por lo que se recomienda realizar una reducción localizando la terminología que permita identificar más relaciones sin pérdida de información.
- Ejecución del algoritmo de FCA, se generan los conceptos formales. Este algoritmo es una implementación propia de otro algoritmo llamado Next Neighbourhoods.

El modelado del corpus DIMH consistió en generar un contexto formal a partir de la información de las fichas y la ejecución del algoritmo FCA considerando la información previa incluida en una taxonomía parcial. Se observó que a pesar de representar las fichas mediante su texto y los términos en la taxonomía, ocurría que casi todos los atributos seleccionados proceden del texto y no de la taxonomía, por lo que se realizó un nuevo refinamiento representando las fichas solo con los términos de la taxonomía mediante normalizaciones como representar en un único término las mismas realizaciones léxicas de la misma palabra (singular-plural, masculino-femenino), diferentes formas ortográficas de un mismo término, o la propia jerarquía de la taxonomía.

La inclusión de la información taxonómica previa al procesamiento del algoritmo FCA genera un contexto formal de entrada más grande y por lo tanto la ejecución del algoritmo será más lenta, por lo tanto, hay que hacer uso de alguna solución intermedia por ejemplo consistente en incluir los términos de las clases de la jerarquía en el término a representar, esta solución se denominó *Índice con tipologías (reducido)*. También se aplicaron otras experimentaciones como la denominada *Índice sin tipología*, que consistía en eliminar los términos más frecuentes.

Finalmente se desarrolló una aplicación web [9] para visualizar los resultados de la búsqueda sobre los conceptos formales y sobre la base de la terminología.

2.2.2. Musivaria

El proyecto “Iconografía musivaria en la Península Ibérica en época romana: investigación y difusión desde el campo de las Humanidades Digitales” [3], proyecto de una tesis doctoral [11], se ha considerado interesante para este proyecto.

La palabra musivaria designa a la técnica y arte de elaborar mosaicos, este proyecto en concreto se centra en el diseño de una herramienta informática que permita la gestión de la información recopilada sobre los mosaicos romanos encontrados en la península ibérica. Existe un gran volumen de información al respecto en distintos formatos que ha sido tratado

con una metodología consistente en aplicar técnicas cualitativas como la triangulación de datos y cuantitativas como el macro análisis. Otra herramienta determinante fue la creación de la ontología TaxMOS HD. En su web de referencia https://musivariahd.com se puede encontrar más más información y los contenidos completos del proyecto.

El corpus de base usado para este proyecto es el Corpus de Mosaicos de España http://www.proyectos.cchs.csic.es/mosaicosromanos/ que aúna almacenamiento de datos gráficos y textuales, modelos 3D, vídeos, presentación de la información mediante geolocalización, utilización de estándares geográficos y vocabularios controlados como GeoNames e iconográficos como Iconclass y el AAT del Getty.

A partir de ese corpus se crea inicialmente una base de datos con una muestra representativa de 141 municipios y distritos, con la intención de ir ampliándola progresivamente, con 276 mosaicos, conteniendo información textual y geoespacial, así como imágenes, vídeos y modelos 3D para conformar un completo catálogo.

La base de datos forma parte del sistema informático denominado 2ArchIS que usa tecnología del lado del servidor AMP, servidor Apache, motor de base de datos MySQL y lenguaje PHP y HTML.

El proyecto desarrolla la aplicación de un modelo matemático para el estudio y el análisis estadístico del catálogo antes indicado, este modelo se esquematiza con (1) un SIG para la ubicación espacial, (2) el análisis iconográfico de los mosaicos, (3) la clasificación Iconclass de los mosaicos y (4) un análisis estadístico.

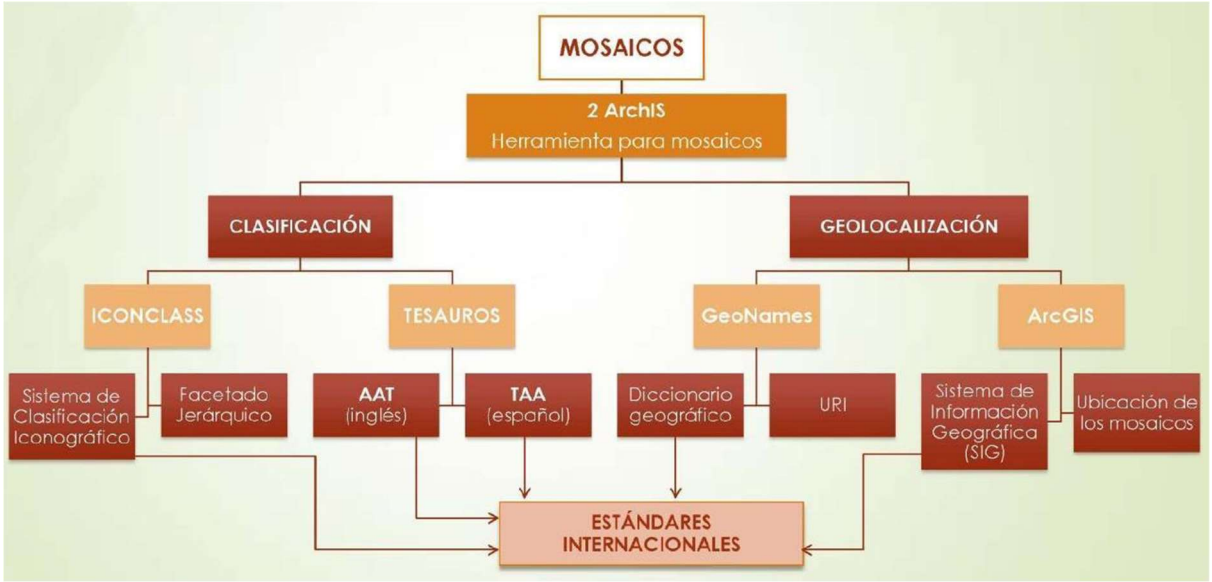


Figura 1 - Esquema para las tareas de clasificación y recursos utilizados para organizar el contenido

El desarrollo de la investigación se ha llevado a cabo desde una perspectiva de abstracción aplicando el modelo Distant Reading de Franco Moretti, aplicando metodología de Teoría de Redes que se analizan tanto a nivel geográfico como histórico y textual aplicando técnicas de Co-word Analysis, correlación entre términos, Teoría del Actor-Red TAR, Análisis Estadístico de Datos Textuales AEDT, Método de regresión lineal o correlación de Pearson, aplicación de visualización de datos con grafos GEPHI y la herramienta para análisis de textos VOYANT-TOOLS. En la figura 1 puede observarse un esquema para las dos tareas básicas (clasificación y geolocalización), así como los recursos existentes utilizados en cada una de ellas.

2.3 Recursos de información en abierto

Las Humanidades Digitales son interdisciplinares y requieren de la convivencia y colaboración entre profesionales e investigadores de distintas áreas de conocimiento, especialmente de humanistas e informáticos [9]. A la hora de abordar un proyecto en esta área lo primero es comprobar si ya existe alguna solución con las herramientas tecnológicas actuales (libres o de código propietario), y que permita, bien sea de manera parcial o total, desarrollar una propuesta y en caso de ser de manera parcial se tendrá que desarrollar software específico, que posiblemente pueda ser reusado en otros proyectos.

Como en comentó anteriormente en esta memoria, contextualizamos las Humanidades Digitales en el marco de la Inteligencia Artificial IA, el Análisis de Datos y el Procesamiento del Lenguaje Natural PLN, pues actualmente estas competencias aparecen con muy alta probabilidad en proyectos de este tipo. Estas áreas tecnológicas están en un desarrollo exponencial en la actualidad y el interés de este apartado es mostrar la adecuación y viabilidad de los recursos tecnológicos disponibles para almacenamiento y gestión de la información y concluir sobre la tecnología más adecuada. Así se comienza presentando brevemente las bases de datos relacionales (como el usado en el proyecto MUSIVARIA) y modelos de almacenamiento no relacionales o basados en la indexación (como el usado en el proyecto DIMH). También se describen brevemente los almacenamientos basados en principios ontológicos, que, aunque se han revisado, no se utilizarán en este trabajo.

2.3.1. Bases de datos relacionales

La primera decisión trascendental a la hora de iniciar un proyecto de Humanidades Digitales es la de seleccionar una tecnología para el almacenamiento de la información acorde a los requerimientos del proyecto, con esta tecnología se podrá diseñar el modelo de datos a crear. Así existirán proyectos en los que las clásicas bases de datos relacionales podrían ser suficientes y otros en los que sería imposible gestionar adecuadamente datos heterogéneos en cuanto a formato o requisitos específicos.

El tipo de base de datos más usado hasta la primera década del siglo XXI son las bases de datos relacionales [4], este modelo relacional se pasa en los principios postulados por Edgar Frank Cook (IBM) por el año 1970. Una base de datos está formada por un conjunto de tablas (relaciones), que a su vez están formadas por una serie de campos (columnas) y registros (filas), los primeros definen las características y los segundos son las entradas de datos. Como característica fundamental destaca la capacidad de relacionar tablas de datos entre sí, existen por lo tanto una estructuración básica inicial, y generalmente los datos que

se almacenan son datos textuales con diferentes formatos específicos como textos y valores numéricos.

Existen diversos gestores de bases de datos de este tipo, pero su gestión se basa en el lenguaje de consulta SQL (Structured Query Language), con él se pueden realizar las operaciones básicas CRUD Create – Read – Update – Delete). Dentro de estos gestores destacamos el gestor propietario de a Microsoft Access y otro libre como es MySQL. El proyecto MUSIVARIA HD usa este modelo de base de datos, en concreto usa el gestor de bases de datos MySQL.

2.3.2. Bases de datos no relacionales

El aumento exponencial de la información en las últimas décadas en todos los ámbitos, la diversidad de los formatos de esta información y la necesidad de la alta disponibilidad de esa información lleva de manera irremediable al diseño de nuevos paradigmas en el campo del almacenamiento de los datos, surgiendo las denominadas bases de datos no relacionales o NoSQL (no solo SQL, pues si bien se pueden realizar consultas SQL en ellas es a modo de apoyo y no como funcionalidad intrínseca).

En estas bases de datos los datos no tienen una estructura definida, y por lo tanto los conceptos de tablas, campos, registros, pierden su razón de ser y se generan otros conceptos de almacenamiento basados en la indexación de diversos objetos como documentos, grafos, datos clave/valor, datos multivalor, orientadas a objetos, datos tabulares, etc.

A continuación, se relacionan una serie de gestores de este tipo:

- **Cassandra**, basado en un modelo de almacenamiento tabular, es de código abierto y escrito en JAVA, soportada por la Apache Software Foundation, permite gran volumen de datos de manera distribuida [12]. Tiene como principal objetivo la escalabilidad lineal y la disponibilidad. Su arquitectura está basada en una serie de nodos iguales que se comunican con el protocolo P2P (máxima redundancia). Ofrece soporte robusto para múltiples centros de datos con replicación asíncrona sin servidor maestro lo que permite operaciones de baja latencia para todos los clientes. Con gran rendimiento. Su modelo de datos consiste en particionar las filas que se reorganizan en tablas donde las claves primarias de cada tabla tienen un primer componente que es la clave de partición, dentro de una partición, las filas son agrupadas por las columnas restantes de la clave, las demás columnas pueden ser indexadas por separado de la clave primaria. Con lenguaje propio CQL (Cassandra Query Language) con sintaxis similar a SQL. TWITTER usa este gestor de base de datos.

- **MongoDB**, orientado a documentos y de código abierto, los datos se guardan en estructuras de datos llamadas BSON que son especificaciones similares a JSON, con un esquema dinámico que consigue una integración de los datos fácil y rápida. Con un uso muy amplio en entornos industriales [13].

  A la hora de la realización de los trabajos de indexación que se realizan con otras tecnologías además de las bases de datos NoSQL, destacamos el siguiente proyecto:
- **Lucene**, es un motor de búsqueda de texto de código abierto, con una comunidad muy activa. Se trata de una librería de búsqueda completa de textos, no es un servidor o un rastreador web. Este concepto de búsqueda completa se refiere a que no se realiza la búsqueda en los datos contenidos en una base de datos, sino que realiza una búsqueda en todos los datos de un texto.

Hay varias herramientas basadas en Lucene de los que en esta memoria destacaremos principalmente a SOLR y ELASTICSEARCH.

- SOLR es un motor de búsqueda basado en Apache y en la librería LUCENE, escrito en JAVA y de código abierto, permite integrar motores de búsqueda verticales, que, a diferencia de los motores de búsqueda generales, pueden enfocarse en un segmento específico [14].

  SOLR es un acrónimo de Searching On Lucene Replication. Entre sus características principales destacamos la posibilidad de restringir las búsquedas mediante filtrado, la búsqueda por facetas nos ofrece sugerencias de filtrado, clasificación de los resultados de las búsquedas, búsquedas por sinónimos, integración con bases de datos. Funciona recorriendo los documentos seleccionados e incorporándonos a un índice, proceso llama indexado añadiendo las palabras clave de los documentos que hayamos indicado al índice. Este índice acepta datos de muchas fuentes, tales como archivos XML, CSV, archivos Word o PDF. Solr en lugar de buscar en el texto mismo, realiza la búsqueda de la palabra clave en el índice, y a continuación nos indica en qué documentos se encuentra dicha palabra clave. Este tipo de índice se llama índice invertido porque la estructura de los datos se basa en las palabras clave en lugar de basarse en la página. Esta es la mayor diferencia funcional con nuevas tecnologías basadas en modelos del lenguaje ya que aportan información para contrastar las respuestas.
- ELASTICSEARCH es un motor de búsqueda y recuperación de documentos de tipo JSON basado en Apache Lucene para proveer a diversas aplicaciones de capacidades de búsqueda a texto completo mediante una API RESTful. Escrito en JAVA y publicado como código abierto bajo las condiciones de la licencia Apache.

Provee igualmente un completo Query DLS (Lenguaje Específico de Dominio) basado en JSON para la definición de las búsquedas, con la posibilidad de añadir y posteriormente buscar y analizar sobre grandes cantidades de datos en tiempo real de forma distribuida. Tiene disponibles varias librerías para clientes como Java, Ruby, PHP, .NET, Python…

Permite una arquitectura distribuida, escalable horizontalmente y en alta disponibilidad, diseñado para permitir búsquedas muy rápidas de texto completo, permite almacenar documentos JSON indexando todos sus campos. Por todas estas características se recomienda para soluciones de buscador, analítica de métricas, análisis de logs y análisis de seguridad.

En la Tabla 7 - SOLR vs ELASTICSEARCH se muestra una comparativa entre ambos recursos: SOLR y ELASTICSEARCH.

SOLR	ELASTICSEARCH
Visualización de datos	
	Mejor visualización de datos gracias a Kibana
Popularidad	
	Más popular
Administración	
	Más fácil
Personalización	
Más personalizable	
Capacidades de exportación	
Más formatos de exportación	
Procesamiento de grandes cantidades de datos	
Mayor capacidad de procesar grandes cantidades de datos	
Escalabilidad	
	Más escalable
Tipo de búsqueda	
Más orientada a textos	Más orientada a búsquedas analíticas y agrupamiento de datos
Documentación	
Mejor y más detallada documentación	
Comunidad	
Amplia comunidad de código abierto	La compañía, elastic, controla la dirección del proyecto

Tabla 7 - SOLR vs ELASTICSEARCH

2.3.2. Almacenamiento basado en ontologías

El almacenamiento basado en ontologías es un enfoque bastante popular para la gestión de conocimientos y datos en diversas aplicaciones. Una ontología es una representación formal y explícita de los conceptos y relaciones en un dominio de conocimiento específico. Este tipo de almacenamiento usa una ontología para organizar y estructurar datos de manera más efectiva haciendo más énfasis en la semántica.

En la actualidad, se está trabajando en el desarrollo de técnicas y herramientas para la integración de múltiples ontologías, lo que permitiría la gestión y organización de datos más complejos y heterogéneos. Destacan las ontologías dinámicas, que son aquellas que pueden ser modificadas y actualizadas en tiempo real, lo que permite una gestión más ágil y eficiente de los datos. El uso de técnicas de aprendizaje automático en el almacenamiento basado en ontologías permite la detección de patrones y relaciones en los datos, lo que a su vez facilita la toma de decisiones y el descubrimiento de nuevos conocimientos. La integración de tecnologías *blockchain* en el almacenamiento basado en ontologías permite la creación de sistemas más seguros y transparentes para el intercambio de datos y conocimientos.

El almacenamiento basado en ontologías dispone de una gran variedad de herramientas para su implementación y desarrollo, como:

- Lenguajes de ontologías con los que se crean y definen las ontologías: OWL y RDF.
- Sistemas de gestión de ontologías, herramientas que permiten la creación, gestión y almacenamiento de ontologías, como pueden ser Protégé, TopBraid Composer y PoolParty.
- Tecnologías de almacenamiento de datos, que almacenan y permiten el acceso a los datos basados en ontologías, se usan diversas tecnologías: como bases de datos relacionales, de grafos y sistemas de almacenamiento de tripletes RDF.
- Tecnologías de visualización de datos, que permiten la visualización de datos basados en ontologías de manera más accesible e intuitiva para los usuarios, como por ejemplo Cytoscape y OntoGraf.
- Tecnologías de integración de datos con herramientas que permiten la integración de datos de diversas fuentes en una ontología común, por ejemplo, Karma, Ontop y Silk.
- Tecnologías de razonamiento, que permiten la inferencia de nuevos conocimientos a partir de una ontología existente, como por ejemplo Pellet y HermiT.

La excelente estructuración del catálogo del experto humanista hace que no sea necesario el desarrollo de una nueva ontología para estructurar la información.

2.4 Herramientas para Análisis de Datos

La investigación empírica, o basada en evidencias observables se realiza en varias fases que deben ser coherentes entre sí. Las principales fases de la investigación empírica son: las preguntas, el problema, las hipótesis, la selección del método, la definición y medición de las variables, el diseño o selección de muestras, el análisis de datos y la interpretación y valoración de resultados [4].

El proceso de investigación comienza con la formulación de cuestiones para las que no tenemos explicación satisfactoria, y en ese caso diremos que tenemos un problema o problemas a resolver. Tanto el problema como las cuestiones deben ser definidos de manera clara y precisa. Las hipótesis son enunciados comprobables, y son explicaciones provisionales de las cuestiones planteadas en la investigación. Las variables cuantitativas admiten la relación “mayor que”, en cambio, las variables cualitativas no lo admiten. Por ejemplo, el valor 50 de la variable “peso” es mayor magnitud que 40. En cambio, los valores de la variable “tipo de trabajo” no tienen relación cuantitativa.

Las variables tienen características diferenciadas según el tipo de medición con que obtenemos los datos, lo cual es un factor determinante para la selección de las técnicas estadísticas de análisis de datos. Medir es asignar números a objetos o sucesos de acuerdo con un conjunto de reglas previamente establecidas, y su finalidad es obtener datos lo más válidos y precisos que sea posible. Entre los principales métodos de la investigación descriptiva se encuentran el observacional y el de encuestas.

La investigación observacional consiste en registrar el comportamiento en el entorno habitual del sujeto. La observación sin intervención tiene por finalidad observar el comportamiento tal como ocurre de forma natural, y en ella el observador se limita a registrar lo que observa, sin manipular ni controlar. La investigación con encuestas se caracteriza por utilizar cuestionarios para registrar las respuestas de los sujetos. La finalidad más habitual de la investigación con encuestas es la descripción de pensamientos, opiniones y sentimientos. El principal inconveniente es el sesgo introducido por el elevado índice de encuestas no contestadas y la dificultad para trabajar con muestras representativas.

Actualmente los datos observados pueden ser analizados con aplicaciones de ordenador, por lo que deben ser almacenados en archivos informáticos. Las bases de datos contienen datos provenientes de un número de observaciones más o menos grande respecto de un conjunto de variables que también puede llegar a ser bastante grande. Si bien ya se describieron de manera breve algunas herramientas para el análisis de datos, a continuación,

se detallan dos de especial relevancia para el caso de uso planteado, en el que no hay encuestas y solo está registrado l observado.

VOYANT TOOLS (https://voyant-tools.org) es un entorno de análisis y lectura de texto basado en la Web que facilita la lectura y las prácticas interpretativas profesionales, humanistas, estudiantes de humanidades, así como para el público en general. Sirve para estudiar textos alojados en línea o que se hayan editado cuidadosamente y estén alojados en el computador personal.

Asimismo, permite, de manera fácil, leer y entender diferentes estadísticas sobre los vocablos (frecuencia absoluta, frecuencia normalizada, asimetría estadística y palabras diferenciadas), buscar palabras clave en contexto y exportar los datos y las visualizaciones en diferentes formatos como csv, png y html. De igual manera, posibilita la visualización de tendencias por medio de gráficos de distribución que muestran los términos más concurrentes en el corpus textual.

También facilita la revisión y lectura de textos completos al indicar la cantidad de texto que tiene cada documento, proporciona una visión general de ciertas estadísticas del corpus y muestra diversas concordancias que indican la concurrencia de palabras o términos claves con un poco de contexto gramatical.

Hay que destacar que es un proyecto de código abierto, el cual está disponible a través de GitHub. El código está bajo una licencia GPL3 y el contenido de la aplicación web cuenta con una licencia Creative Commons By Attribution, por lo tanto, hay permiso para crear y usar capturas de pantalla y videos de la herramienta, siempre y cuando se den los créditos correspondientes.

Presenta una interfaz muy intuitiva y amigable, se pueden analizar URLs, o cargar archivos de distintos formatos como MS Word, MS Excel, PDF, XML…Tras introducir el texto a analizar se realiza un escaneado de los datos contenidos y se muestra la información sobre la frecuencia y tendencia de las palabras contenidas. Se pueden examinar varias visualizaciones y ver las frecuencias, es aconsejable eliminar las palabras vacías (stopwords), también se pueden ver otros resúmenes de datos interesantes.

A modo de experimentación se han agrupado todos los archivos MS Word entregados por el humanista peticionario de este proyecto para la zona denominada Tracia para observar el funcionamiento de esta aplicación, obteniendo unos resultados muy satisfactorios.

La segunda herramienta de análisis (lingüístico) que se destaca es SPACY, una librería de software de código abierto bajo licencia MIT y escrita en el lenguaje de programación PYTHON y CYTHON (extensión del lenguaje C para Python) y tiene como objetivo facilitar la

puesta en producción, crear una aplicación lista para su uso por el consumidor final, está diseñada para facilitar tareas de procesamiento avanzado del lenguaje natural [15].

SpaCy se creó con el objetivo de facilitar la creación de productos reales. Es decir, la librería no es tan solo una librería con la que nos quedamos en el plano más técnico y de más bajo nivel dentro de las capas que componen una aplicación de software, desde los algoritmos más internos hasta las interfaces más visuales.

La librería contempla los aspectos prácticos de un producto de software real, en el que es necesario tener en cuenta aspectos tan importantes como:

- Las grandes cargas de datos que se requieren procesar, aspecto muy importante en proyectos de Humanidades Digitales.
- La velocidad de ejecución, puesto que cuando tenemos una aplicación real, necesitamos que la experiencia sea lo más fluida posible y no podemos soportar largos tiempos de espera entre ejecuciones de los algoritmos.
- El empaquetamiento de funcionalidades de NLP (como NER) listas para desplegar en uno o varios servidores de producción, así SpaCy, no solo proporciona herramientas de código de bajo nivel, sino que soporta los procesos desde que creamos (compilamos y construimos una parte de una aplicación de software) hasta que integramos esta parte algorítmica con otras partes de la aplicación como las bases de datos o las interfaces de usuario final. Algunas de las funcionalidades más destacadas que nos ofrece la librería spaCy son POS Tagging, Dependency Parsing, Named entities, Tokenización, Segmentación de frases, Rule – based matching, es decir podremos sacar siempre que queramos tokens, postags, árboles de dependencia o entidades nombradas. Incluye también modelos de word embeddings.
- La optimización de modelos de NLP para que puedan ejecutarse fácilmente en servidores estándar (basados en CPU) sin necesidad de usar procesadores gráficos (GPU)
- La visualización gráfica integrada para facilitar la depuración o el desarrollo de nuevas funcionalidades.
- Importante mencionar también, su fantástica documentación, desde su web más introductoria hasta su comunidad en Github. Esto facilita enormemente la rápida adopción entre la comunidad de desarrollo.
- Dispone de un gran número de modelos y flujos pre-entrenados (73 flujos) en 22 idiomas diferentes. Además del soporte para más de 66 idiomas, entre ellos los modelos optimizados para el español.

Tras varias pruebas, se decidió que no era necesario su uso en el proyecto que se describe en esta memoria para la etapa de organización de la información y el desarrollo de un buscador.

Capítulo 3: Diseño

El corpus del catálogo universal para el proyecto ARQUEOGRIEGOS se generará a partir de los documentos aportados por el humanista, que serán principalmente de tres tipos:

- Contenidos de texto, información descriptiva sobre los yacimientos de la Antigua Grecia.
- Contenidos de imagen, fotografías de los yacimientos.
- Contenidos de Planos, planos de los yacimientos en formato CAD.

El alcance de este proyecto se centra en la creación del diseño del modelo para el almacenamiento de la información aportada por el humanista y el acceso a esa información, y si bien lo ideal es que esta información fuera lo más completa posible para lo que con toda probabilidad habría que realizar un proceso de enriquecimiento de esta, como por ejemplo dotar a las imágenes de información geoespacial, se opta finalmente por centrar el proyecto solo en los aspectos textuales.

Uno de los objetivos de esta memoria es determinar la idoneidad del uso de tecnologías de almacenamiento, indexación y búsqueda para proyectos en Humanidades Digitales y ver cómo estas tecnologías son más adecuadas por diversos factores como pueden ser la eficiencia, escalabilidad, tolerancia a fallos… y en general mostrar que efectivamente las bases de diseño de tecnologías como SOLR son mejores para la mayoría de los proyectos en Humanidades Digitales con requisitos similares al proyecto desarrollado en esta memoria.

Para validar esta afirmación, se ha considerado muy oportuno el diseño y la implementación de varias aproximaciones, incluyéndose al final una discusión sobre la viabilidad de este tipo de proyectos con las tecnologías actuales.

3.1 La base de datos relacional ARQUEOGRIEGOS_SQL

Para implementar la base de datos para los documentos de la región de la Antigua Grecia llamada Ática, adaptándolos para poder ser importados en las tablas correspondientes, y habilitándose el subdominio https://arqueogriegos.3isi.com para la realización de pruebas, se ha procedido como sigue:

- Diseño de la estructura de la base de datos, atendiendo a la distribución de los documentos entregados por el humanista sobre la región Ática se puede observar una estructura bien definida en cada uno de los documentos de Word que hay por yacimiento. Esta estructura está compuesta por los campos que podemos ver en la tabla a continuación,

ID	COD	REGION	ARCHIVO	INTRO	ACCESO	HISTORIA	MITOLOGIA	YACIMIENTO	MUSEO
1	0	ATICA	GEOGRAFIA DEL ATICA	X					
2	00	ATICA	LOS 11 REYES MITICOS DEL ATICA	X					
3	1	ATICA	ELEUSIS	X	X	X	X	X	X
4	2	ATICA	MEGARA	X	X	X	X	X	X
5	3	ATICA	HERAION DE PERACHORA	X	X	X		X	
6	4	ATICA	AIGOSTHENA	X	X	X	X	X	
7	5	ATICA	ELEUTERAS	X	X	X	X	X	
8	6	ATICA	TEMPLO DE APOLO ZOSTER	X	X	X	X	X	
9	7	ATICA	SOUNION	X	X	X	X	X	
10	8	ATICA	MUSEO DEL LAVRIO	X	X				X
11	9	ATICA	THORIKOS	X	X	X	X	X	
12	10	ATICA	VRAUNION	X	X	X	X	X	X
13	11	ATICA	ICARION	X	X	X	X	X	
14	12	ATICA	MARATON	X	X	X	X	X	X
15	13	ATICA	RAMNOUS	X	X	X	X	X	
16	14	ATICA	ANFIAREIO	X	X	X	X	X	

Tabla 8 - Estructura de los documentos de Ática

Se puede comprobar como los documentos de textos en formato .doc tienen una serie de apartados claramente definidos y con suficiente uniformidad como para determinar que estos campos podrán ser llevados a la estructura de la base de datos.

Así se creará una Relación llamada Yacimientos en la que se crearán campos de identificación como ID, CODIGO, REGION, ARCHIVO y otros campos como INTRO, ACCESO, HISTORIA, MITOLOGIA, YACIMIENTO y MUSEO que serán los que contendrán la información textual como tal.

Además de esta relación se crean otras tres relaciones llamadas FOTOS, PLANOS y DOCS que contendrán las rutas de acceso de los demás recursos aportados por el humanista como son los archivos de imagen que son fotografías obtenidas desde cámaras fotográficas, planos digitalizados en formato .jpg y documentos de otro tipo también digitalizados a formato .jpg

- Formato de los datos a insertar en las relaciones, los datos aportados por el humanista necesitan ser adaptados para poder ser insertados en las relaciones de la base de datos relacional que hemos creado para tal efecto. Así, los documentos .doc podemos convertirlos a texto plano, o usar los que ya convertirnos a .xml para insertarlos de manera nativa a SOLR. Hay que poner atención al formato del tipo de codificación para que al insertarlos en la base de datos los textos se mantengan de manera íntegra y no se produzcan sustituciones de unos caracteres por otros, seguidamente se recopila toda la información en un archivo común y distribuirlo en algún formato de archivo como .csv que permite importar los datos a la base de datos (se ha creado el archivo arqueogriegos-sql.csv a tal efecto). En el caso de las imágenes .jpg de las fotografías, planos y .docs convendrá adaptar los nombres de los archivos para que sean accesibles desde enlaces de hipervínculo.

Para habilitar el acceso a la base de datos en primer lugar necesitamos una plataforma que la almacene y posteriormente nos permita acceder a los datos contenidos en ella, para ello se han habilitado dos opciones que se detallan a continuación:

- Servidor local, mediante la aplicación XAMPP basada en tecnología AMP (Apache, MySQL y PHP) del lado del servidor podremos disponer del motor de bases de datos MySQL y crear la base de datos. Posteriormente, accedemos a la utilidad PHPMYADMIN y mediante la ejecución del siguiente script:

```
USE arqueogriegos_sql;
-- Eliminar cualquier dato previo de la tabla yacimientos
TRUNCATE TABLE yacimientos;
-- Cargar los datos del archivo CSV en la tabla yacimientos
LOAD DATA INFILE 'C:/datos/arqueogriegos-sql.txt'
INTO TABLE yacimientos
FIELDS TERMINATED BY ';'
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

Se insertarán todos los datos de texto en la relación yacimientos para posteriormente poder realizar consultas sobre ellos.

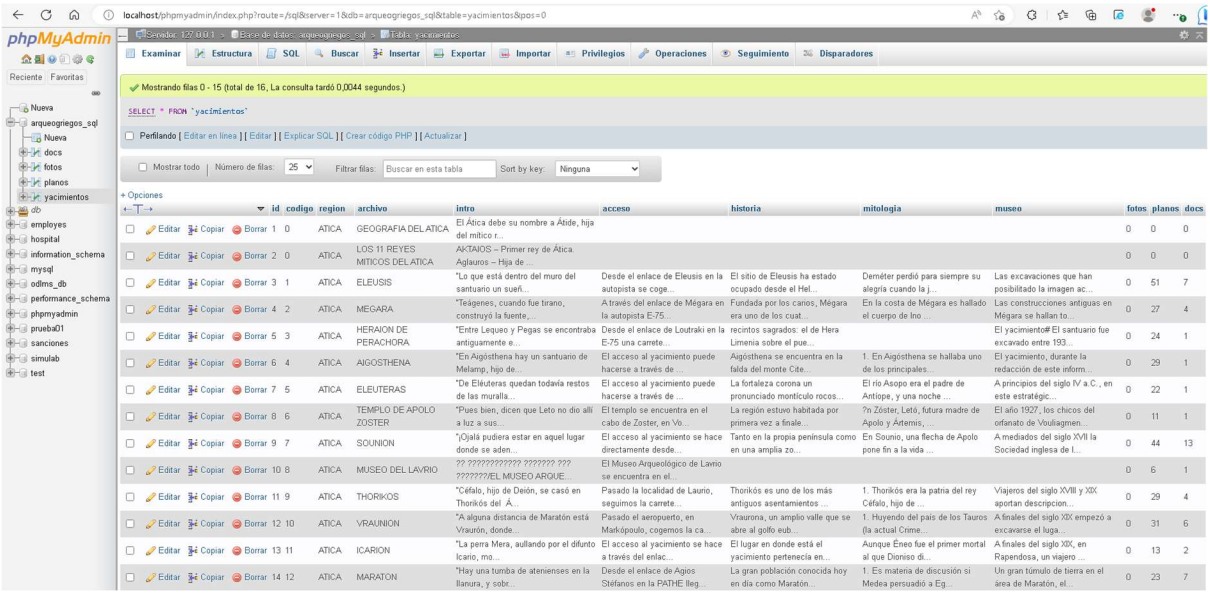


Figura 2 - Vista de la Relación YACIMIENTOS en servidor local

- Servidor remoto, se ha habilitado el subdominio https://arqueogriegos.3isi.com para la realización de pruebas de rendimiento en un entorno de producción.

3.2 Prototipo Apache SOLR

Con la revisión de las tecnologías de almacenamiento y acceso a la información que se ha llevado a cabo en el apartado anterior de esta memoria, se decide crear el diseño del modelo con tecnologías de indexación como SOLR o ELASTICSEARCH, siendo SOLR el elegido, esgrimiendo como principales argumentos los que relacionamos a continuación:

- Tecnología más orientada a textos, ELASTICSEARCH está más orientado a búsquedas analíticas
- Mayor capacidad para procesar grandes cantidades de datos
- Comunidad muy amplia de código abierto
- Código libre gratuito, ELASTICSEARCH tiene módulos de pago y cabe la posibilidad que el proyecto por completo pase a ser propietario

Una vez seleccionada la tecnología a usar conviene destacar que uno de los objetivos principales de esta memoria es la de mostrar si se obtienen mejoras sustanciales usando este tipo de tecnología en contraposición al uso de bases de datos relacionales clásicas. Como se verá en los apartados siguientes la configuración detallada y minuciosa de SOLR puede requerir una curva de aprendizaje muy amplia para obtener los mejores resultados de eficiencia y se considera que conseguir esos resultados no entra en el objeto de esta memoria pues sería necesario un estudio mucho más amplio que el que se realizará.

Hay una primera experimentación en SOLR con la colección de archivos de la región Ática entregados por el humanista tal y cual, es decir sin ningún tipo de modificación ni preprocesamiento, con tres aproximaciones.

3.3.1. Instalación de SOLR

Para poder crear el diseño del modelo de este proyecto lo primero es proceder a la instalación de SOLR, para ello descargamos la versión más actual publicada en la página web oficial https://solr.apache.org a la fecha de la redacción de esta memoria se descarga la versión 9.2. y en concreto la versión Binary release solr-9.2.tgz

En este caso la instalación se realizará en un ordenador con sistema operativo Windows 10, para proceder a su instalación se procederá a descomprimir el archivo descargado en una ubicación concreta, en nuestro caso en C:\solr.

Como se indica en la completísima página web de guía de referencia de Apache SOLR https://solr.apache.org/guide/solr/latest/index.html [16]. La selección de la plataforma seleccionada es una decisión importante, puesto que algunos de los comandos indicados en esta guía de referencia serán distintos para sistemas operativos WINDOWS. La decisión de

este proyecto de seleccionar WINDOWS está motivada por temas de aplicación del proyecto a entornos ya operativos a los que se podrían aplicar lo que se verá en esta memoria, y quizá para otros escenarios lo más recomendable seria instalar SOLR en sistemas LINUX. Es necesario disponer de JAVA instalado en el equipo en el que se quiere lanzar SOLR.

Figura 3 - Versiones de JAVA instaladas

En la figura a continuación se puede ver la distribución de carpetas de la distribución que se ha descargado para acceder a SOLR.

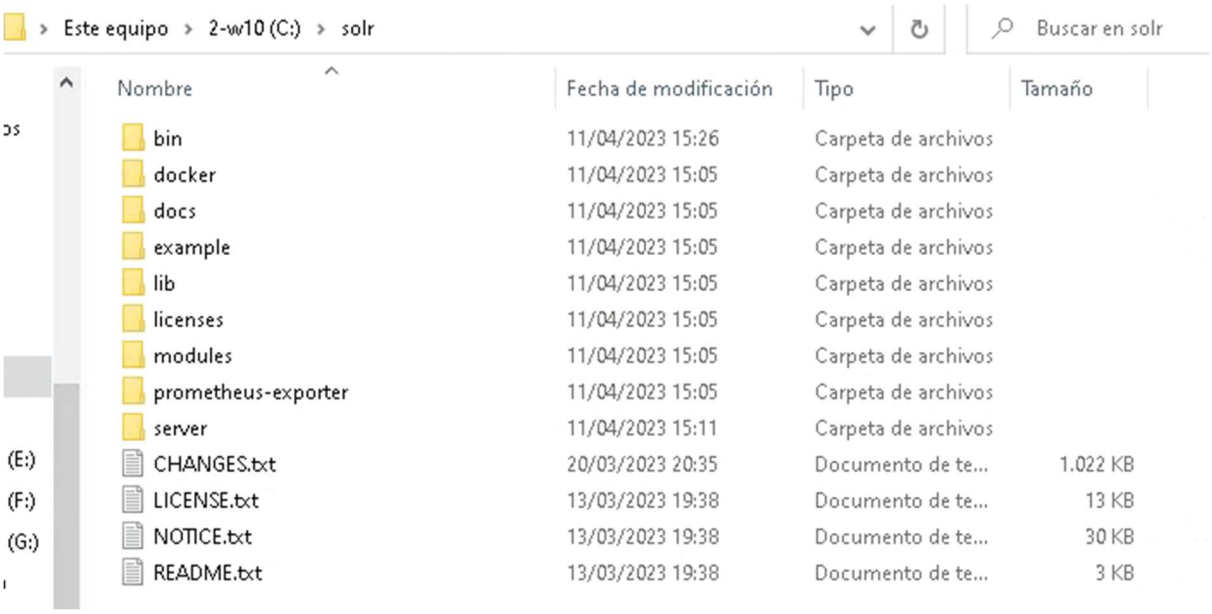


Figura 4 - Listado de directorios de SOLR 9.2

Para lanzar SOLR de manera inicial abrimos un terminal de consola, nos situamos en el directorio **C:\solr\bin** y procedemos a ejecutar el comando **solr start**

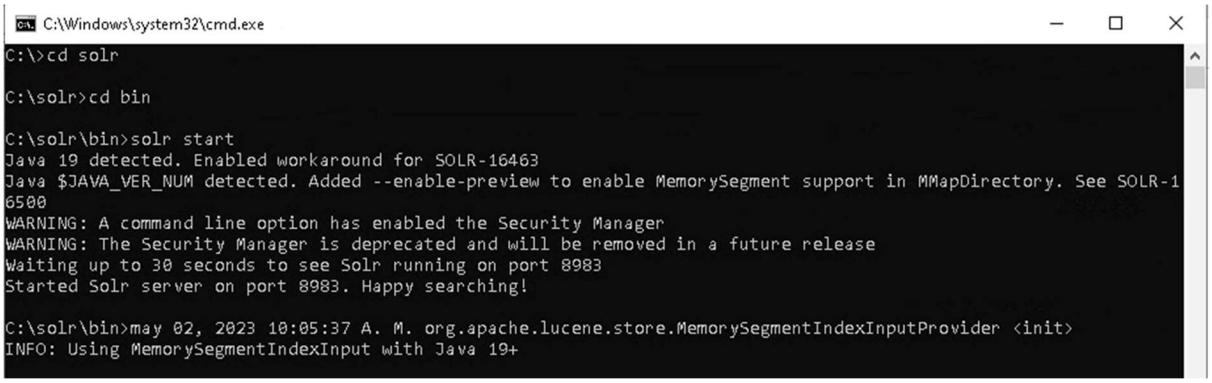


Figura 5 - Ejecución de SOLR

A continuación, podemos acceder a la dirección de gestión de SOLR accediendo desde un navegador web en el mismo ordenador donde se ha ejecutado, para ello introducimos la dirección http://localhost:8983

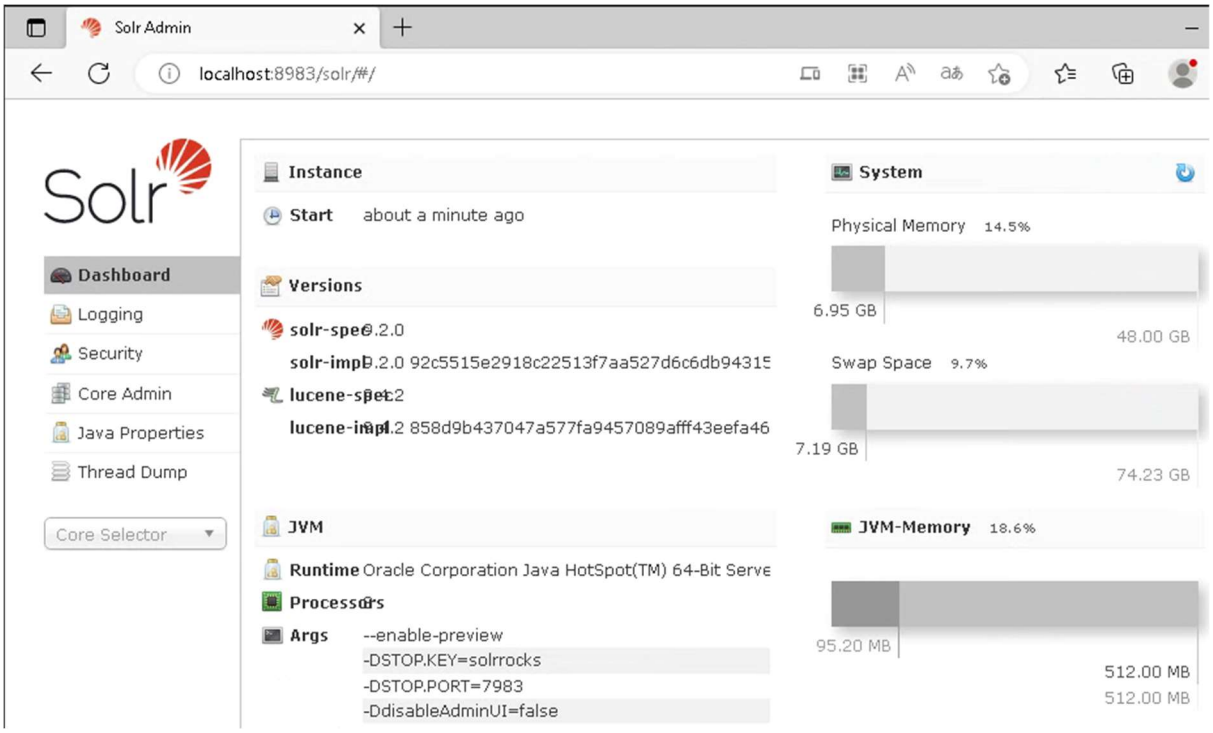


Figura 6 - SOLR Web Admin UI

Para facilitar el acceso al servidor se ha habilitado una dirección online y mediante configuración especifica en el router al que está conectado el ordenador usado para este proyecto, se accede a este servidor mediante https://solr.dyanalias.com:8983.

Para conseguirlo es necesario editar el archivo solr.in.cmd ubicado en C:\solr\bin modificando el siguiente código:

```
REM Solr to accept connections on all network interfaces
REM set SOLR_JETTY_HOST=127.0.0.0
```

A este:

```
REM Solr to accept connections on all network interfaces
set SOLR_JETTY_HOST=0.0.0.0
```

También podemos cambiar el puerto por defecto 8983 a otro de nuestra elección para añadir más seguridad habilitando ese otro puerto no conocido, para ello en el mismo archivo editamos el código siguiente:

```
REM Sets the port Solr binds to, default is 8983
REM set SOLR_PORT=8983
```

A este:

```
REM Sets the port Solr binds to, default is 8983
set SOLR_PORT=8097
```

3.3. Creación de Colecciones para experimentación

Para iniciar el diseño del modelo a implementar es necesario conocer varios conceptos básicos sobre el funcionamiento interno de SOLR, así como las funcionalidades y características de sus componentes.

En primer lugar, entendemos por CLÚSTER al grupo de servidores (Nodos) que ejecutan instancias de SOLR y existen dos modos principales de operar un Clúster.

Antes de conocer el funcionamiento de estos dos modos, conviene conocer una serie de conceptos como son:

- SHARD, son las divisiones, los fragmentos, que se pueden hacer a un índice lógico, cada una de estas divisiones contiene, por lo tanto, un subconjunto del índice general.

- REPLICA, para proporcionar opciones de respaldo ante errores cada Shard (fragmento) se puede copiar como una réplica, incluso para índices que no están divididos.

- LEADER, cuando se han creado réplicas de los fragmentos de un índice hay que seleccionar uno de ellos con el líder, y en ellos cae la responsabilidad de ser una fuente de verdad del índice completo. En contraposición, el resto de los fragmentos se denominarán Followers (seguidores).

- CORE, cada una de las réplicas se denomina también Core (núcleo) y podemos definir un Core como un índice individual, con sus archivos de configuración asociados, que incluyen solrcofig. xml y los archivos Schema.

Con estos conceptos claros, podemos describir de manera breve los dos modos de operación antes mencionados:

- SOLRCLOUD, este modo utiliza Apache ZooKeeper para proporcionar la gestión centralizada de clústeres que es su función principal. ZooKeeper rastrea cada nodo del clúster y el estado de cada núcleo en cada nodo.

En este modo, los archivos de configuración se almacenan en ZooKeeper y no en el sistema de archivos de cada nodo. Todas las colecciones comparten las mismas configuraciones. Esta es una centralización adicional de la gestión de clústeres, ya que las operaciones se pueden realizar en toda la colección a la vez. Cuando se realizan cambios en las configuraciones, un solo comando para recargar la colección recargaría automáticamente cada núcleo individual que es miembro de la colección.

ZooKeeper también maneja el equilibrio de carga y la conmutación por error. Las solicitudes entrantes, ya sea para indexar documentos o para consultas de usuarios, se pueden enviar a cualquier nodo del clúster y ZooKeeper enrutará la solicitud a una réplica apropiada de cada fragmento. En SolrCloud, el líder es flexible, con mecanismos incorporados para la elección automática del líder en caso de falla en el líder.

- USER-MANAGED, este modo requiere que las actividades de coordinación de clústeres para las que SolrCloud normalmente usa ZooKeeper se realicen manualmente o con scripts locales. Si el corpus de documentos es demasiado grande para un índice de un solo fragmento, la lógica para crear fragmentos se deja completamente al usuario. No hay formas automatizadas o programáticas para que Solr cree fragmentos durante la indexación.

En el modo administrado por el usuario, el concepto de líder y seguidor se vuelve crítico. La identificación de qué nodo albergará la réplica principal y qué host serán réplicas determina cómo se configura cada nodo. En este modo, todas las actualizaciones de índice se envían solo al líder. Si el líder falla, no hay un mecanismo de conmutación por error incorporado. Una réplica podría seguir atendiendo consultas si las consultas se dirigieran específicamente a ella. Cambiar una réplica para que sirva como líder requeriría cambiar solrconfig.xml y las configuraciones en todas las réplicas y recargar cada núcleo.

Una de las primeras decisiones a tomar en el diseño de nuestro modelo es determinar el modo en el que ejecutaremos el Cluster de nuestro catálogo, para ello hay que considerar que si bien nuestro catálogo se puede hacer muy grande, inicialmente con el fin de validar si SOLR resulta óptimo para nuestro modelo, el catálogo solo será una pequeña parte del total por lo tanto usaremos el Modo Cloud para que el sistema sea responsable de la organización de los Cores que se generen para la creación de los índices a partir de los documentos que se carguen en ellos.

No es objeto de esta memoria especificar los detalles de la arquitectura de SOLR, pero de cara a la creación de las colecciones que usaremos para experimentar con los tres maneras de crear colecciones para posteriormente insertar los documentos en ellas mostramos el Diagrama 1 - Arquitectura General de Apache SOLR [17]

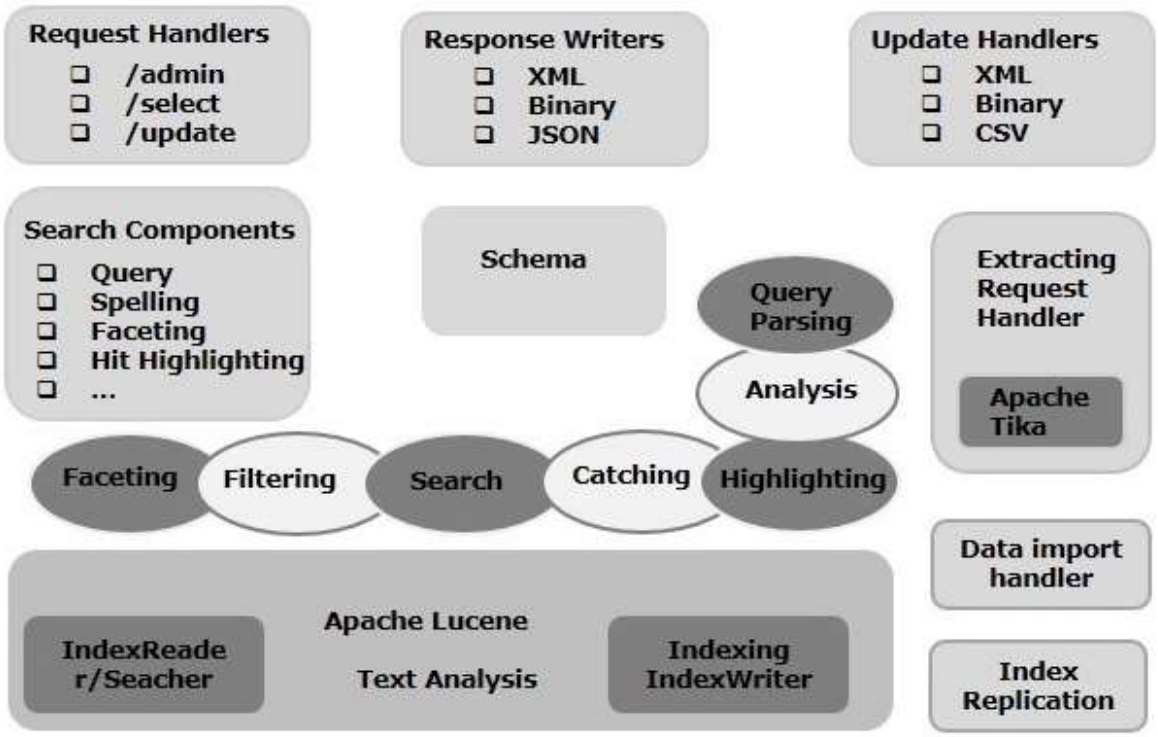


Diagrama 1 - Arquitectura General de Apache SOLR

Podemos ver algunos de los componentes que nos servirán para crear las colecciones con los tres métodos que se describirán, uno de ellos será mediante Apache TIKA que se incluye en el componente *Extracting Request Handler* y los otros dos métodos que requerirán el componente *Data Import Handler* para la gestión de la importación de los documentos a las colecciones.

A continuación. se detallará la creación de las tres colecciones siguiendo distintos métodos, en cada uno de ellos se incluirá un diagrama en el que se resumen los procesos a realizar tanto por el usuario como por el propio SOLR.

COLECCIÓN 1 - COLLECTION ARQUEOGRIEGOS-DOC

Para esta primera experimentación con una colección en SOLR se procederá a insertar los archivos de la región Ática entregados por el humanista tal y cual, es decir sin ningún tipo de modificación ni preprocesamiento. Estos archivos son de tipo .doc, .jpg y .dwg, es decir texto, imagen y CAD, y para ello usaremos el kit de herramientas Apache TIKA que viene cargado en la distribución que hemos instalado de SOLR.

Paso 1, creación de la colección

Si los documentos que necesita indexar están en formato binario, como Word, Excel, PDF, etc., Solr incluye un controlador de solicitudes que utiliza Apache Tika para extraer texto e indexarlo en Solr. Tika funciona internamente sintetizando un documento XHTML a partir del contenido central del documento analizado y crea uno o más campos de texto a partir del contenido. También exporta los metadatos del documento (aparte del XHTML). Tika produce metadatos como Título, Tema y Autor de acuerdo con especificaciones como DublinCore. Los metadatos disponibles dependen en gran medida de los tipos de archivos y de lo que a su vez contienen. Algunos de los metadatos generales creados por Tika, se describen a continuación.

Podemos afirmar que crear colecciones con este framework nos facilitaría la inserción de los documentos a indexar, pero posiblemente no sea la manera óptima por los efectos colaterales que puedan aparecer por el formato y contenido de los archivos binarios que carguemos a la colección.

Vamos a proceder a la creación de una colección que denominaremos ARQUEOGRIEGOS-DOC a partir de los archivos originales entregados por el humanista, centrándonos únicamente en los archivos de tipo .doc y .jpg.

Desde un terminal de consola CMD y en la ubicación C:\solr\bin ejecutamos el comando siguiente: solr start -e schemaless -Dsolr.modules=extraction.

Tras unos instantes obtenemos la siguiente salida por la consola, confirmando la iniciación correcta del servidor:

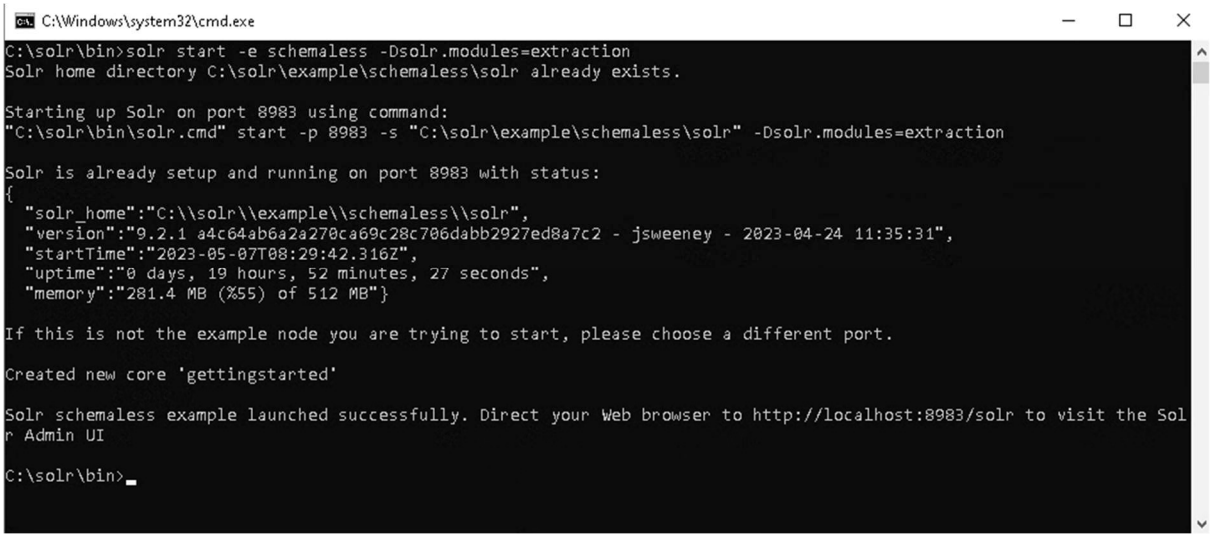


Figura 7 - Creación de la Colección gettingstarted

De esta manera crearemos la colección que SOLR carga a modo de ejemplo y que por defecto se llama *gettingstarted*, si accedemos al cliente web de SOLR procedemos a cambiar el nombre a ARQUEOGRIEGOS-DOC. Esta acción la podemos hacer desde el cliente web de SOLR (Solr Admin UI), desde el menú Core Admin y la opción Rename.

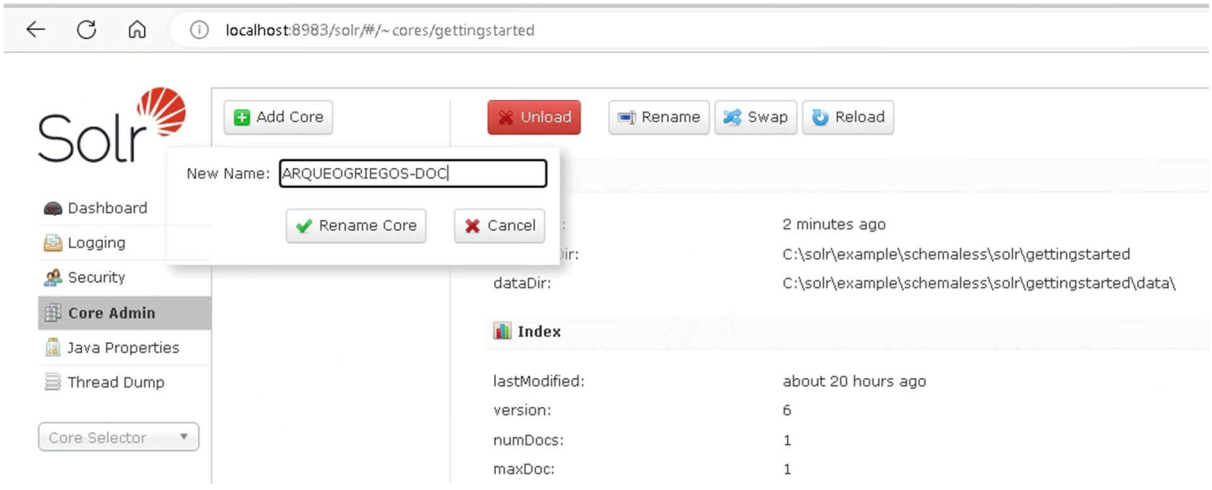


Figura 8 - Colección ARQUEOGRIEGOS-DOC

A continuación, ejecutamos el comando CURL siguiente:

```
curl -X POST -H "Content-type:application/json" ^
-d "{ "add-requesthandler": { "name": '/update/extract', "class":
'solr.extraction.ExtractingRequestHandler', "defaults": {
"lowernames": "true", "captureAttr":"true"} } }" ^
 "http://localhost:8983/solr/ARQUEOGRIEGOS-DOC/config"
```

En este caso se habilita el módulo de extracción de TIKA que será el responsable de obtener la información de los archivos binarios que carguemos a la colección, se agrega al a colección creada el controlador update/extract para habilitar Solr Cell.

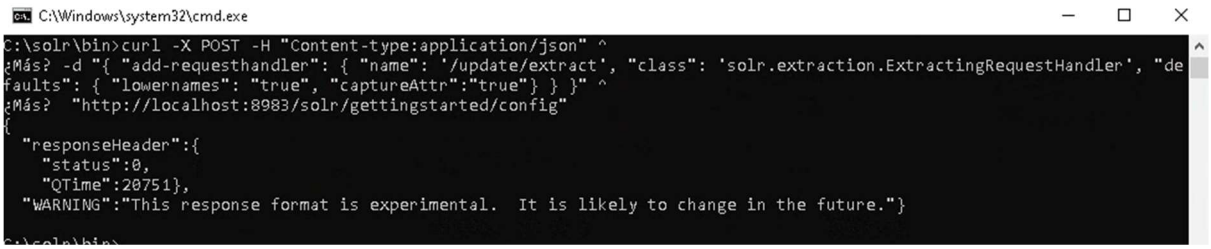


Figura 9 - Colección ARQUEOGRIEGOS-DOC con módulo de extracción TIKA cargado

Paso 2, selección del modo de esquema

Como se ha indicado con anterioridad con este método de creación de colecciones e inserción de documentos no se requiere la creación de un esquema predefinido en el que se definan y se creen los campos personalizados para estructurar la información contenida en los documentos que se insertan. En la figura siguiente vamos a ver la estructura que en la que se guardan los campos por defecto de un documento .doc que se ha agregado a la colección ARQUEOGRIEGOS-DOC. Para ello accedemos al cliente web de SOLR y nos dirigimos al menú QUERY dentro de la colección, en el campo q del formulario de consulta introducimos la cadena de caracteres *:* con lo que nos cargara todos los documentos de la colección, inicialmente solo carga los 10 primeros, ya que los parámetros starts y rows tienen los valores 0 y 10 respectivamente, por ultimo pulsamos el botón de la parte inferior con el texto EXECUTE QUERY

Se observa que hay multitud de campos que se han creado automáticamente, algunos de estos campos son meta, p, a, id, cp_revision, date, company… y muchos de ellos son datos propios del archivo .doc. Un campo importante es content pues el que realmente tiene la información textual de cada uno de los archivos añadidos a la colección.

Vamos a limitar la consulta a un documento en concreto, por ejemplo, el nombre de uno de los yacimientos como puede ser MEGARA, en el momento de la inserción en la colección se puso el id 4, y como se siguió un orden de inserción, podemos hacer la consulta insertando en el campo q la cadena de texto id:doc4 y pulsando el botón Execute Query, obtenemos este resultado:

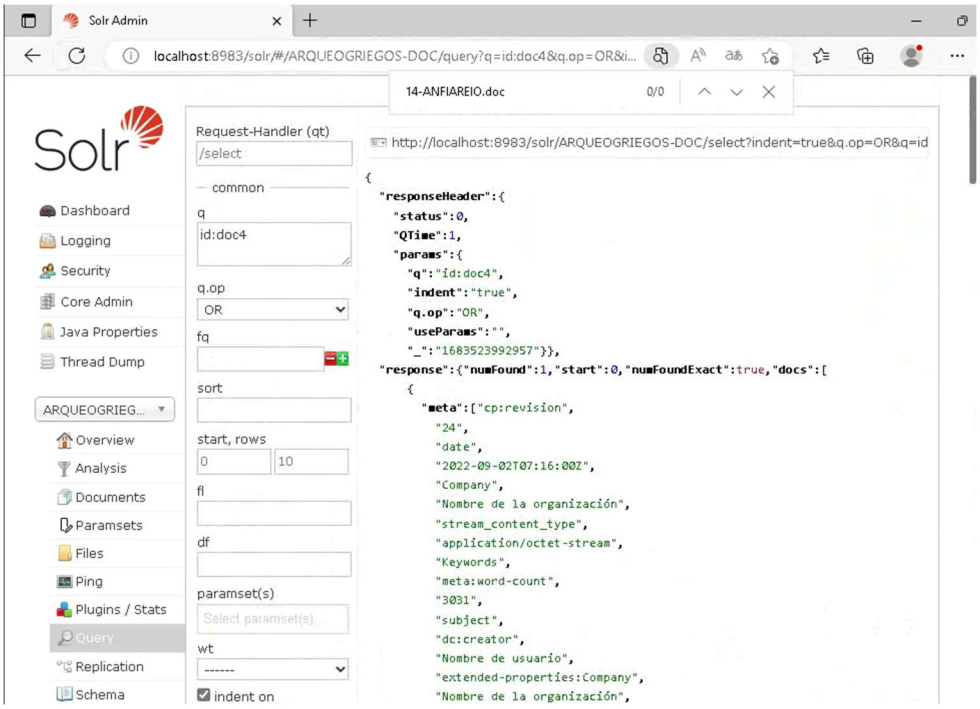


Figura 10 - Consulta mostrando el Documento con id 4

Que el *framework* Apache TIKA haga la tarea automática de reconocer los campos a la hora de insertar los documentos puede tener ventajas e inconvenientes, pueden darse proyectos en los que almacenar información sobre los propios archivos no sea necesario y para otros proyectos en cambio sea un aspecto vital.

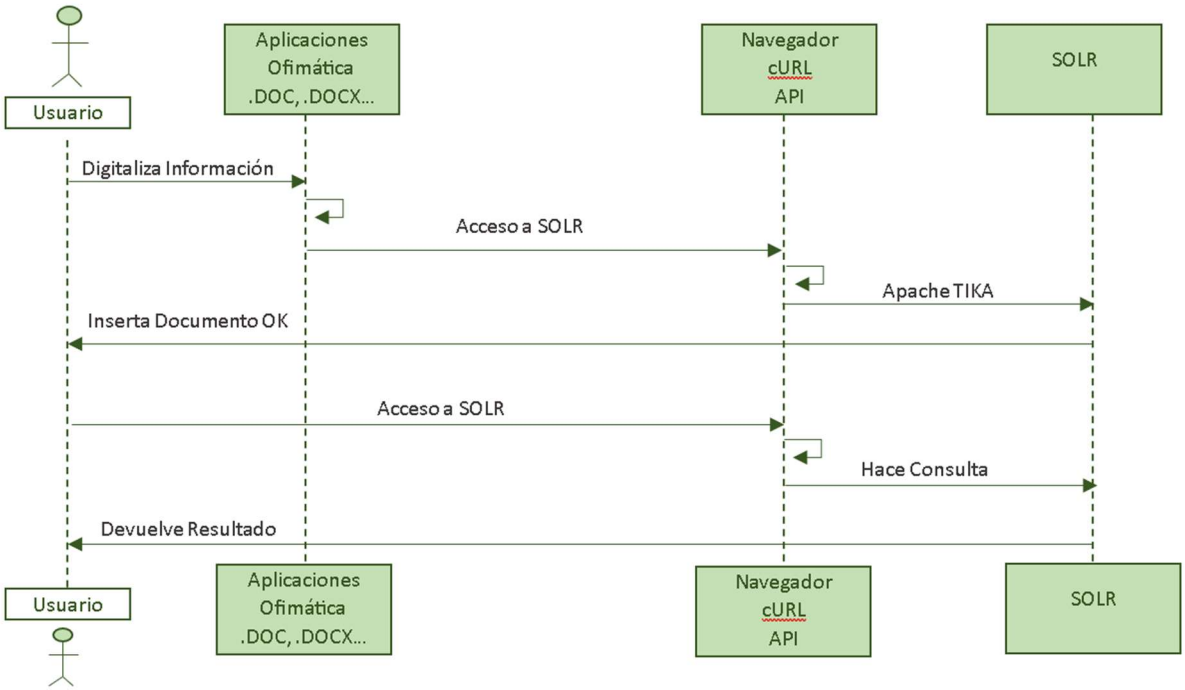


Diagrama 2 - Inserción Documentos en Colección mediante Apache TIKA

El diagrama ofrece una secuencia muy reducida de todas las opciones y elementos que implementa Apache TIKA, el objeto de simplificar el diagrama (y los que se mostrarán en los otros dos métodos de inserción de documentación) para es ocultar al lector detalles que no aportan información sobre este proyecto, para ampliar esta información se puede consultar la guía de referencia [16] de SOLR en la que la información está muy completa y detallada.

También hay que destacar que las funcionalidades de Apache TIKA suponen una facilidad para los usuarios que no tengan un amplio conocimiento de SOLR y quieran evitar la especificación de un esquema predefinido y creado a medida del proyecto. Luego, la decisión del diseño de una aplicación que usará SOLR debe tener claros los objetivos para establecer si usar el framework Apache TIKA o no.

COLECCIÓN 2 - COLLECTION ARQUEOGRIEGOS-XML

Para esta segunda experimentación con la colección se insertarán los archivos de la región Ática entregados por el humanista, pero con una serie de modificaciones con respecto a la primera aproximación:

- Para los archivos de texto .doc se ha procedido a transformarlos con la aplicación Microsoft Word a formato .xml.
- Para los archivos de imagen .jpg se han añadido referencias en el nombre del archivo al yacimiento del que es cada una de las imágenes.
- Para los archivos de CAD no se ha realizado ninguna modificación, puesto que no es objeto de estas experimentaciones, aunque quizá sería conveniente convertirlas a .jpg y tratarlas como a las .jpg anteriores.

A continuación, se detalla el proceso para la creación de la colección que llamaremos ARQUEOGRIEGOS-XML y las posteriores acciones para cargar los archivos a ella.

Paso 1, creación de la colección

Podemos hacerlo mediante la línea de comando usando este comando desde una consola de terminal en C:\solr\bin

```
solr start -c ARQUEOGRIEGOS-XML -s 2 -rf 2
```

Donde el parámetro -c indica que iniciamos el servidor en Modo Cloud, el parámetro -s 2 indica que crearemos 2 Shards (fragmentos) para la colección y por último el parámetro -rf 2 indica que crearemos un factor de réplica de 2. Con esto la colección quedará creada y en sucesivas ejecuciones de SOLR solo necesitaremos lanzar el servidor y la colección ya estará disponible. Se accede mediante el comando: solr start -c

También podemos crear la colección accediendo al Cliente de SOLR (interfaz web) en la dirección local http://locahost:8983 si nos encontramos trabajando en el ordenador local, o a https://solr.dyanalias.com:8983, habilitada para este proyecto si nos encontramos en un ordenador remoto.

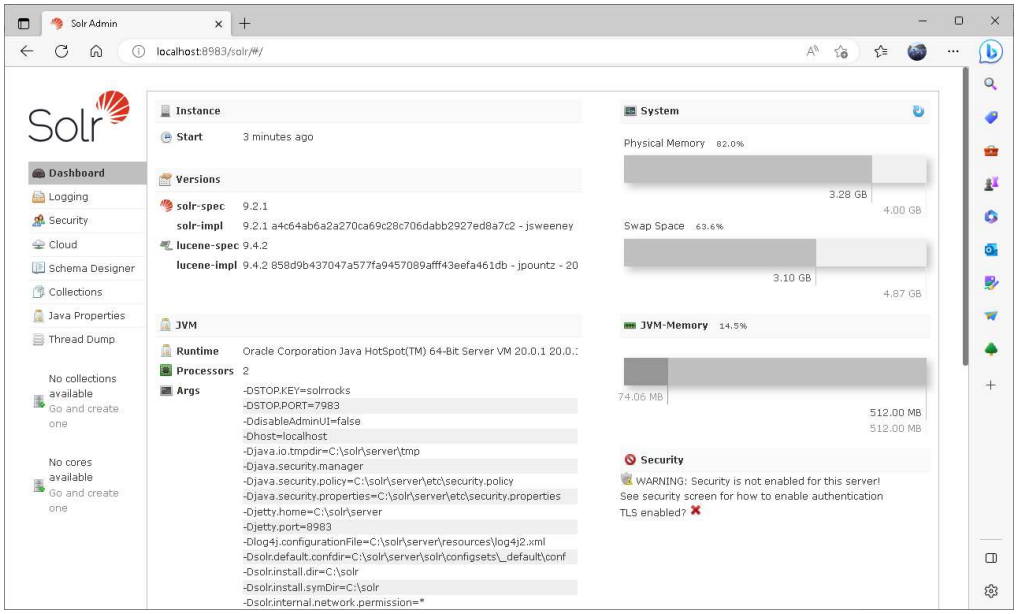


Figura 11 - Página de Inicio del Cliente SOLR Interfaz Web

Al ejecutar el servidor en Modo Cloud tendremos acceso al apartado COLLECTIONS y veremos las opciones de Add Collection.

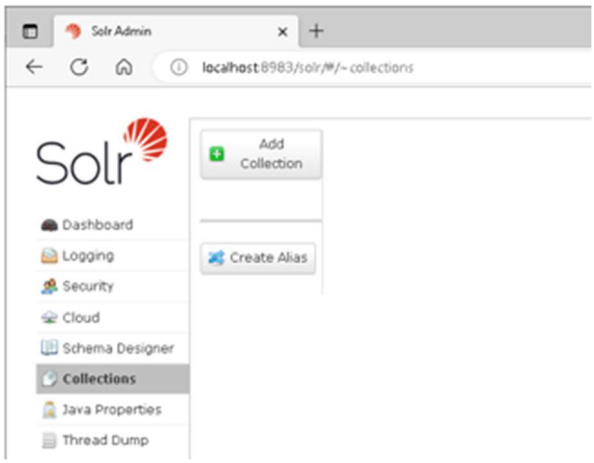


Figura 12 - Crear una Colección

En el formulario de creación de la Colección indicamos los parámetros de nombre, configuración, número de Shards y el factor de replicación.

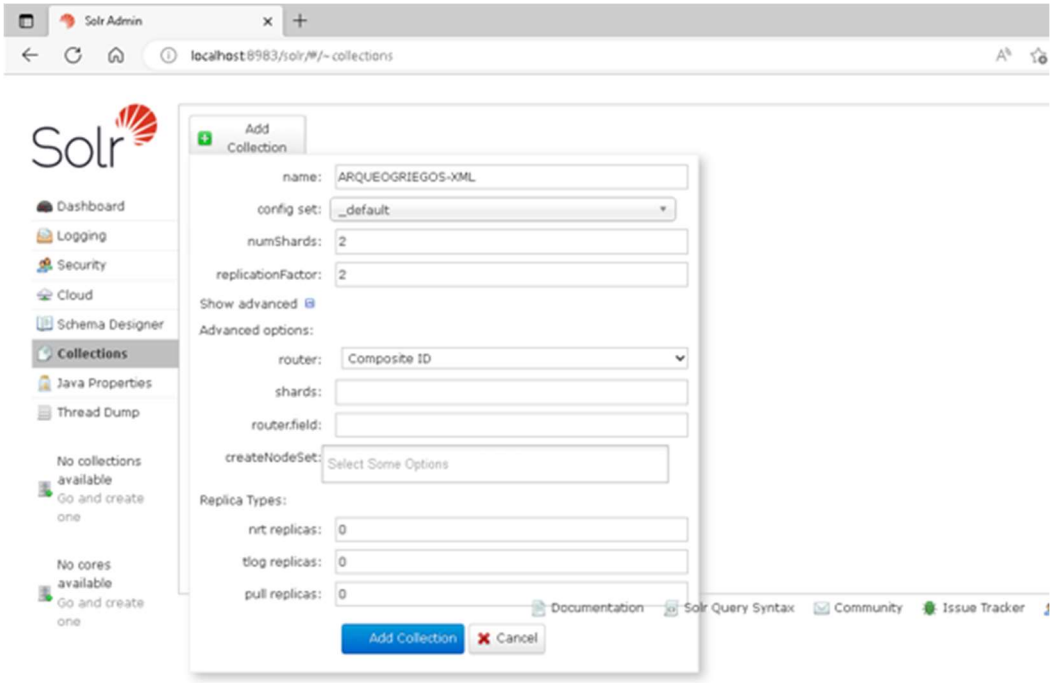


Figura 13 - Parámetros de la Colección a crear

Una vez creada la Colección podemos acceder a ella cargándola desde el cliente de SOLR y así tendremos acceso a sus opciones.

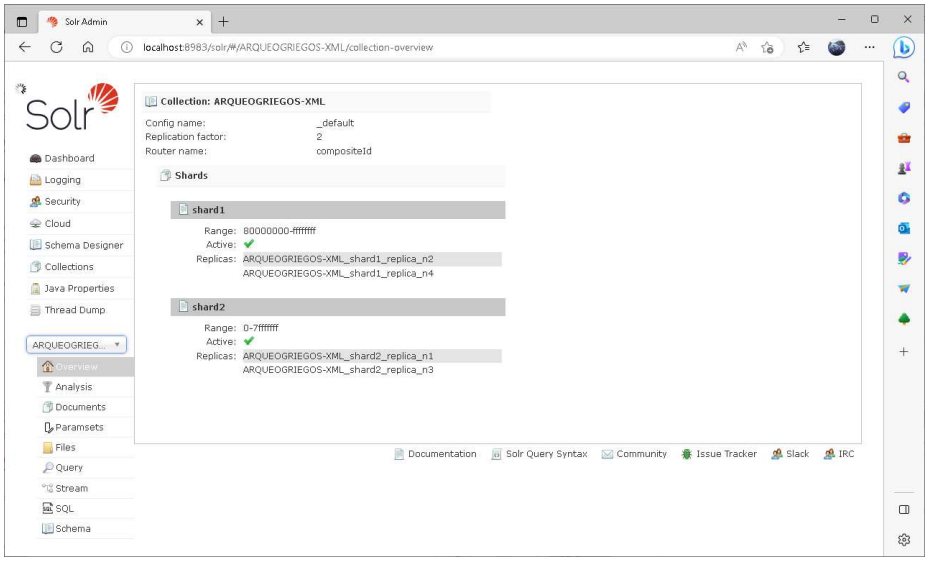


Figura 14 - Nueva Colección creada

Paso 2, selección del modo de esquema

Una decisión muy importante a la hora de administrar la colección creada es la de usar un esquema creado específicamente o usar el Modo Sin Esquema para que SOLR administre

los campos en los que se almacenaran los datos de los documentos que carguemos en la colección. Las implicaciones de esta elección son varias puesto que, si decidimos crear un esquema personalizado, en primer lugar, hay que definir estos campos indicando los tipos de campo para cada uno de ellos y posteriormente habrá que editar los documentos fuente para clasificar los datos de esos documentos en los archivos .XML que serán cargados en la colección.

En el Modo Sin Esquema el servidor SOLR tiene los mecanismos necesarios para adivinar los tipos de campos de la información contenida en los archivos que cargamos a la colección, este factor justifica que para esta experimentación se use este modo puesto que los datos contenidos en los documentos aportados por el humanista contienen información de todo tipo y quizá la creación del esquema personalizado sea una tarea inabordable, especialmente teniendo en cuenta que el modo sin esquema pueda conseguir los mismos objetivos.

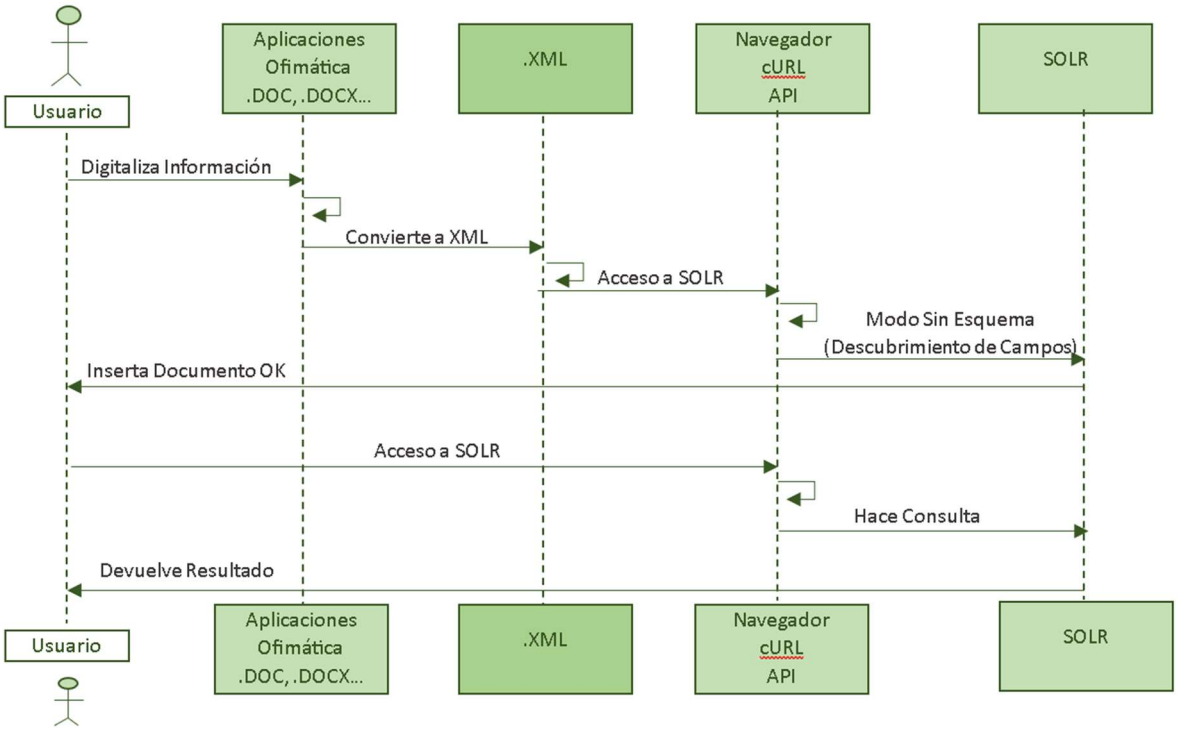


Diagrama 3 - Inserción Documentos en Colección Modo Sin Esquema

Para comprobar que estamos usando el modo sin esquema con la configuración de los campos que se crean por defecto podemos ejecutar el siguiente comando sobre la colección que hemos creado anteriormente:

curl -X GET http://localhost:8983/api/collections/ARQUEOGRIEGOS-xml/schema/fields

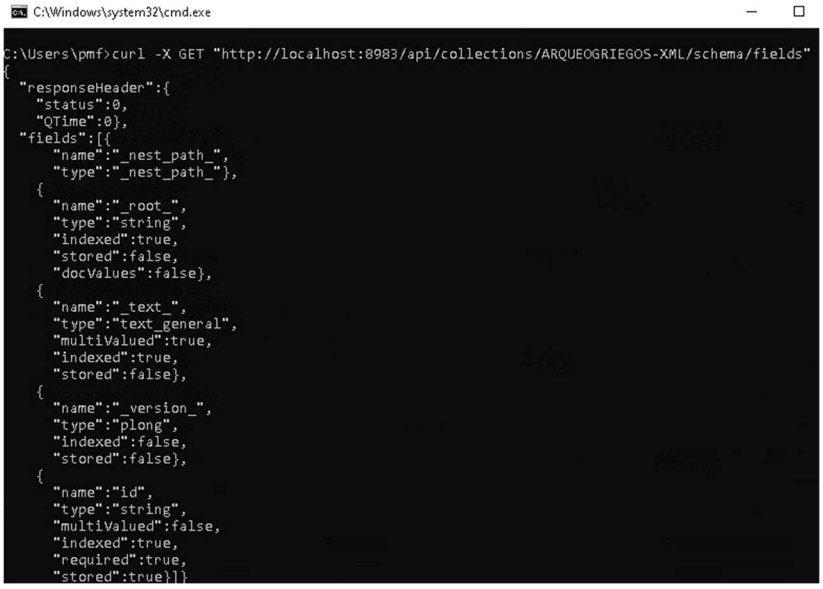


Figura 15 - Información del esquema por defecto usado en el Modo Sin Esquema

COLECCIÓN 3 - COLLECTION ARQUEOGRIEGOS-ESQ

En la tercera experimentación se procederá a crear un esquema personalizado para la colección que llamaremos ARQUEGRIEGOS-ESQ, para ello podemos usar dos métodos de creación y/o modificaciones de los tipos y campos que conforman los esquemas de las colecciones en SOLR, estos dos métodos son:

- Diseñador de Esquemas, desde la interfaz web de SOLR podemos diseñar de manera interactiva el esquema que creamos más conveniente para la colección, ver Figura 11.
- Esquema de API, mediante una API HTTP que proporciona acceso de lectura y escritura al esquema de SOLR de cada colección (SOLR en Modo Cloud) o núcleo (si se ejecuta en Modo Usuario) mediante el uso de comando cURL o aplicaciones específicas. Más información sobre esta API en la referencia [17].

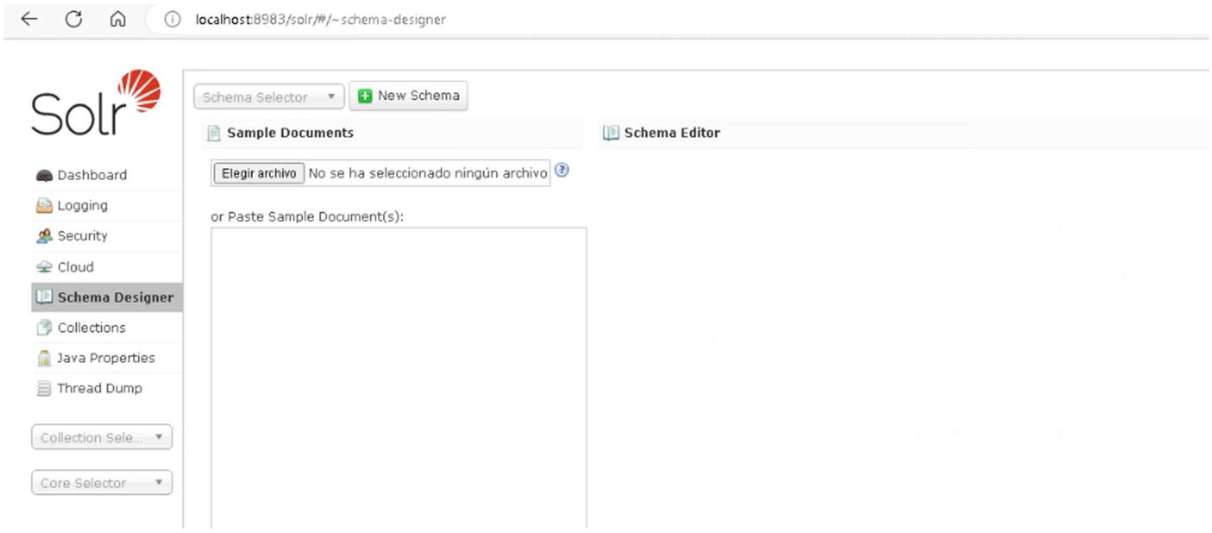


Figura 16 - Diseñador de Esquema

A continuación, se detalla el proceso para la creación de la colección mencionada.

Paso 1, creación de la colección

Podemos hacerlo mediante la línea de comando desde una consola de terminal en C:\solr\bin:

```
solr start -c ARQUEOGRIEGOS-ESX -s 2 -rf 2
```

Donde el parámetro -c indica que iniciamos el servidor en Modo Cloud, el parámetro -s 2 indica que crearemos 2 Shards (fragmentos) para la colección y por último el parámetro -rf 2 indica que crearemos un factor de réplica de 2. Con esto la colección quedará creada y en sucesivas ejecuciones de SOLR solo necesitaremos lanzar el servidor y la colección ya estará disponible, podemos acceder mediante el comando: solr start -c

También se creará, como en el caso anterior, la colección accediendo al Cliente de SOLR (interfaz web) en la dirección local http://locahost:8983 si nos encontramos trabajando en el ordenador local, o a https://solr.dyanalias.com:8983 que hemos habilitado para este proyecto si nos encontramos en un ordenador remoto.

Para la creación de esta colección es conveniente leer primero el Paso 2 que sigue a continuación pues se usará como base el Esquema que se crea en ese paso 2. Se usa el cliente web de SOLR usando como parámetro config set el esquema al crearlo lo hicimos con el nombre de sql y que al crearlo desde el cliente web se denominó .designer_sql

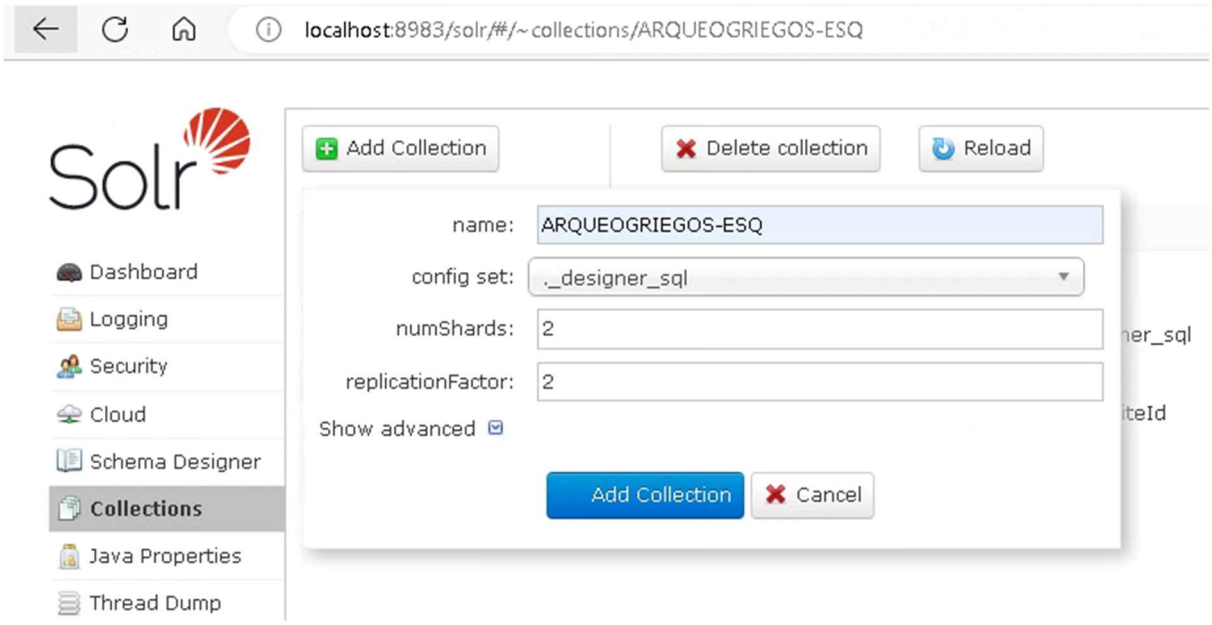


Figura 17 - Creación de la Colección ARQUEOGRIEGOS-ESQ

Paso 2, selección del modo de esquema.

Una vez decidida la creación de un esquema personalizado para una colección determinada, la siguiente decisión a tomar es la determinación de la granularidad de los documentos que se insertaran en la colección para su posterior indexación sobre la que los usuarios del sistema realizaran diversas búsquedas de información.

La determinación sobre qué debe representar un documento en su índice impulsa todo el proceso de diseño de un esquema [18] y es clave pensar en lo que los usuarios del sistema querrán encontrar en sus búsquedas. Para el caso de nuestro proyecto, como se ha podido ver en los documentos de las regiones de Ática y Tracia, los textos son grandes y se intuye una organización en apartados como introducción, acceso, historia, mitología, museo...

Esta estructuración subyacente nos proporciona una base muy definida para plantear una granularidad media a la hora de definir los campos que queremos en nuestro esquema y que resultaría muy adecuada. Con granularidad media nos referimos a que no sería una estrategia muy buena la de introducir en un único campo todo el texto de un documento (granularidad alta), al igual que tampoco lo seria definir multitud de campos para clasificar por ejemplo los elementos de características de las construcciones de los yacimientos (tipos de columnas, tipos de edificios…), pues los propios textos dan descripciones muy generales más que detalles. Si fuera al contrario nos llevaría a pensar en usar otro tipo de granularidad más baja, muy similar a como podría ser un proyecto en el que predominaran datos de tipo eminentemente numéricos, y para el que probablemente hubiéramos escogido otra tecnología de almacenamiento, indexación y búsqueda como ElasticSearch.

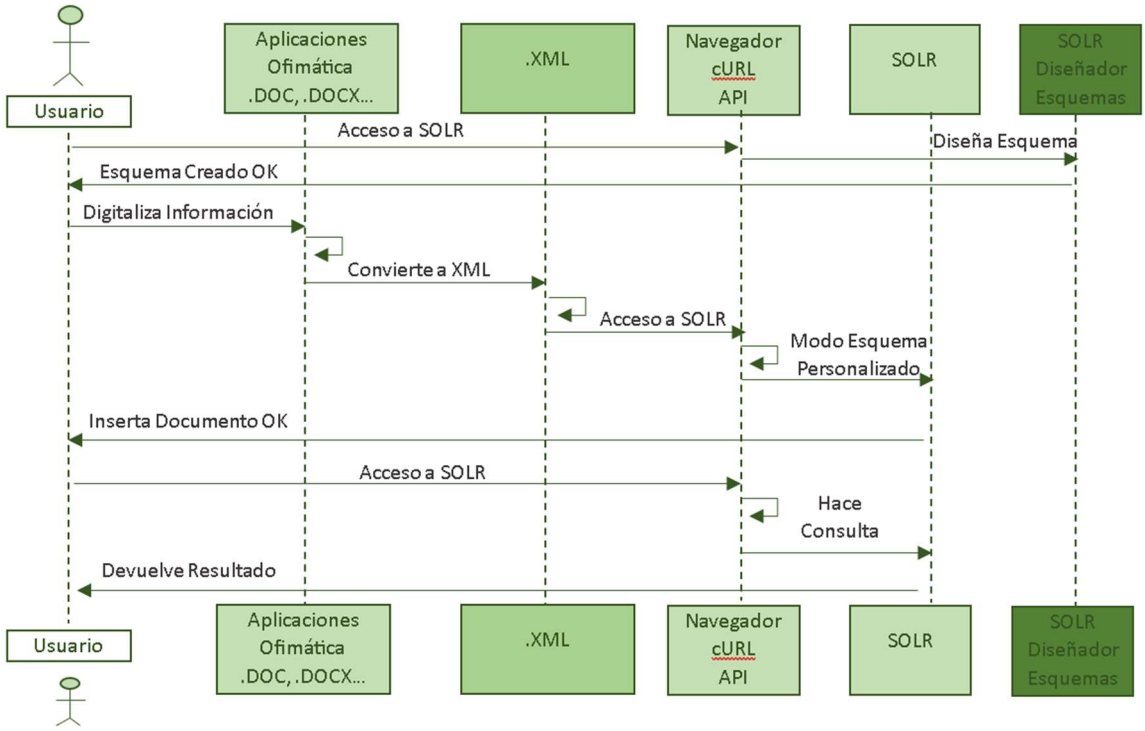


Diagrama 4 - Inserción Documentos en Colección con Método Espquema Personalizado

Será importante también en la definición del esquema diferencias que campos se indicarán como Indexados y Almacenados, los primeros son aquellos que son determinantes desde la perspectiva del proceso de búsqueda y los segundos aquellos que no, pero que igualmente son útiles para mostrar los resultados de una búsqueda.

Igualmente, importante es la definición de algunos de los parámetros de los campos como si son requeridos o no, si son multivalor, dinámicos…, en el caso de esta colección no se usarán campos de estos dos últimos tipos. Los campos que se usarán para esta colección serán tipos de campos incluidos en SOLR, que se relacionan a continuación:

- ID numerador general de la colección
- CODIGO código de la región siguiendo el formato de los documentos entregados por el humanista
- REGION nombre de la región de la Antigua Grecia
- ARCHIVO nombre original del archivo entregado por el humanista, suele referirse al nombre del yacimiento
- INTRO texto de introducción del documento
- ACCESO información sobre como es el acceso al yacimiento
- HISTORIA información recopilada sobre la historia del yacimiento
- MITOLOGIA información relacionada con elementos mitológicos del yacimiento
- YACIMIENTO información relacionada con el propio yacimiento como tal
- MUSEO información sobre el museo, si existiera, en el propio yacimiento
- FOTOS número de imágenes fotográficas disponibles para el yacimiento
- PLANOS número de imágenes de planos del yacimiento
- DOCS numero de otro tipo de imágenes del yacimiento
- LANG idioma del documento
- TIMESTAMP fecha de registro del documento

Atendiendo a la relación de campos anterior este sería el esquema para la colección ARQUEOGRIEGOS-ESQ:

```
<schema name="_sql" version="1.5">
<fields>
      <field     name="id"     type="string"     indexed="true"
stored="true" required="true"/>
      <field    name="codigo"    type="string"    indexed="true"
stored="true" required="true"/>
      <field    name="region"    type="string"    indexed="true"
stored="true" required="true"/>
      <field    name="archivo"    type="string"    indexed="true"
stored="true" required="true"/>
      <field  name="intro"  class="solr.StrField"  indexed="true"
stored="true"/>
      <field name="acceso" class="solr.StrField" indexed="true"
stored="true"/>
      <field        name="historia"        class="solr.StrField"
indexed="true" stored="true"/>
      <field       name="yacimiento"       class="solr.StrField"
indexed="true" stored="true"/>
      <field  name="museo"  class="solr.StrField"  indexed="true"
stored="true"/>
      <field     name="fotos"     type="int"     indexed="true"
stored="true"/>
```

```
      <field     name="planos"     type="int"     indexed="true"
stored="true"/>
      <field      name="docs"      type="int"      indexed="true"
stored="true"/>
      <field    name="timestamp"    type="tdate"    indexed="true"
stored="true"/>
      <field     name="lang"     type="string"     indexed="true"
stored="true" required="true"/>
 </fields>
</schema>
```

Una opción para crear el nuevo esquema al que llamaremos sql seria desde el Diseñador de Esquemas de la interfaz web de SOLR, para ello accedemos a la opción Schema Designer y pulsamos el botón New Schema, llamaremos a este esquema sql:

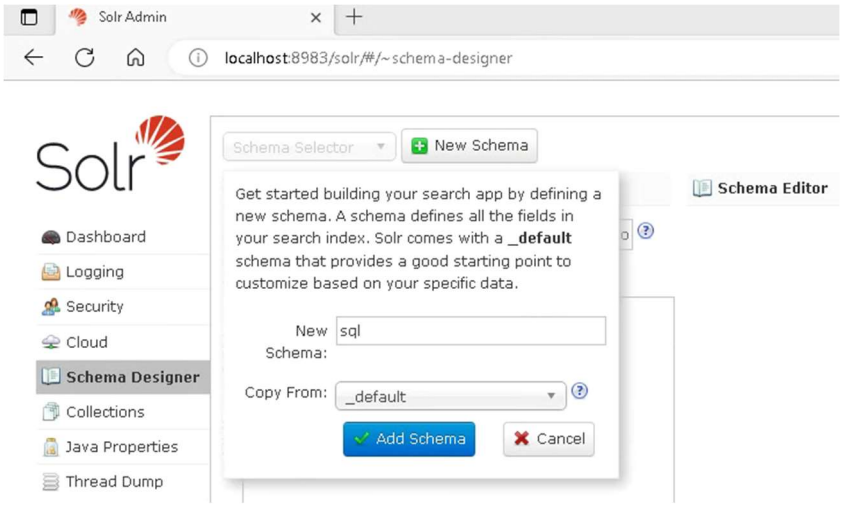


Figura 18 - Creación del Esquema SQL

Para que SOLR cree los campos necesarios para este esquema, una opción es la de carga un documento con los campos que serán necesarios para representar nuestros documentos. Si introducimos el siguiente documento de ejemplo en la casilla correspondiente:

```
<add>
<doc>
<field name="id">4</field>
<field name="codigo">2</field>
<field name="region">ÁTICA</field>
<field name="archivo">MEGARA</field>
<field name="intro">texto omitido para no extender este codigo
</field>
<field name="acceso">texto omitido para no extender este codigo
</field>
```

```
<field name="historia">texto omitido para no extender este codigo
</field>
<field  name="mitologia">texto  omitido  para  no  extender  este
codigo  </field>
<field  name="yacimiento">texto  omitido  para  no  extender  este
codigo  </field>
<field name="museo">texto omitido para no extender este codigo
</field>
<field name="fotos">27</field>
<field name="planos">4</field>
<field name="docs">1</field>
<field name="timestamp">2023-05-17T09:30:22Z/HOUR</field>
<field name="lang">es</field>
</doc>
</add>
```

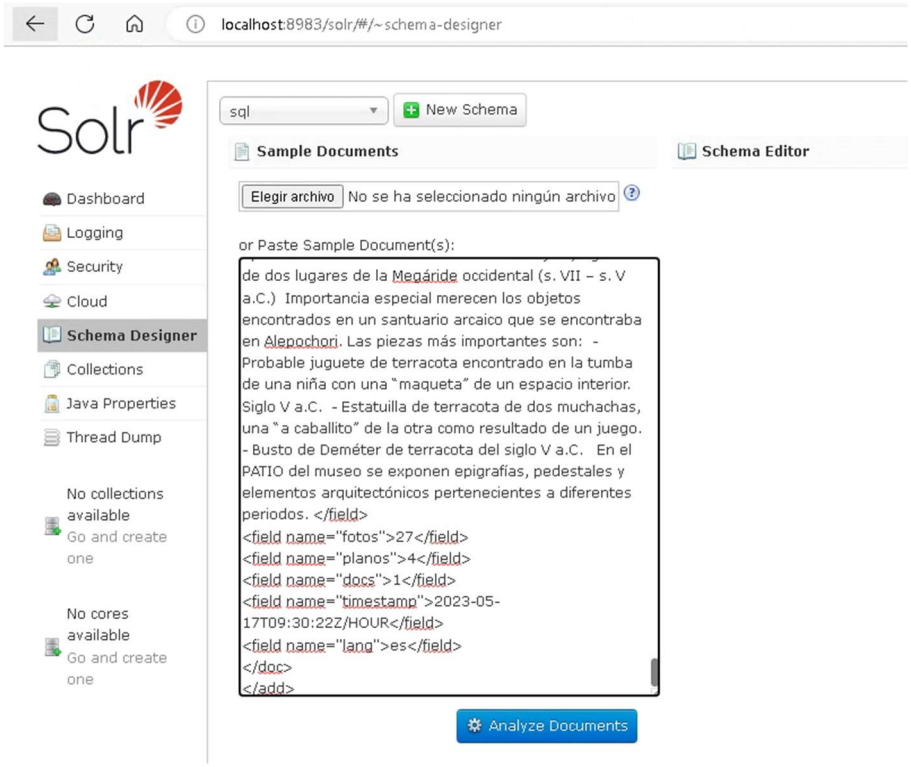


Figura 19 - Inserción de Documento de ejemplo para tipificar campos

Obsérvese que en el código anterior no se han copiado los textos de los campos con la información principal de los campos para no extender esta memoria de manera innecesaria. Pulsamos el botón Analyze Documents y SOLR creara el esquema con los campos que hemos incluido en nuestro documento y decidiendo los parámetros de cada uno de los campos. Esta es la pantalla que muestra el cliente web, una vez que se finaliza el proceso de creación del esquema:

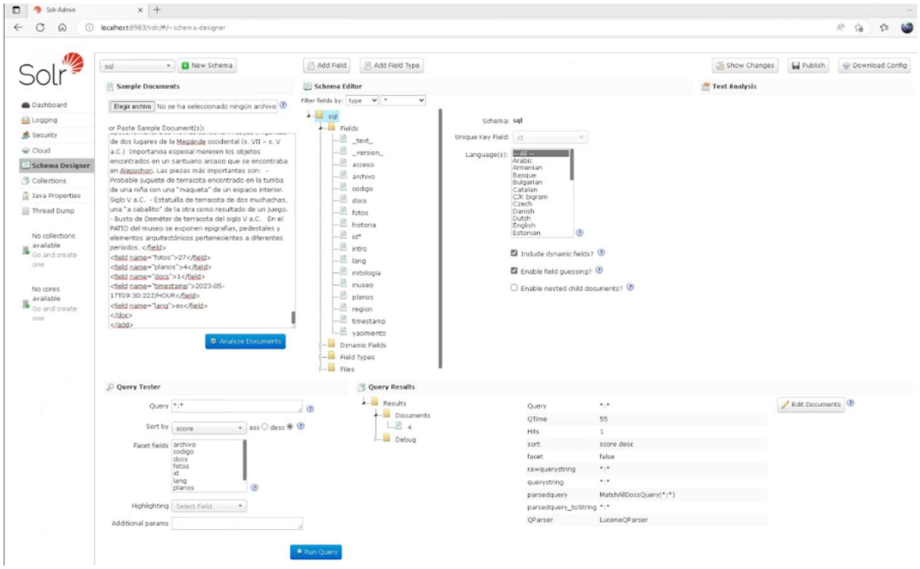


Figura 20 - Campos tipificados de manera automática

A continuación, se muestra una tabla con estos parámetros para los campos que se han generado automáticamente en el esquema sql.

CAMPO /PROPIEDADES	ID	CODIGO	REGION	ARCHIVO	INTRO	ACCESO	HISTORIA	MITOLOGIA	YACIMIENTO	MUSEO	FOTOS	PLANOS	DOCS	TIMESTAMP	LANG
Indexed	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Stored	X				X	X	X	X	X	X					
Multi-Valued					X	X	X	X	X	X					
Doc Values	X	X	X	X							X	X	X	X	X
Use Doc Values as Stored	X	X	X	X							X	X	X	X	X
Required	X														
Tokenized					X	X	X	X	X	X					
Sort Missing Last	X		X	X										X	X
Uninvertible	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Term Vectors															
Term Positions															
Term Offset															
Term Payloads															
Omit Norms	X	X	X	X							X	X	X	X	X
Omit Positions															
Omit Term Frequencies & Positions	X	X	X	X							X	X	X	X	X
Store Offsets with Positions															
Large															

Tabla 9 - Tipos y agrupamientos de campos

Podemos comprobar que la asignación de los tipos de campo de SOLR es acertada, puesto que crea 4 tipos de campos principalmente:

- Campo de identificación del documento, en este grupo se ha clasificado a campo ID
- Campos de gestión del documento, en este grupo se encuadrado los campos CODIGO, REGION, ARCHIVO, TIMESTAMP y LANG que son campos que básicamente contienen información relacionada con el propio archivo y que no contiene un texto interesante para el análisis de este
- Campos de información del documento, se encuadran los campos INTRO, ACCESO, HISTORIA, MITOLOGIA, YACIMIENTO y MUSEO que como ya sabemos, o intuimos, son los campos en los que hemos incluido la información textual de los documentos

- Campos numéricos del documento, se encuadran los campos FOTOS, PLANOS y DOCS que contienen datos numéricos sobre el número de los elementos de esos tipos (foto, planos y docs) que se entregaron para cada uno de los yacimientos.

En caso de que la asignación automática de campos no fuera considerada la más adecuada para el proyecto a desarrollar existe la opción que crear los campos desde el diseñador de esquemas uno a uno y especificando las características para cada uno de ellos.

Si bien este proyecto, y la distribución de la información aportada por el humanista se puede clasificar de manera prácticamente inmediata pues la información es entregada con una clasificación muy definida y los formatos en los que se almacena esa información también, la clasificación automática que ha realizado el diseñador de esquemas se considera optima puesto que ha detectado correctamente todos y cada uno de los campos. Podemos ver en la figura a continuación las propiedades de uno de los campos como puede ser el campo de texto que hemos llamado HISTORIA en el diseñador de esquemas del cliente web.

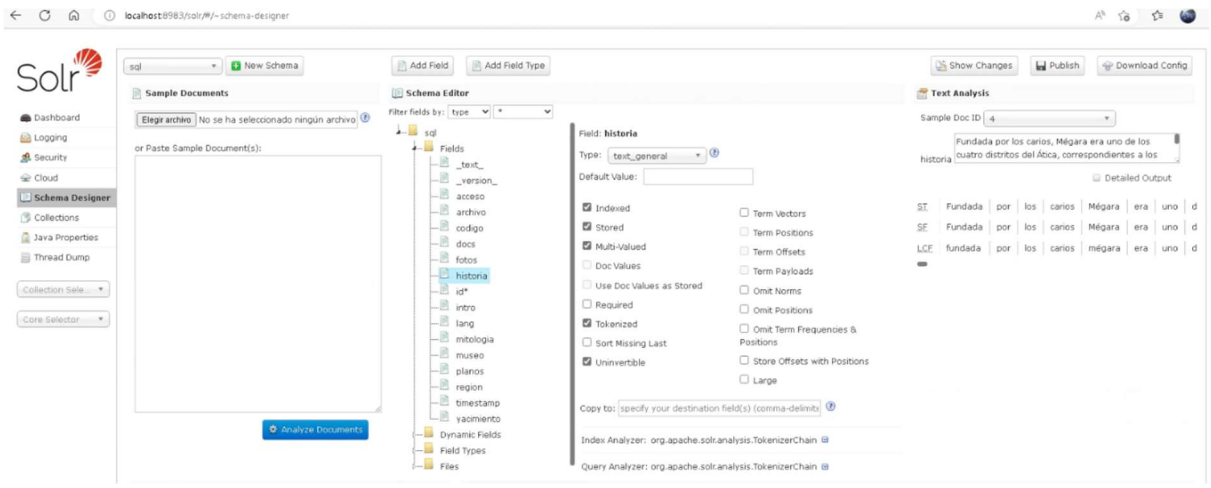


Figura 21 - Detalle de Campo en Diseñador de Esquema

Si bien algunas de las características aplicables a los campos ya han sido definidas anteriormente en esta memoria a continuación, se comentan de manera breve la funcionalidad de las características de los campos.

- Indexed, el contenido del campo se indexa para permitir búsquedas y consultas eficientes, la información no se almacena por completo.
- Stored, el contenido del campo se almacena por completo para su recuperación en los resultados de una búsqueda.
- Multi-Valued, campo que puede contener múltiples valores en un solo documento.
- Doc Values, campo que se guarda de una manera especial en unas estructuras de datos optimizadas para manejar de manera más eficiente ciertas operaciones como las clasificaciones o agrupaciones.

- Use Doc Values as Stored, campo que usa los Doc Values como una manera adicional de almacenamiento para los valores de un campo con el objeto de facilitar una acceso más rápido y eficiente a los valores completos del campo en los resultados de búsquedas y consultas, hay que considerar que el uso de esta característica supone implicaciones en el aumento del tamaño del índice y del uso de memoria lo que puede llevar a degradación del servicio.
- Required, campo requerido, si su valor es True y el campo no tiene valor asociado el sistema rechazara su inserción en la colección.
- Tokenized, si bien este campo no se presenta como tipo de campo en la guía de referencia de SOLR [16] la característica está presente al definir un campo y consiste en un proceso automático que se aplica a ciertos tipos de campo durante la indexación dividiendo el texto en términos individuales para facilitar esa indexación y una búsqueda eficiente.
- Sort Missing Last y Sort Missing First, campos que permiten controlar la ubicación de los documentos cuando no exista campo de ordenación.
- Uninvertible, este campo permite almacenar y recuperar los valores completos de un campo en los resultados de búsqueda sin indexarlos en el índice invertido, puede ser útil cuando se necesita mostrar el contenido completo del campo o cuando el campo no es adecuado para la indexación y búsqueda. Hay que considerar que el uso de esta característica supone implicaciones en el aumento del tamaño del índice y del uso de memoria lo que puede llevar a degradación del servicio, y como se marca por motivos de historial de versiones lo más oportuno es marcarlo como False en los campos del esquema que hemos creado para este apartado.
- Term Vectors, Term Positions, Term Offset y Term Payloads, campos que indican a SOLR que mantenga vectores de términos completos pada cada documento, se pueden usar para acelerar el resultado, pero suponen un costo sustancial en términos de tamaño del índice.
- Omit Norms, campo que omite las normas asociadas al campo, normas que solo son necesarias para los campos de texto completo.
- Omit Term Frequencies & Positions, campo que omite las frecuencias, las posiciones y las cargas útiles de las publicaciones, aporta mayor rendimiento para los campos que no requieren ese tipo de información, reduce el espacio de almacenamiento innecesario para el índice
- Omit Positions, campo similar al anterior pero conserva la información de frecuencia de términos.

- Store Offsets with Positions, campo que permite almacenar tanto las posiciones de los términos como los desplazamientos correspondientes en el texto original, destacando así los términos coincidentes, la recuperación de fragmentos de texto relevante y la implementación de funcionalidades avanzadas de búsqueda, eso sí hay que tener en cuenta que aumenta el tamaño del índice y un posible impacto negativo en el rendimiento.
- Large, los campos grandes siempre se cargan de forma diferida y solo ocuparán espacio en la memoria caché del documento si el valor real es < 512 KB, requiere stored="true"y multiValued="false" y está diseñado para campos que pueden tener valores muy grandes para que no se almacenen en caché en la memoria.

3.4 Inserción de documentos en una colección

En este apartado se describe el proceso de la inserción o inclusión de los documentos en la colección creada con anterioridad.

3.4.1. Inserción de documentos con Apache TIKA

La inclusión de los documentos en la colección que denominamos ARQUEOGRIEGOS-DOC, se realiza usando el kit de herramientas APACHE TIKA. Para cargar un archivo de texto del tipo .doc de Microsoft Word tal cual, se ejecuta el comando:

```
curl                    "http://localhost:8983/solr/ARQUEOGRIEGOS-
DOC/update/extract?literal.id=doc1&commit=true"                -F
myfile=@C:/solr/example/exampledocs/ÁTICA/DOC/00-GEOGRAFIA-DEL-
ÁTICA.doc
```

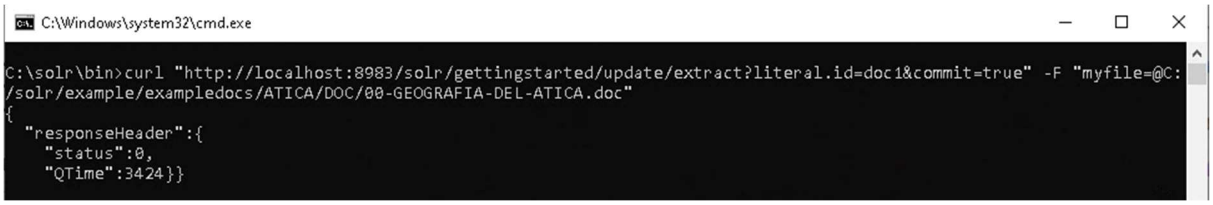


Figura 22 - Información del esquema por defecto usado en el Modo Sin Esquema

Cabe recordar que el entorno de desarrollo usado para este proyecto usa las versiones de las aplicaciones para sistema operativo Microsoft WINDOWS, en concreto la versión WINDOWS 10, por ello en los códigos que se incluyen en esta memoria aparecen comandos con caracteres distintos a los publicados en la documentación de la página oficial de SOLR [16], como ejemplo en algunos casos se usaran las comillas dobles “ en sustitución de las

comillas simples ‘ o para la ejecución de comandos de varias líneas en cURL se usa como salto de párrafo el carácter ^ en lugar del carácter \.

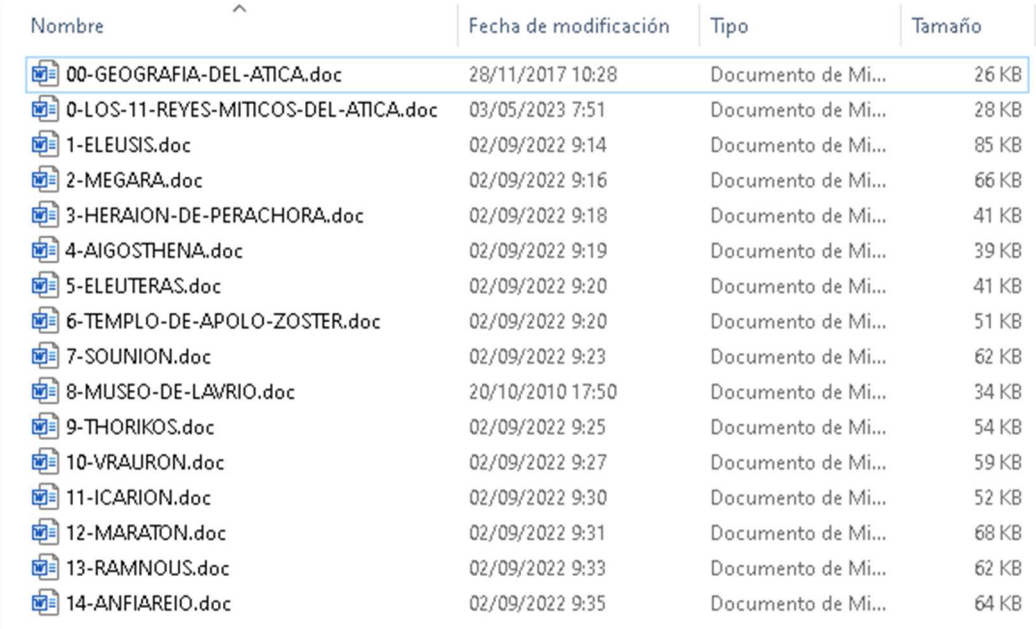
Si fuera necesario eliminar todos los documentos del catálogo usamos este comando de C:\solr:

```
java -Dc=ARQUEOGRIEGOS-DOC -jar example\exampledocs\post.jar
example\exampledocs\delete_all.xml
```

Previamente es necesario crear el archivo delete_all.xml en la carpeta C:\solr\bin\example\exampledocs con este código:

```
<delete>
   <query>*:*</query>
</delete>
```

Repetimos el proceso de inserción de documentos para el resto de los documentos entregados por el humanista para la región de ÁTICA. Se muestra a continuación el contenido de la carpeta con estos archivos.



Nombre	Fecha de modificación	Tipo	Tamaño
00-GEOGRAFIA-DEL-ATICA.doc	28/11/2017 10:28	Documento de Mi...	26 KB
0-LOS-11-REYES-MITICOS-DEL-ATICA.doc	03/05/2023 7:51	Documento de Mi...	28 KB
1-ELEUSIS.doc	02/09/2022 9:14	Documento de Mi...	85 KB
2-MEGARA.doc	02/09/2022 9:16	Documento de Mi...	66 KB
3-HERAION-DE-PERACHORA.doc	02/09/2022 9:18	Documento de Mi...	41 KB
4-AIGOSTHENA.doc	02/09/2022 9:19	Documento de Mi...	39 KB
5-ELEUTERAS.doc	02/09/2022 9:20	Documento de Mi...	41 KB
6-TEMPLO-DE-APOLO-ZOSTER.doc	02/09/2022 9:20	Documento de Mi...	51 KB
7-SOUNION.doc	02/09/2022 9:23	Documento de Mi...	62 KB
8-MUSEO-DE-LAVRIO.doc	20/10/2010 17:50	Documento de Mi...	34 KB
9-THORIKOS.doc	02/09/2022 9:25	Documento de Mi...	54 KB
10-VRAURON.doc	02/09/2022 9:27	Documento de Mi...	59 KB
11-ICARION.doc	02/09/2022 9:30	Documento de Mi...	52 KB
12-MARATON.doc	02/09/2022 9:31	Documento de Mi...	68 KB
13-RAMNOUS.doc	02/09/2022 9:33	Documento de Mi...	62 KB
14-ANFIAREIO.doc	02/09/2022 9:35	Documento de Mi...	64 KB

Figura 23 - Directorio con los documentos .doc para la región ÁTICA

Finalmente, se han cargado todos los documentos en la colección ARQUEOGRIEGOS-DOC, son un total de 16 documentos.

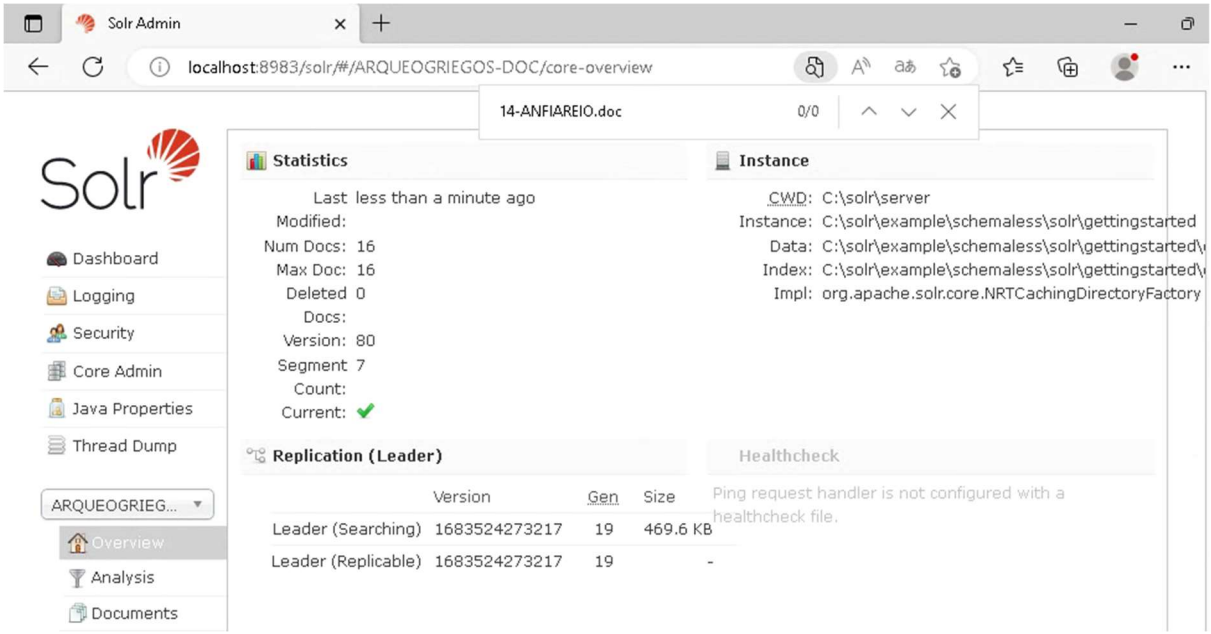


Figura 24 - Vista de SOLR Web Admin IU con resumen indicando los 16 documentos

3.4.2. Inserción de documentos con formato específico y modo sin esquema

En este apartado se describe el proceso de la inserción de los documentos en la colección, a la que denominamos ARQUEOGRIEGOS-XML. Los documentos que se van a insertar son los mismos 16 archivos que ya se incluyeron en la colección ARQUEOGRIEGOS-DOC y que corresponden la región de la Antigua Grecia conocida como ÁTICA.

En primer lugar, recordamos que esta colección se generó con el Modo Sin Esquema y que por lo tanto hay que editar los documentos para adecuarlos a los campos que forman el Modo sin Esquema. Se configuran los campos id, name y description, que contendrán un número del 1 al 16 para el id, los nombres de los documentos para el campo name y el texto contenido en los archivos para el campo description, así por ejemplo para el primer archivo llamado 00-GEOGRAFIA-DEL-ÁTICA insertaremos este archivo en formato .XML:

```
<doc>
<field name="id">1</field>
<field name="name">GEOGRAFIA DEL ÁTICA</field>
<field name="d">GEOGRAFÍA DEL ÁTICA - El Ática debe su nombre a
Átide, hija del mítico rey  Kranaos. Anteriormente el país se
llamaba Actea en memoria de Aktaios, el primer rey mítico del
Ática.
Está separada del Peloponeso por el Canal de Corinto que tiene
una longitud de 6.300 m, anchura 25 m y 8 m de profundidad.
Su topografía es montañosa con algunas llanuras, principalmente
en el este.
A continuación, se enumeran los accidentes geográficos más
importantes que tienen reflejo en el plano adjunto:
```

```
Montes: Geraneia (1.369 m), Pateras (1.132 m), Kitheronas (1.043
m), Párnitha (1.413m), Imitós (1.026 m) y el Pendeli (1.108 m).
Llanuras: Maratón.
Ríos: Asopós, Kifisós e Ilisós.
Golfos: Alciones, Mégara, Elefsina y Sarónico.
Bahías: Porto Germeno, Anávisos, Thorikós, Porto Rafti y Vraona
Cabos: Leptó, Melangani, Tichos, Sounio, Foniá, Mavronori,
Marathona, Drakonera y Agía Marina.
</field>
</doc>
```

Para introducir el archivo .XML, o mejor el contenido del archivo, podemos abrir el cliente web de SOLR y desde el Menú Documents de la colección ir a la opción Document Type XML y copiar el contenido antes indicado.

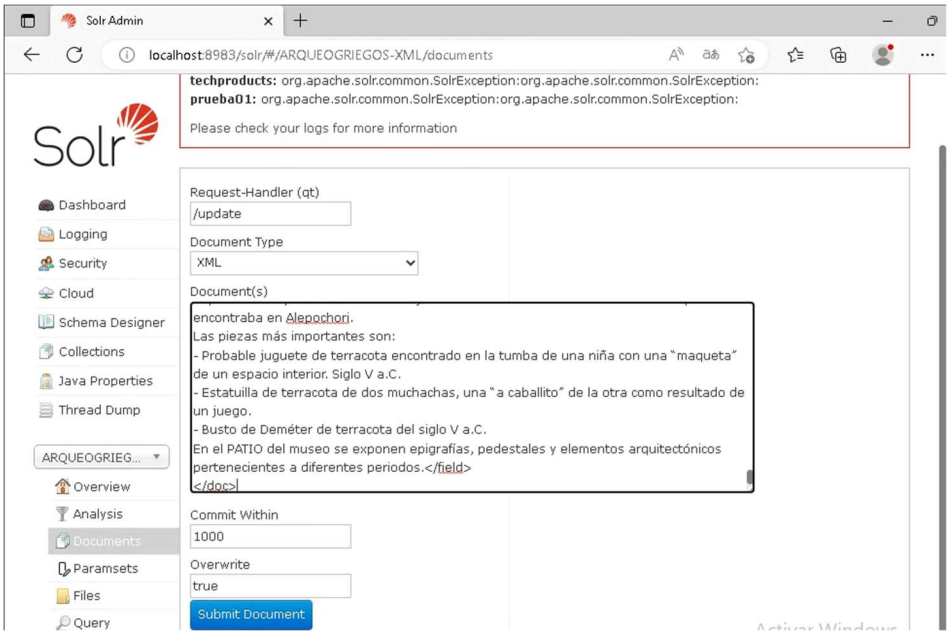


Figura 25 - Inserción de documento en colección

A continuación, hacemos clic en el botón Submit Document y el contenido se cargará en la colección con los contenidos de los campos especificados.

Para comprobar que los datos del documento se han cargado correctamente podemos ir al Menú Query en el cliente web de SOLR y escribir la cadena de texto id:1 en la casilla q de la pantalla para las consultas, a continuación, hacemos clic sobre el botón de la parte inferior y comprobamos si el documento aparece en la ventana de resultados con los datos que hemos copiado previamente.

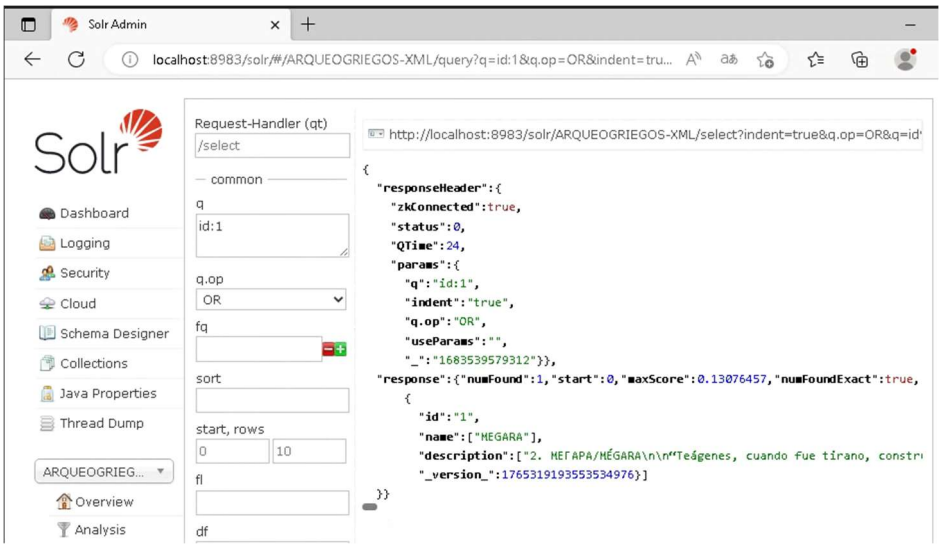


Figura 26 - Comprobación de documento cargado correctamente en colección

Repetimos el proceso para los otros 15 documentos, o bien creamos un único archivo con todos los documentos y sus correspondientes campos para insertarlos en la colección.

3.4.3. Inserción de documentos con esquema predefinido

En este apartado se describe el proceso de la inserción de los documentos en la colección ARQUEOGRIEGOS-ESQ, al igual que usamos el cliente web para la creación del esquema a aplicar a esta colección y en concreto insertando un primer documento se procedió a la determinación de los tipos de campos para cada uno de los campos, procedemos en este apartado a cargar los documentos restantes de la región ÁTICA en esa colección.

A partir del archivo que hemos denominado arqueogriegos-importacion-esquema.xml en el que tenemos este formato:

```
<add>
<doc>
<field name="id">1</field>
<field name="codigo">0</field>
<field name="region">ÁTICA</field>
<field name="archivo">GEOFRAFIA DEL ÁTICA</field>
<field name="intro"> texto omitido para no extender este codigo
</field>
<field name="acceso"> </field>
<field name="historia"> </field>
<field name="mitologia"> </field>
<field name="yacimiento"> </field>
<field name="museo"> </field>
<field name="fotos">0</field>
```

```
<field name="planos">0</field>
<field name="docs">0</field>
<field name="timestamp">2023-05-17T09:30:22Z/HOUR</field>
<field name="lang">es</field>
</doc>
… … …
<doc>
<field name="id">16</field>
<field name="codigo">14</field>
<field name="region">ÁTICA</field>
<field name="archivo">ANFIAREIO </field>
<field name="intro"> texto omitido para no extender este codigo
</field>
<field name="acceso"> texto omitido para no extender este codigo
</field>
<field name="historia"> texto omitido para no extender este codigo
</field>
<field name="mitologia"> texto omitido para no extender este
codigo </field>
<field name="yacimiento"> texto omitido para no extender este
codigo </field>
<field name="museo"> </field>
<field name="fotos">41</field>
<field name="planos">9</field>
<field name="docs">1</field>
<field name="timestamp">2023-05-17T09:30:22Z/HOUR</field>
<field name="lang">es</field>
</doc>
</add>
```

Accedemos al cliente web en primer lugar y a continuación a la colección ARQUEOGRIEGOS-ESQ, ahora se accede a la opción Documentos, seleccionamos en Document Type la opción XML y en el espacio Documento copiamos el contenido del archivo arqueogriegos-importa-esquema.xml que nos cargara todos los documentos, en concreto son 16, que compondrán esta colección, hacemos clic en el botón Submit Document.

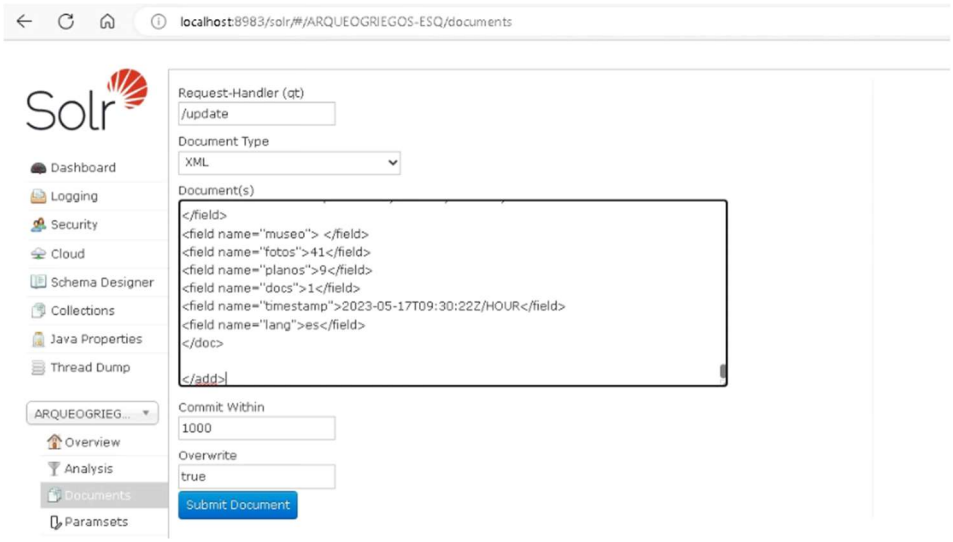


Figura 27 - Inserción de varios documentos en colección ARQUEOGRIEGOS-ESQ

Una vez que ha terminado el proceso de inserción de los documentos en la colección, se comprueba si los 16 documentos han quedado correctamente incluidos accediendo a la sección Query de la colección en el cliente web con los parámetros por defecto de esta.

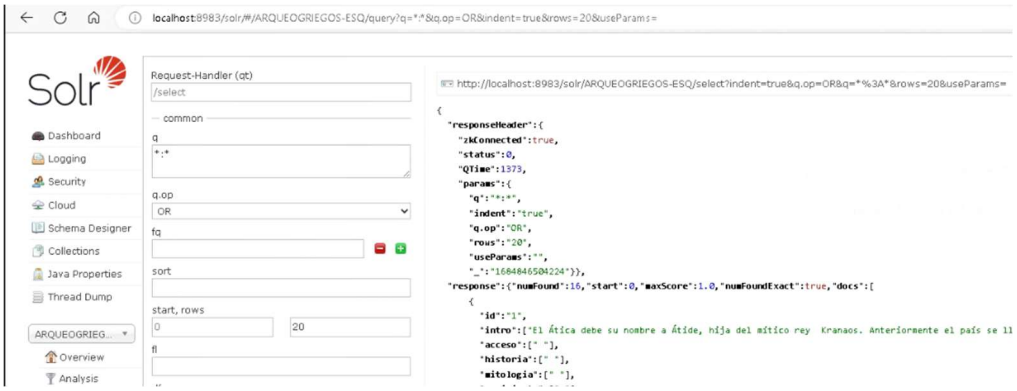


Figura 28 - Comprobación de Documentos insertados en Colección ARQUEOGRIEGOS-ESQ

Capítulo 4: Prototipo desarrollado y pruebas

Para el desarrollo del prototipo a implementar hay que recordar los requisitos que se establecieron y que de manera más breve reseñamos a continuación:

- REQUISITO 1: habilitar una "cantera" de materiales aptos para una función didáctica que incluyera una recuperación de la información ágil y eficiente.
- REQUISITO2: habilitar un acceso a información a modo de guía de viaje para los visitantes de un yacimiento arqueológico o museo de los recogidos en el estudio.

Con estas premisas se diseñan e implementan dos aplicaciones web, relacionadas entre sí, por un lado y para atender al requisito 1 se habilita un servidor con SOLR y por otro lado, y para dar respuesta al requisito 2 se habilita una página web con el administrador de contenidos WORDPRESS. El trabajo realizado se describe en el siguiente apartado.

Los dos apartados siguientes en este capítulo están dedicados a las pruebas a realizar sobre las aplicaciones desarrolladas. Las pruebas iniciales pensadas para ambas aplicaciones estaban basadas en la resolución de preguntas mediante el uso de la interfaz web de Apache SOLR para el caso del requisito 1, y de acceso a la información para el caso del requisito 2. Así pues, en el siguiente apartado, denominado PRUEBA HDH2023 CORTA, se presenta una prueba de consultas básicas a ambas aplicaciones en la que buscaremos respuesta a una de las preguntas planteadas por el humanista, en concreto a la *Pregunta 3: ¿Cuál es el conjunto de teatros que hay en una determinada región de las que está dividido el estudio?*

A finales de mayo de 2023, la tutora de este proyecto plantea la posibilidad de presentar este trabajo al VI Congreso de la HDH [19], que se celebrará en Logroño del 18 al 20 de octubre de 2023 y que lleva como título "HDH2023 Encuentros y transformaciones: las Humanidades Digitales como propuesta transdisciplinar" y como es aceptado, procedemos a ampliar la batería de pruebas a las aplicaciones con la llamada PRUEBA HDH2023 LARGA.

4.1. Servidores

4.1.1 Servidor SOLR

Esta aplicación web ya ha quedado suficientemente descrita en capítulos anteriores, instalándose en un servidor local que tiene habilitado el acceso remoto a través de un servicio

de dinámico de DNS, quedando disponible en la dirección web publica http://solr.dynalias.com:8983 y con acceso protegido mediante los siguientes parámetros: Username: **fpanos3** Password. **comm23** (ver figura siguiente).

La colección que se ha creado en última instancia y a la que hemos llamado ARQUEOGRIEGOS-ESQ contenía los documentos aportados por el humanista de la región de la Antigua Grecia denominada Ática, se amplía con el mismo procedimiento con la colección de los documentos de Ática, los documentos de la región denominada Tracia, que si bien son únicamente cuatro documentos nos permitirá hacer pruebas más completas sobre la colección.

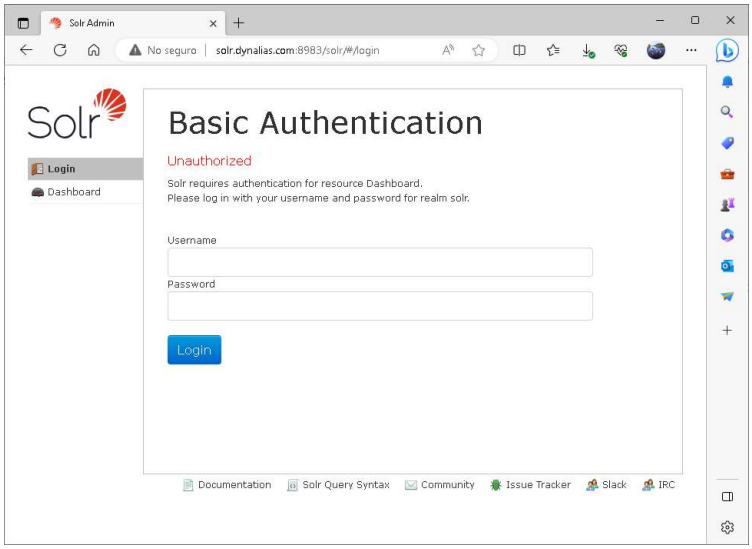


Figura 29 - Acceso a SOLR en servidor local accesible online

Una vez que se hayan insertado estos documentos en la colección podemos determinar que nuestro prototipo estará listo para pasar al proceso de pruebas y evaluación, este prototipo o modelo inicial estará compuesto por el servidor SOLR en ejecución en un ordenador habilitado a tal efecto, para las pruebas se realizara un acceso local desde la dirección http://localhost:8983 e igualmente quedara habilitado un acceso remoto en la dirección http://solr.dynalias.com:8983. Este servidor esta iniciado en Modo Cloud con la colección ARQUEGRIEGOS-ESQ cargada.

4.1.2. Página web WORDPRESS

Para esta aplicación web se ha decidido usar el Sistema de Administración de Contenidos (CMS Content Management System) WORDPRESS en su última versión 6.2.2 para dejar disponible de manera pública y en línea la información del estudio del humanista de una manera más amigable. Se selecciona WORDPRESS porque para habilitar esta página web era vital encontrar algún módulo de terceros que pudiera establecer la conexión entre la

propia página web y el servidor SOLR habilitado en la dirección http://solr.dynalias.com:8983 puesto que el desarrollo de ese modulo sería una tarea inabordable en este proyecto por razones obvias. Una vez buscadas opciones al respecto se encontraron dos Sistemas de Administración de Contenidos (CMS Content Management System) que tenían disponible un módulo de ese tipo, estos CMS y sus módulos son los siguientes:

- CMS DRUPAL y módulo Search API SOLR [20]

- CMS WORDPRESS y modulo (plugin) WPSOLR FREE [21]

Tras una revisión a ambas opciones finalmente se toma de decisión de usar WORDPRESS y el Plugin WPSOLR FREE porque se intuye como una mejor solución para dar cumplimiento al Requisito 2 propuesto por el humanista.

Para la instalación de este sistema se cuenta con el apoyo desinteresado de la empresa Tres Isi Mirobriga, s.l.u. que nos proporciona un hosting profesional, con recursos limitados evidentemente, y un subdominio sobre el que realizar la instalación de la página web con WORDPRESS y el plugin ya mencionado.

La dirección de acceso a esta página web es https://arqueogriegos.3isi.com y para el acceso al backend de administración https://arqueogriegos.3isi.com/wp-admin para cuyo acceso podemos introducir las mismas credenciales que indicamos anteriormente para el acceso el servidor con Apache SOLR instalado.

Figura 30 - Acceso a la página web https://arqueogriegos.3isi.com

4.2. Prueba HDH2023: ARQUEOGRIEGOS (Ática y Tracia)

El humanista responsable de la información de la colección ARQUEOGRIEGOS nos traslada varias preguntas cuyas respuestas pudieran ser encontradas de la manera más rápida posible por el sistema a desarrollar e implementar del proyecto ARQUEOGRIEGOS. Para ello, se selecciona una de ellas:

PREGUNTA 3, ¿Cuál es el conjunto de teatros que hay en una determinada región de las que está dividido el estudio?

Para dar respuesta a esta pregunta se usarán una serie de recursos y aplicaciones con el fin de comparar los tiempos de respuesta de las consultas, comenzamos por la Pregunta 3, y podemos encuadrar esos recursos y aplicaciones en cuatro procesos:

4.2.1 Respuesta con VOYANT TOOLS

En primer lugar, hacemos un estudio de las dos zonas de las que el humanista nos ha entregado información, que son Ática y Tracia, y realizamos un análisis de la frecuencia con la que aparecen los términos a buscar, que serían TEATRO para la Pregunta3 y ZEUS para la pregunta 2. Se va a realizar el análisis solo para esta deidad puesto que el proceso sería similar para el resto de las deidades relacionadas en la pregunta. A continuación, hay que decidir si los términos aparecidos responden a lo preguntado o no, pues podría aparecer el resultado sin relación con la pregunta. La información aportada por el humanista para estas dos regiones está en los siguientes 22 documentos:

ÁTICA (16 documentos): Historia (1); Geografía (1); Yacimientos (14)

TRACIA (6 documentos): Historia (1); Geografía (1); Yacimientos (4)

Estos 22 documentos .DOC entregados por el humanista han sido transformados en archivos .TXT, .XML para insertar en SOLR, en concreto cargamos en VOYANT TOOLS los .TXT. Vemos que este término TEATRO aparece en 28 ocasiones, 27 en su forma singular “teatro” y 1 en su forma plural “teatros”.

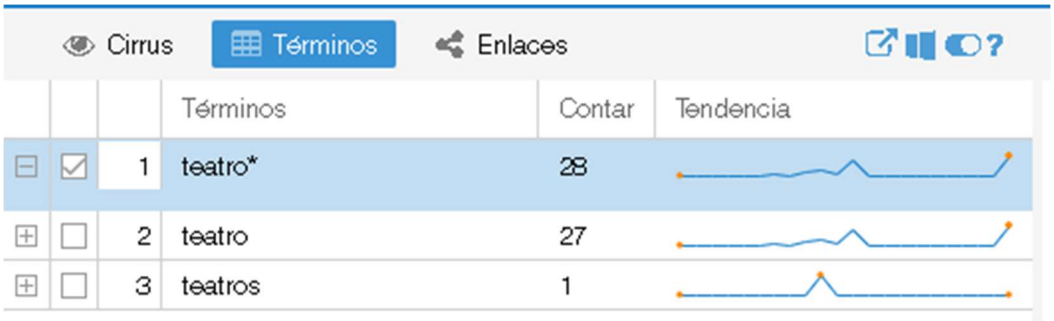
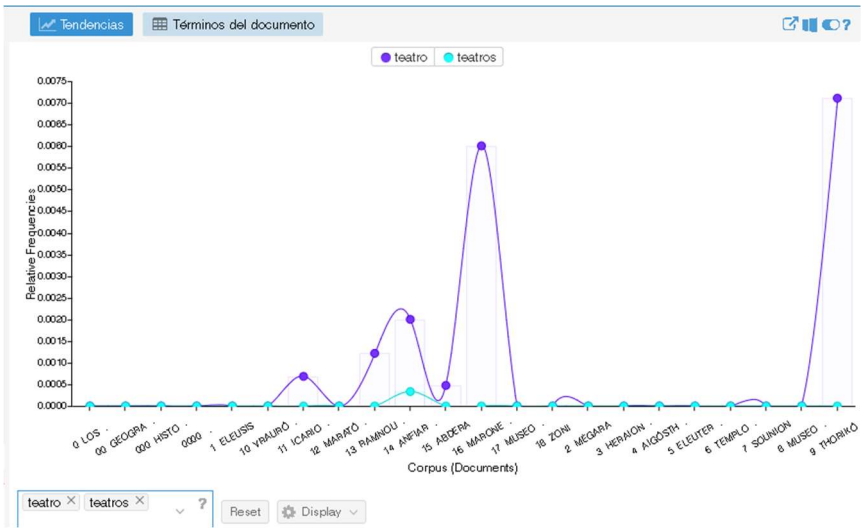
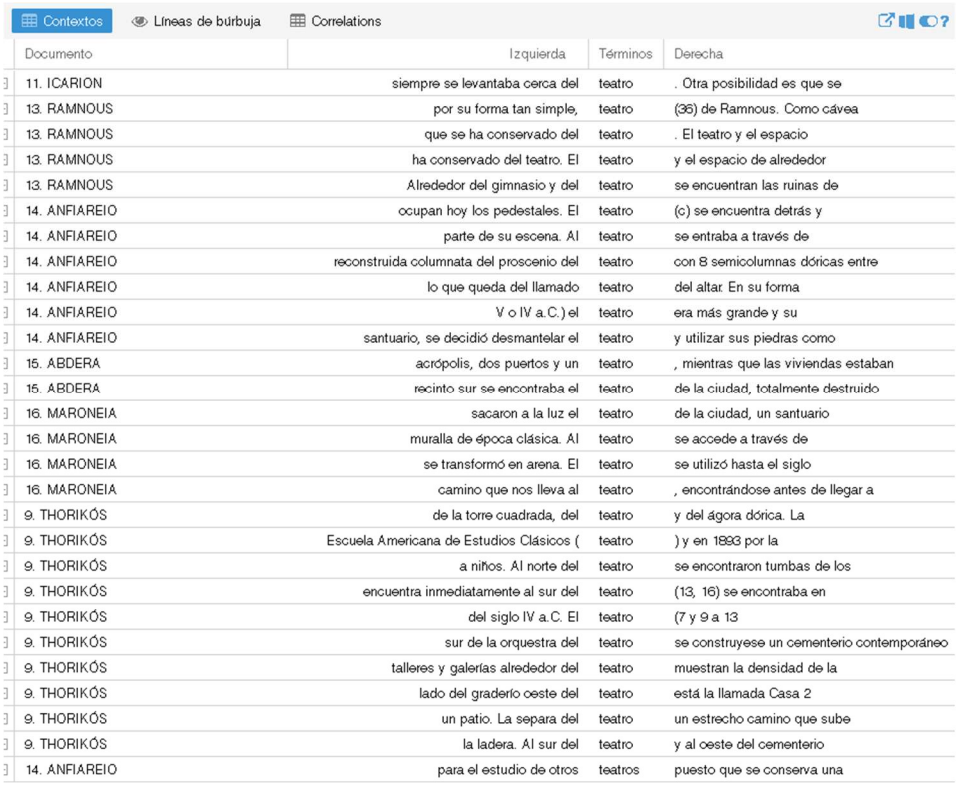


Figura 31 - Captura de la web de VOYAN TOOLS (Términos)





Documento	Izquierda	Términos	Derecha
11. ICARION	siempre se levantaba cerca del	teatro	. Otra posibilidad es que se
13. RAMNOUS	por su forma tan simple,	teatro	(36) de Ramnous. Como cávea
13. RAMNOUS	que se ha conservado del	teatro	. El teatro y el espacio
13. RAMNOUS	ha conservado del teatro. El	teatro	y el espacio de alrededor
13. RAMNOUS	Alrededor del gimnasio y del	teatro	se encuentran las ruinas de
14. ANFIAREIO	ocupan hoy los pedestales. El	teatro	(c) se encuentra detrás y
14. ANFIAREIO	parte de su escena. Al	teatro	se entraba a través de
14. ANFIAREIO	reconstruida columnata del proscenio del	teatro	con 8 semicolumnas dóricas entre
14. ANFIAREIO	lo que queda del llamado	teatro	del altar. En su forma
14. ANFIAREIO	V o IV a.C.) el	teatro	era más grande y su
14. ANFIAREIO	santuario, se decidió desmantelar el	teatro	y utilizar sus piedras como
15. ABDERA	acrópolis, dos puertos y un	teatro	, mientras que las viviendas estaban
15. ABDERA	recinto sur se encontraba el	teatro	de la ciudad, totalmente destruido
16. MARONEIA	sacaron a la luz el	teatro	de la ciudad, un santuario
16. MARONEIA	muralla de época clásica. Al	teatro	se accede a través de
16. MARONEIA	se transformó en arena. El	teatro	se utilizó hasta el siglo
16. MARONEIA	camino que nos lleva al	teatro	, encontrándose antes de llegar a
9. THORIKÓS	de la torre cuadrada, del	teatro	y del ágora dórica. La
9. THORIKÓS	Escuela Americana de Estudios Clásicos (	teatro	) y en 1893 por la
9. THORIKÓS	a niños. Al norte del	teatro	se encontraron tumbas de los
9. THORIKÓS	encuentra inmediatamente al sur del	teatro	(13, 16) se encontraba en
9. THORIKÓS	del siglo IV a.C. El	teatro	(7 y 9 a 13
9. THORIKÓS	sur de la orquestra del	teatro	se construyese un cementerio contemporáneo
9. THORIKÓS	talleres y galerías alrededor del	teatro	muestran la densidad de la
9. THORIKÓS	lado del graderío oeste del	teatro	está la llamada Casa 2
9. THORIKÓS	un patio. La separa del	teatro	un estrecho camino que sube
9. THORIKÓS	la ladera. Al sur del	teatro	y al oeste del cementerio
14. ANFIAREIO	para el estudio de otros	teatros	puesto que se conserva una

Figura 32 - Capturas de la web de VOYAN TOOLS (Tendencias y Contextos)

Analizando con detalle las apariciones del término vemos que aparece en los yacimientos de THORIKOS, ICARION, RAMNOUS, ANFIAREIO en Ática y ABDERA y MARONEIA de Tracia. De estos yacimientos, todos excepto ICARION, tienen restos de un teatro.

Para responder a la Pregunta 3, deberíamos utilizar los contenidos de otras de las herramientas disponibles en VOYANT TOOLS, en este caso si observamos el apartado CONTEXT observar un patrón en el que se podría afirmar que siempre que figura el término

“teatro” tras los términos “al”, “del” o “el” se hace referencia a la existencia de un teatro. Así, de manera más o menos breve podemos determinar que la respuesta es:

El conjunto de teatros en ÁTICA es de 3 y en TRACIA de 2

TIEMPO CONSULTA: dentro de la aplicación web de VOYANT TOOLS no se muestran tiempos de ejecución de las consultas realizadas, o al menos no he conseguido acceder a esos tiempos, aunque es inmediato.

4.2.2 Respuesta con la BASE DE DATOS RELACIONAL

Se hace una consulta mediante la utilidad PHPMYADMIN en la base de datos **arqueogriegos_sql** que se ha diseñado y alimentado con la información del humanista, que está alojada en un servidor online y cargada en la dirección https://arqueogriegos.3isi.com y esta accesible desde las herramientas de administración del servidor que la aloja, aunque por razones de seguridad no se habilita el acceso al gestor de esta base de datos a cualquier usuario, pero se ha habilitado un segundo servidor https://solr.dynalias.com/phpmyadmin. En esta base de datos se ha almacenado toda la información de los documentos entregados por el humanista para la realización de las consultas.

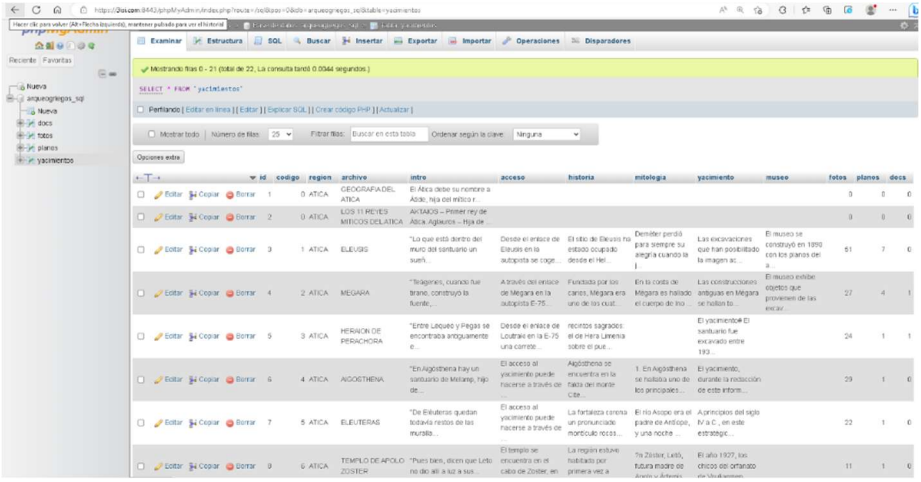


Figura 33 - Acceso a PHPMYADMIN a BBDD arqueogriegos_sql

Si buscamos la palabra teatro obtenemos este resultado:

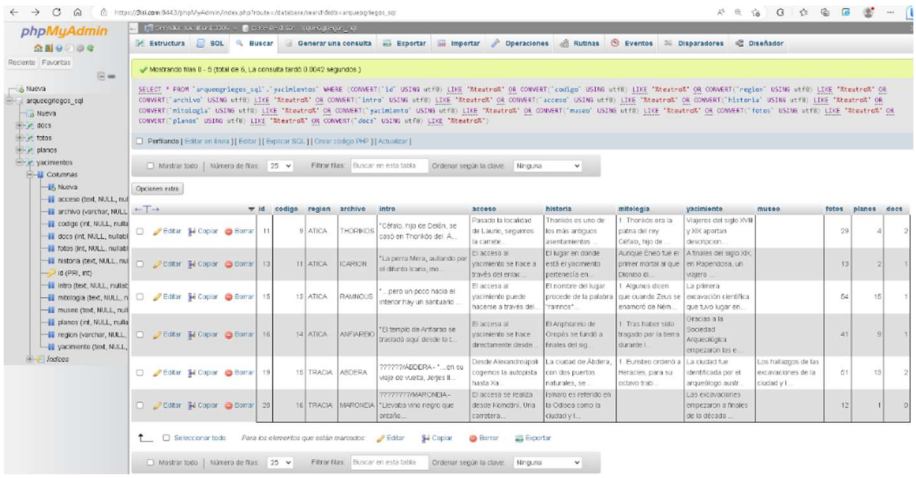


Figura 34 - Consulta del término TEATRO en la BBDD arqueogriegos_sql

Esta sería la consulta en SQL:

```
SELECT * FROM `arqueogriegos_sql`.`yacimientos` WHERE (CONVERT(`id` USING utf8) LIKE '%teatro%' OR CONVERT(`codigo` USING utf8) LIKE '%teatro%' OR CONVERT(`region` USING utf8) LIKE '%teatro%' OR CONVERT(`archivo` USING utf8) LIKE '%teatro%' OR CONVERT(`intro` USING utf8) LIKE '%teatro%' OR CONVERT(`acceso` USING utf8) LIKE '%teatro%' OR CONVERT(`historia` USING utf8) LIKE '%teatro%' OR CONVERT(`mitologia` USING utf8) LIKE '%teatro%' OR CONVERT(`yacimiento` USING utf8) LIKE '%teatro%' OR CONVERT(`museo` USING utf8) LIKE '%teatro%' OR CONVERT(`fotos` USING utf8) LIKE '%teatro%' OR CONVERT(`planos` USING utf8) LIKE '%teatro%' OR CONVERT(`docs` USING utf8) LIKE '%teatro%')
```

Se producen 6 coincidencias, es decir el término aparece en 6 yacimientos.

TIEMPO CONSULTA: 0 - 5 (total de 6, La consulta tardó 0,0066 segundos.) o 6,6 milisegundos sobre un CORPUS con 277.283 caracteres y 48254 palabras

Si queremos ver la consulta que nos daría este valor, ejecutamos:

```
SELECT COUNT(*) AS num_coincidencias FROM yacimientos WHERE (CONVERT(`id` USING utf8) LIKE '%teatro%' OR CONVERT(`codigo` USING utf8) LIKE '%teatro%' OR CONVERT(`region` USING utf8) LIKE '%teatro%' OR CONVERT(`archivo` USING utf8) LIKE '%teatro%' OR CONVERT(`intro` USING utf8) LIKE '%teatro%' OR CONVERT(`acceso` USING utf8) LIKE '%teatro%' OR CONVERT(`historia` USING utf8) LIKE '%teatro%' OR CONVERT(`mitologia` USING utf8) LIKE '%teatro%' OR CONVERT(`yacimiento` USING utf8) LIKE '%teatro%' OR CONVERT(`museo` USING utf8) LIKE '%teatro%' OR CONVERT(`fotos` USING utf8) LIKE '%teatro%' OR CONVERT(`planos` USING utf8) LIKE '%teatro%' OR CONVERT(`docs` USING utf8) LIKE '%teatro%');
```

Hay que tener en cuenta que una consulta SQL devolverá un número de coincidencias o un campo de una tabla que contiene la palabra búsqueda, pero no puede hacer ninguna otra interpretación o dar más información. Para entender el contexto de una coincidencia hay

que leer el campo por completo y obtener la interpretación de manera subjetiva, así para responder a la Pregunta 3 necesitaríamos mucho más tiempo que en el caso anterior para llegar a determinar lo mismo: El conjunto de teatros en ÁTICA es de 3 y en TRACIA de 2

4.2.3 Respuesta con - COLECCIÓN ARQUEOGRIEGOS_ESQ

En este caso realizaremos una búsqueda en la colección llamada ARQUEOGRIEGOS_ESQ que se ha desarrollado para ARQUEOGRIEGOS. Como se ha indicado anteriormente, la colección está disponible en un servidor SOLR que esta accesible desde esta dirección https://solr.dynalias.com:8983 para acceder a ella será necesario usar el usuario **fpanos3** y la contraseña **comm23**

Figura 35 - Acceso a SOLR

A continuación, accedemos a la colección llamada ARQUEOGRIEGOS-ESQ y nos dirigimos a la opción QUERY para comenzar con la configuración de la consulta, estos son los parámetros que aplicaremos a esta consulta:

- *Parámetro q: *:** -> con esto indicamos que queremos buscar en todos los documentos de la colección
- *Parámetro fq: yacimiento:teatro* -> Filter Query indica que queremos buscar en el campo yacimiento el termino teatro
- *Parámetro fl: archivo,yacimiento* -> Filter activa un filtro para mostrar los campos que indiquemos en este caso archivo y yacimiento, es decir el nombre del Yacimiento y la localización en el campo yacimiento del término teatro
- *Parámetro df: yacimiento* -> Default Field para enfatizar que la búsqueda debe centrarse en el campo indicado en este caso yacimiento
- *Parámetro hl: habilitado* -> Indica que habilitamos la opción de marcar (highlight) las apariciones del término buscado
- *Parámetro hl.fl: yacimiento* -> Marcamos el campo en el que queremos resaltar los resultados
- *Parámetro hl.usePhraseHighighter: Habilitado* -> Indica que queremos marcar la frase donde se encuentra el termino encontrado
- *Parámetro hl.q: yacimiento:teatro* -> Indicamos el término a buscar y el campo donde queremos hacer la búsqueda

- *Parámetro hl.fragSize: 5* -> Limita la longitud a mostrar de la frase donde se encuentra el termino encontrado

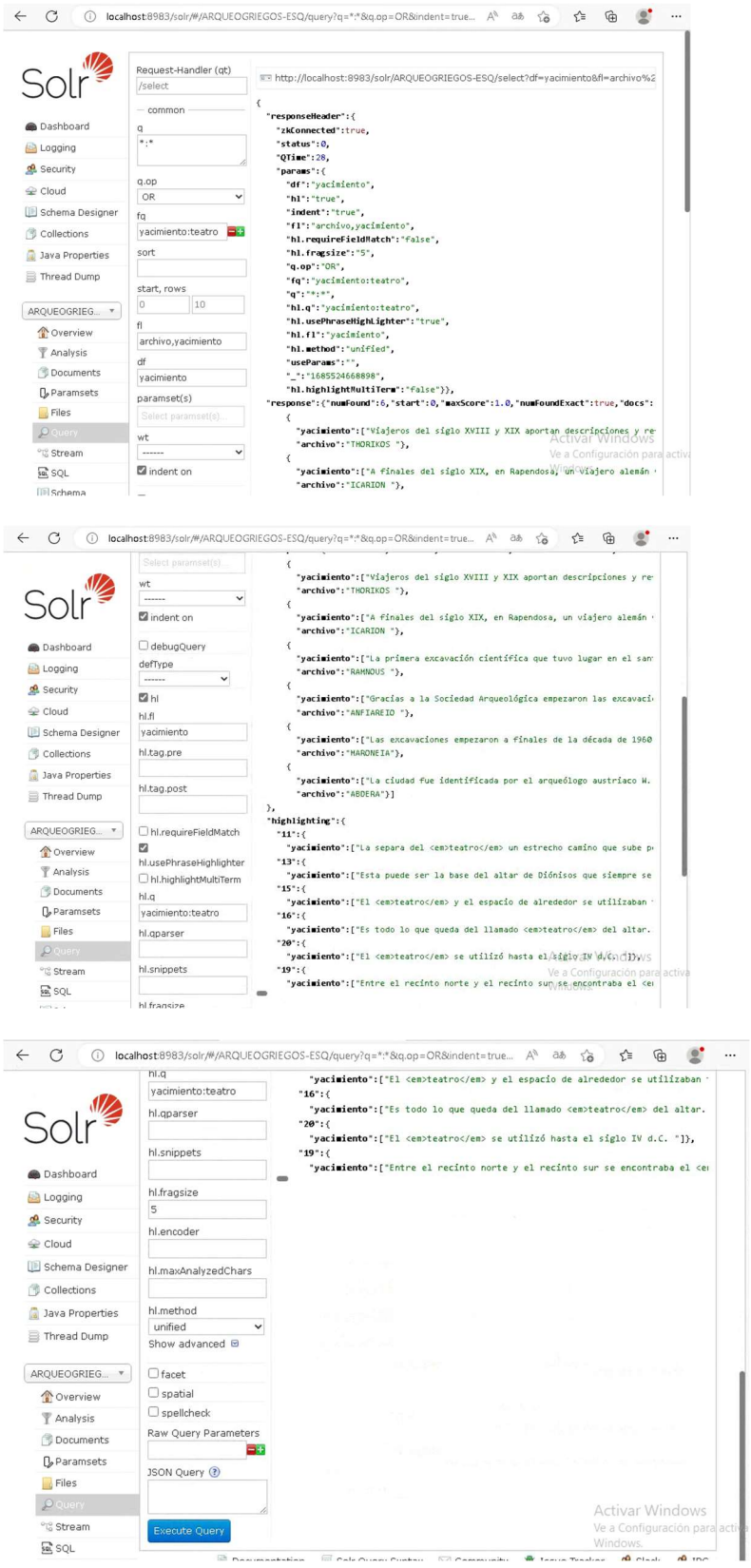


Figura 36 - Búsqueda en la Colección ARQUEOGRIEGOS_ESQ del término TEATRO

La herramienta de búsqueda del cliente web de SOLR quizá no es todo lo amigable que nos gustaría, otra opción para realizar las búsquedas es usar un terminal de consola para realizar consultas cURL que tampoco es la mejor opción para mostrar los resultados a usuarios que no tengan mucha practica con estos entornos, por esto se ha decidido en una primera aproximación que la respuesta a la búsqueda sea mostrada en formato JSON que posteriormente será visualizado en una aplicación online como https://jsonviewer.stack.hu para mejorar la disposición visual del resultado de la búsqueda, podemos verlo en la figura a continuación:

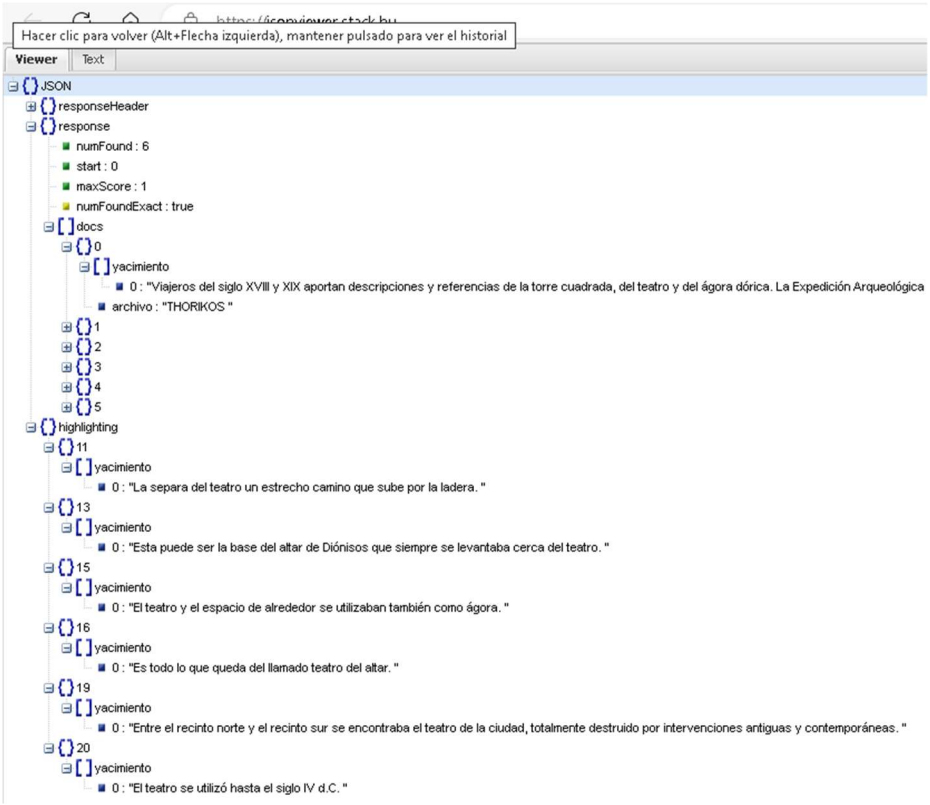


Figura 37 - JSON con el resultado de la búsqueda

En los resultados de la búsqueda anterior la aparición del término buscado es mucho más concreta y localizada que en el caso de la búsqueda en la base de datos relacional, es algo que de antemano ya se intuía puesto que estas bases de datos no son la mejor tecnología de almacenamiento para alojar cadenas de texto tan extensas como en este caso, en cambio las tecnologías como SOLR o bases de datos NoSQL son mucho más apropiadas para buscar datos o valores más concretos, además la velocidad de búsqueda en SOLR es mayor lo que nos confirma que esta tecnología puede ser más apropiada para la necesidad planteada por el humanista. La respuesta es: El conjunto de teatros en ÁTICA es de 3 y en TRACIA de 2

Para obtener los tiempos de ejecución de la consulta realizada podemos activar la opción DEBUG en el cliente web de Apache SOLR o añadiendo ***&debug=timing*** al final de la consulta realizada a través de comando HTTP con lo que obtendremos información detallada de los tiempos usados en las distintas fases de la consulta así como otras informaciones que pudieran ser interesantes para depurar la propia consulta.

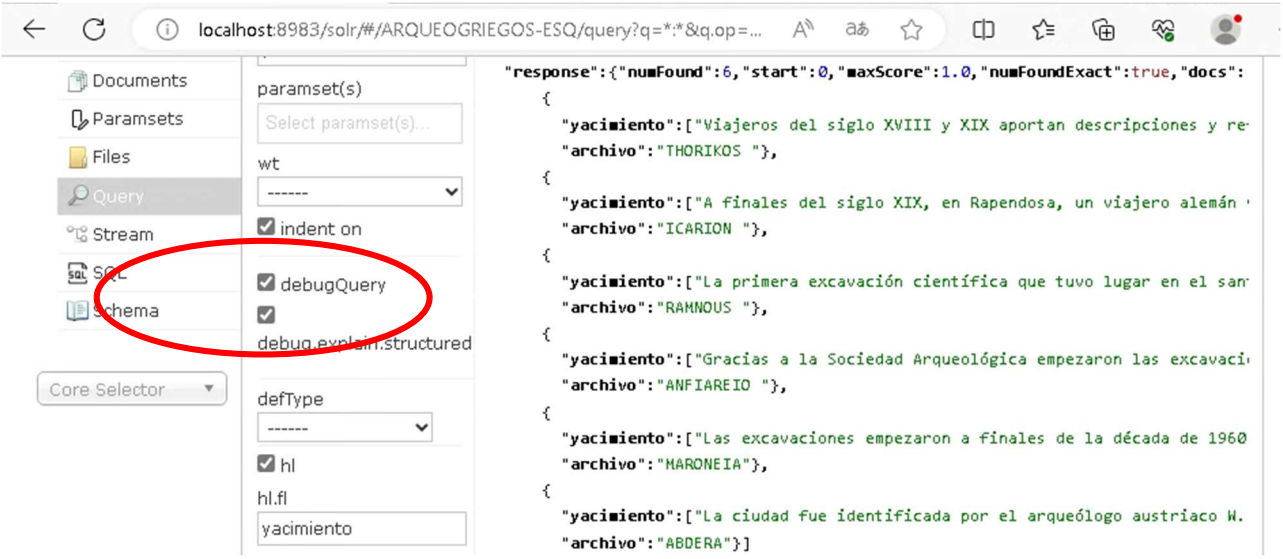


Figura 38 - Activación de la opción DEBUG en cliente web de SOLR

La información sobre el tiempo transcurrido durante la consulta la podemos ver en los valores **time** dentro de las etiquetas **timing**.

Figura 39 - Valores TIME en etiqueta TIMING

Esta consulta la podemos reproducir ejecutando este comando HTTP contra el servidor que aloja a Apache SOLR:

```
http://solr.dynalias.com:8983/solr/#/ARQUEOGRIEGOS-
ESQ/query?q=*:*&q.op=OR&indent=true&fl=archivo,yacimiento&df=yac
imiento&debugQuery=true&debug.explain.structured=true&hl=true&hl
.fl=yacimiento&hl.usePhraseHighLighter=true&hl.q=yacimiento:teat
ro&hl.fragsize=5&fq=yacimiento:teatro&useParams=
```

Hay que tener en cuenta que en el desglose que hace el proceso DEBUG se consideran distintos apartados como pueden ser PREPARE, PROCESS, HIGLIGHTS, EXPAND, STATS, TERMS, DEBUG…para este caso no tendremos en cuenta todos, solo los básicos, para hacer las comparativas con los tiempos de las consultas que se realizarán en las bases de datos relacionales que serán los apartados PROCESS y HIGHLIGHTS. Este último lo consideramos muy importante para encontrar las respuestas, pues recuperar un campo completo nos obligaría a tener que buscar el termino dentro de campos que pueden tener extensiones en números de palabra muy considerables y por lo tanto no sería eficiente en absoluto.

TIEMPO CONSULTA: la consulta ha dado como valor de time 22 milisegundos. Incluiría 17 milisegundos para el apartado PROCESS, que a su vez incluye 2 milisegundos más para el apartado DEBUG, 14 milisegundos para el apartado HIGHLIGHTS y 1 milisegundo para el apartado QUERY. Si descartamos el tiempo de DEBUG de 2 milisegundos tendremos 20 milisegundos, por lo que podemos cifrar el tiempo de la consulta en 0,020 segundos o 20 milisegundos sobre un CORPUS con 277.283 caracteres y 48.254 palabras.

4.2.4 Respuesta mostrada en WORDPRESS

Como última aproximación y teniendo muy presente lo que se indicaba anteriormente que las salidas de las consultas en SOLR no son adecuadas para usuarios no informáticos, se ha implementado una página web con el administrador de contenidos WORDPRESS con el Plugin WPSOLR https://www.wpsolr.com que permite una conexión entre los contenidos publicados en WORDPRESS como Entradas, Paginas… y una colección que se genera a partir de los mismos y que se puede sincronizar en tiempo real. De esta manera conseguimos un acabado a la hora de ver las respuestas del buscador mucho más interesante y personalizable que la salida natural del cliente web de SOLR o las salidas por consola.

La página web indicada esta accesible a través de la dirección https://arqueogriegos.3isi.com y en ella se ha, cargado los textos y las imágenes de los 22 documentos que conforman nuestro corpus inicial entregado por el humanista.

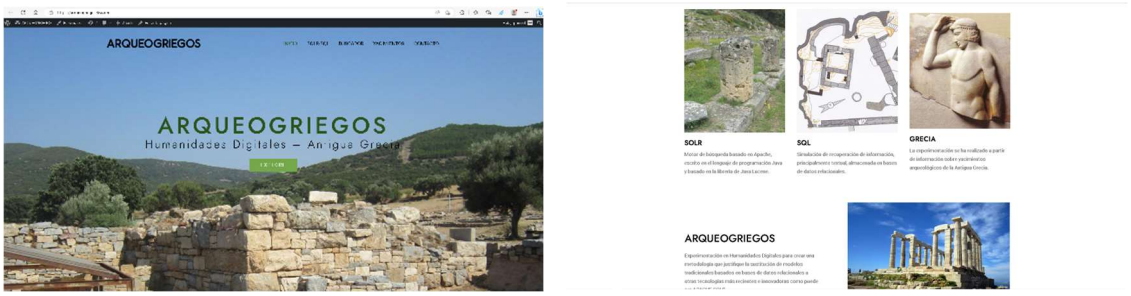


Figura 40 - Capturas de la página de inicio de la web ARQUEOGRIEGOS

En la página YACIMIENTOS accesible desde https://aqueogriegos.3isi.com/yacimientos están publicadas las 22 Entradas con la información textual y multimedia de los yacimientos de las regiones Ática y Tracia. Hay otras páginas como SOLR-SQL y CONTACTO que contienen información muy general sobre este proyecto, pero que no resultan de interés para el análisis.

En relación con la aplicación de Base de Datos y para mostrar opciones de salida más amigables para usuarios no avanzados y la realización de consultas sobre la base de datos MySQL se ha habilitado en la dirección https://arqueogriegos.3isi.com/pregunta-1 un módulo para realizar consultas directamente sobre la base de datos, como por ejemplo el término TEATRO.

Figura 41 - Consulta TEATRO con Plugin para gestión de consultas

En la página BUSCADOR accesible desde https://arqueogriegos.3isi.com/buscador se encuentra habilitado el buscador implementado por el Plugin WPSOLR que permite hacer las consultas al servidor SOLR CLOUD que tenemos instalado en esta dirección https://solr.dyanlias.com:8983

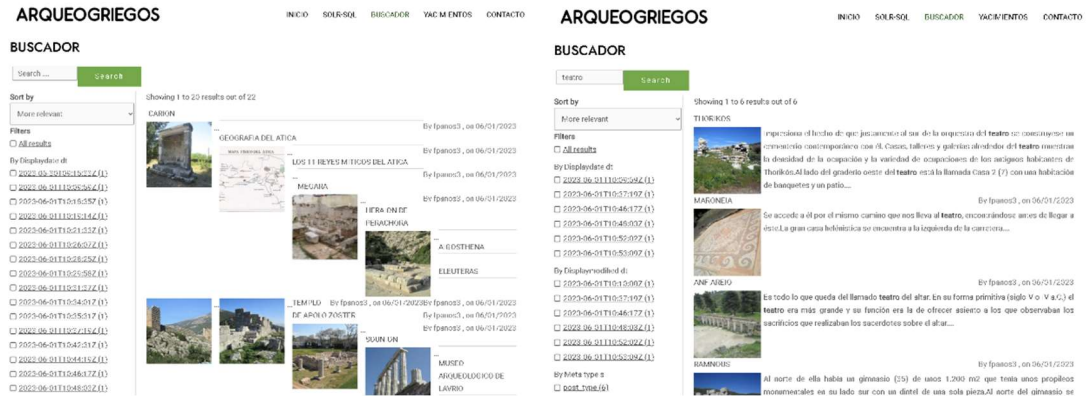


Figura 42 - Buscador de WPSOLR y Resultados para el término TEATRO

Cabe destacar la velocidad de respuesta de las búsquedas, para comprobar esta velocidad se puede hacer una búsqueda con el mismo término TEATRO en la dirección https://arqueogriegos.3isi.com/yacimientos que contiene todas las entradas almacenadas en WORDPRESS y que accede a la base de datos propia de la página web y no a la colección que se almacena en SOLR CLOUD y que es enviada desde la página al servidor SOLR mediante el Plugin WPSOLR.

Figura 43 - Búsqueda del término TEATRO en el buscador de Entradas de WORDPRESS

Como se ha indicado de manera breve anteriormente el Plugin WPSOLR permite generar una colección en el servidor SOLR de destino para posteriormente sincronizar los contenidos que se indiquen en esa colección, que en nuestro caso hemos denominado ARQUEOGRIEGOS_WOR, y usar el buscador propio del plugin para aprovechar las características de SOLR.

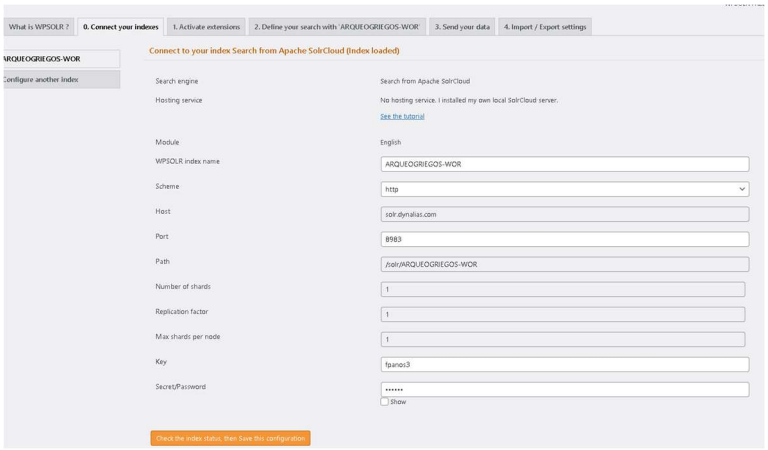


Figura 44 - Configuración del INDEX en WPSOLR

Con esta solución lo que se pretende es mejorar el aspecto final de la recuperación de los resultados de las búsquedas para acercarlas de mejor manera a usuarios no avanzados que pudieran tener más dificultades a la hora de manejar otras interfaces de consulta como pudiera ser el cliente web de SOLR o la ejecución de consultas mediante terminales de consola.

Estas comprobaciones en la interconexión entre la página web en WORDPRESS y el servidor SOLR CLOUD se han realizado con los parámetros iniciales, por lo que se intuye un amplio recorrido de mejora al ir ajustando los parámetros tanto del servidor SOLR como del Plugin WPSOLR para obtener mejores rendimientos y resultados de búsqueda. Se trata por lo tanto de una aproximación inicial, este proyecto continuará analizando las opciones disponibles en SOLR para mejorar en todos los aspectos los resultados de las búsquedas con el objetivo que estas sean lo más precisas posibles, para ello habrá que ajustar conceptos como la relevancia o el uso de funcionalidades como las Facetas que, nos permitirán realizar agrupaciones de los resultados para optimizar los resultados dados tras las búsquedas.

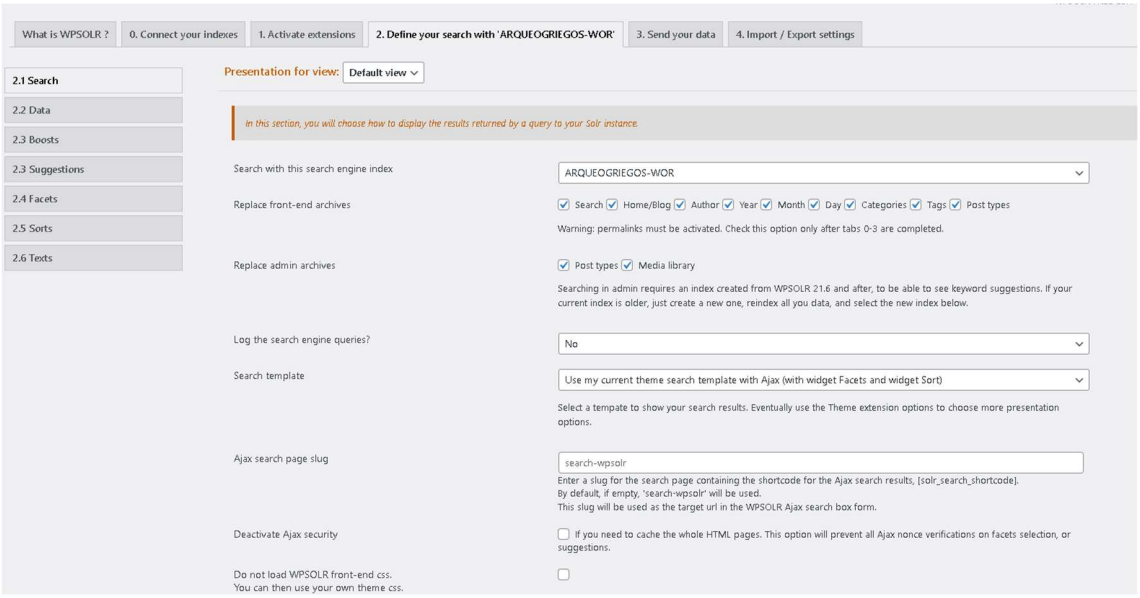


Figura 45 - Parámetros del Plugin WPSOLR

4.3. Prueba 2 HDH2023

Se presenta en lo que sigue la experimentación con un corpus ampliado con las regiones Ática, Tracia, Atenas, El Peloponeso, Sterea Ellada, Tesalia y Macedonia. Para comprobar la variación en los tiempos de las consultas se ha ampliado el catálogo de documentos de yacimientos, pasando de 22 a 151 documentos, que se han introducido en la base de datos MySQL arqueogriegos_sql. A esta nueva base datos se ha llamado **arqueogriegos_sqlx**

Recordamos la pregunta es: ¿Cuál es el conjunto de teatros que hay en una determinada región de las que está dividido el estudio?

TIEMPO CONSULTA: 0 - 5 (total de 6, La consulta tardó 0,0270 segundos.) o 27,0 milisegundos sobre un CORPUS con 1.648.079 caracteres y 286.408 palabras

En la siguiente tabla se pueden ver los tamaños de los documentos usados, los Corpus y los tiempos de ejecución para cada una de las dos consultas realizadas a las dos bases de datos MySQL, así como los porcentajes de incremento de los tres parámetros (documentos, corpus, tiempo) tras la realización de las consultas

			CORPUS			TIEMPOS		INCREMENTOS		
ID	RECURSO	TIPO	DOCS	CARACTERES	PALABRAS	SEGUNDOS	MILISEGUNDOS	DOCUMENTOS	CORPUS	TIEMPO
1	arqueogriegos_sql	BBDD MySQL	22	277823	48254	0,0066	6,6			
2	arqueogriegos_sqlx	BBDD MySQL	151	1648079	286408	0,0270	27,0	586%	309%	475%

Tabla 10 - Muestra de los incrementos de parámetros DOCUMENTOS, CORPUS y TIEMPO en consultas a bases de datos MySQL

Del mismo modo que en el caso anterior, en lugar de ampliar la Colección alojada en el servidor Apache SOR que nombramos como ARQUEOGRIEGOS-ESQ, se procederá a la creación de una nueva Colección que llamaremos ARQUEOGRIEGOS-ESX en la que se insertaran los 151 documentos que conforman el Corpus, pasando igualmente de 22 a 151 documentos, esta Colección se crea con el esquema predefinido que se creó con anterioridad y que llamamos **._designer_sql.**

Una vez creada la colección debemos importar los documentos previamente convertidos a formato .XML, como se ha indicado.

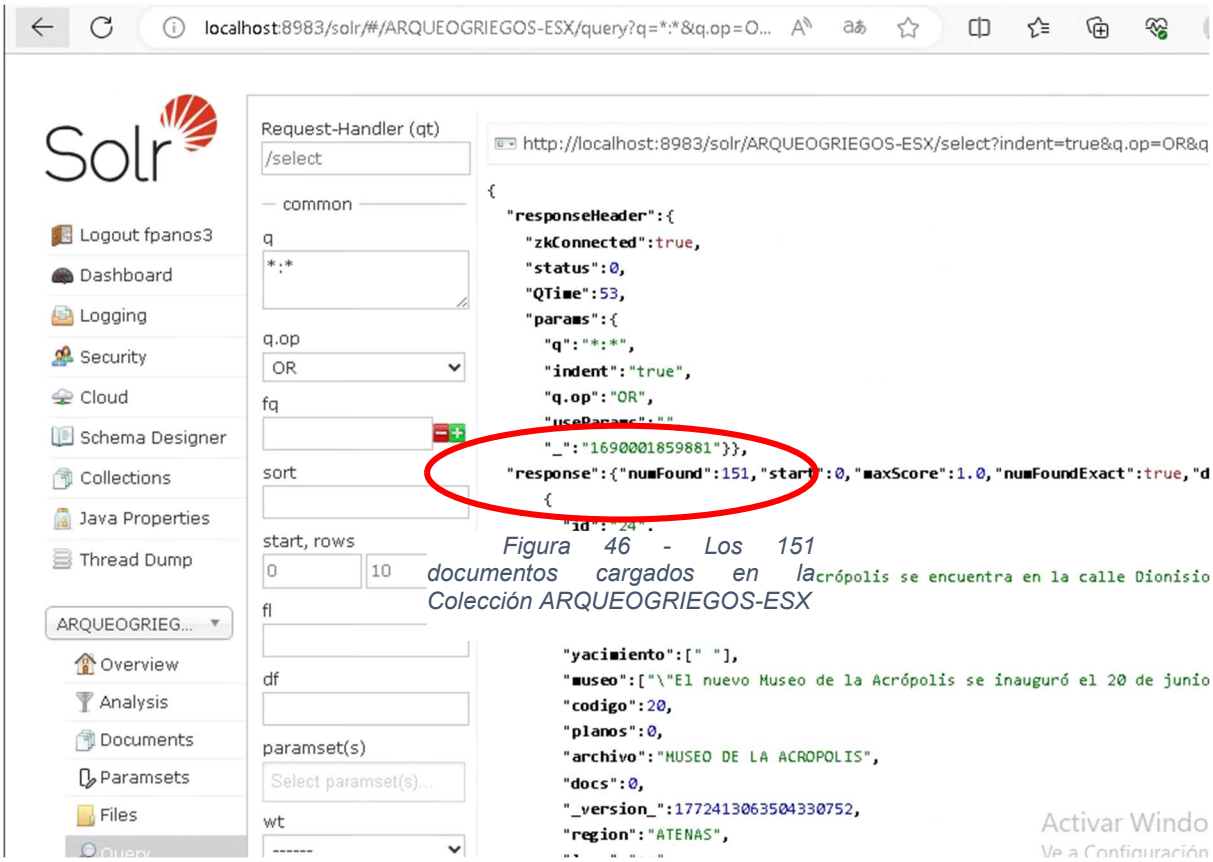


Figura 46 - Los 151 documentos cargados en la Colección ARQUEOGRIEGOS-ESX

Se repetirá la consulta en la se busca el término “teatro” dentro del campo Yacimiento de la colección de SOLR y en la que usábamos las opciones de destacar (highlighting) para reseñar la frase exacta en la que se producía la coincidencia.

Habilitamos también la opción de DEBUG para obtener los tiempos de ejecución de la consulta y finalmente ejecutamos la consulta.

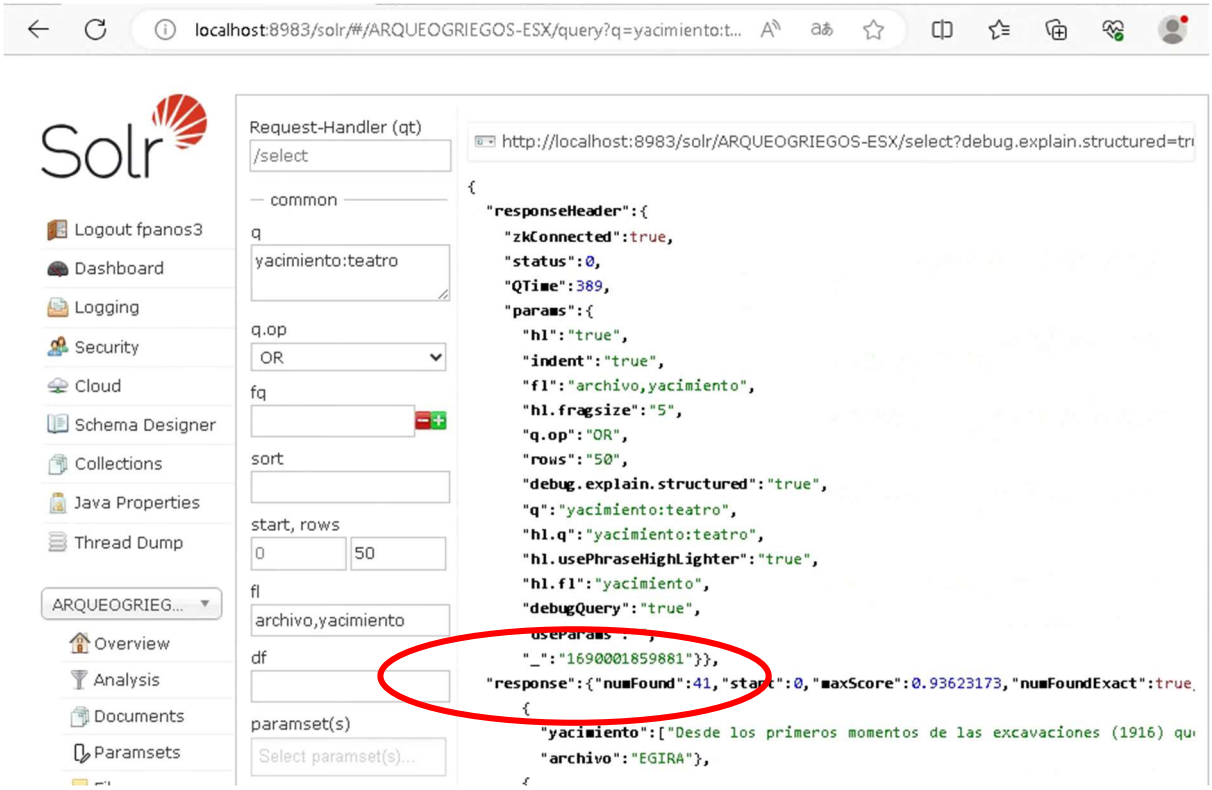
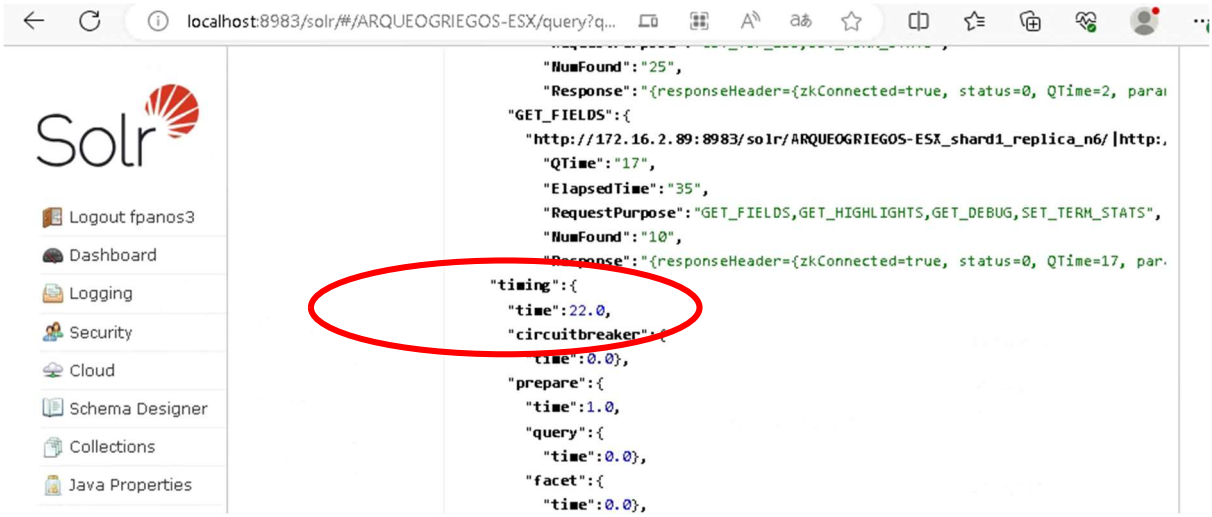


Figura 47 - Ejecución de consulta en Colección ARQUEOGRIEGOS-ESX

En este caso se han obtenido un total de 41 coincidencias del término “teatro”, y el tiempo de ejecución se puede comprobar en las capturas de pantalla siguientes.



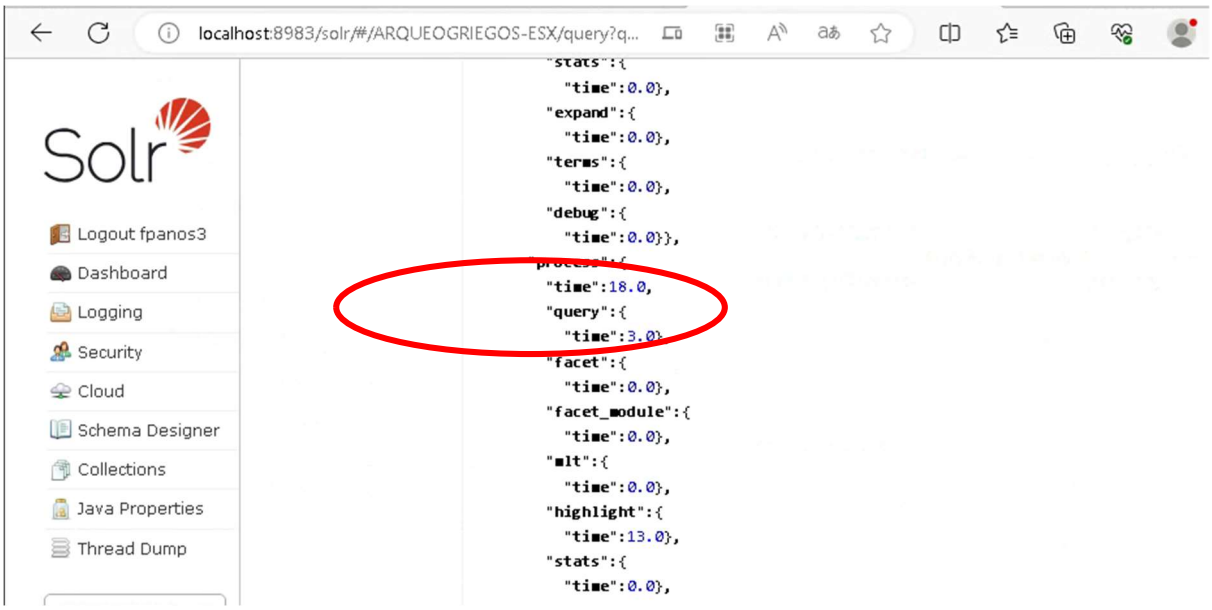


Figura 48 - Tiempos de ejecución de la consulta en ARQUEOGRIEGOS-ESX

Esta consulta la podemos reproducir ejecutando este comando HTTP contra el servidor que aloja a Apache SOLR:

```
http://solr.dynalias.com:8983/solr/#/ARQUEOGRIEGOS-
ESX/query?q=*:*&q.op=OR&indent=true&fl=archivo,yacimiento&df=yac
imiento&debugQuery=true&debug.explain.structured=true&hl=true&hl
.fl=yacimiento&hl.usePhraseHighLighter=true&hl.q=yacimiento:teat
ro&hl.fragsize=5&fq=yacimiento:teatro&useParams=
```

TIEMPO CONSULTA: la consulta ha dado como valor de time 22 milisegundos para la consulta que incluiría 18 milisegundos para el apartado PROCESS, que a su vez incluye 2 milisegundos más para el apartado DEBUG, 13 milisegundos para el apartado HIGHLIGHTS y 3 milisegundo para el apartado QUERY. Si descartamos el tiempo de DEBUG de 2 milisegundos tendremos 20 milisegundos, por lo que podemos cifrar el tiempo de la consulta en 0,020 segundos o 20 milisegundos sobre un CORPUS con 1.648.079 caracteres y 286.408 palabras.

Si bien los resultados de los tiempos de las consultas era algo que se podía intuir al leer las prestaciones de Apache SOLR, la verdad es que resulta gratamente sorprendente comprobarlos, y en casos enque la información que se vaya incorporando a una colección sea muy grande, los tiempos de ejecución de las consultas se incrementarán en porcentajes posiblemente moderados y que en el caso de bases de datos SQL serían muchísimo mayores y por lo tanto inasumibles.

En cualquier caso, estas pruebas ponen de manifiesto claramente la idoneidad de esta tecnología para proyectos en los que el volumen de información en formato de texto es grande o muy grande.

			CORPUS			TIEMPOS		INCREMENTOS		
ID	RECURSO	TIPO	DOCS	CARACTERES	PALABRAS	SEGUNDOS	MILISEGUNDOS	DOCUMENTOS	CORPUS	TIEMPO
3	ARQUEOGRIEGOS-ESQ	Colección SOLR	22	277823	48254	0,0200	20,0			
4	ARQUEOGRIEGOS-ESX	Colección SOLR	151	1648079	286408	0,0200	20,0	586%	309%	0%

Tabla 11 - Incrementos de parámetros DOCUMENTOS, CORPUS y TIEMPO en consultas a Colecciones de Apache SOLR

Todos los archivos que se han generado de una manera u otra están disponibles para quien los quiera comprobar, pudiendo reproducir los resultados obtenidos con ellos en todas las pruebas que se han descrito en este apartado.

Capítulo 5: Conclusiones y trabajos futuros

Este proyecto no solo ha tratado de dar solución a una necesidad muy específica planteada por un humanista, cuyos requerimientos pudieran ser comunes a otros proyectos similares en mayor o menor grado, sino que también ha llevado a crear un sistema informático que contempla el mayor espectro posible de necesidades de proyectos de Humanidades Digitales.

En especial, para el caso de la documentación textual, ha quedado mostrado de manera sucinta que tecnologías como los buscadores e indexadores con Apache SOLR, pueden aportar un valor añadido a la gestión de este tipo de información para que la recuperación de la información se haga de manera más ágil y eficiente, y los usuarios consumidores de estos productos de acceso a sistemas de información tengan experiencias de usuario mucho más amigables de lo que han venido siendo hasta ahora.

5.1. Resumen de las contribuciones del trabajo

El primer objetivo global de este trabajo iba más allá de encontrar una solución de almacenamiento más sofisticada que la disponible originalmente, la realizada por el estudioso humanista creador de la colección, que ya era de mucha calidad desde el punto de vista de la organización de contenidos. El objetivo era disponer de un sistema de navegación y búsqueda, teniendo en cuenta que la colección es de gran tamaño y que las tecnologías que dan soporte a estas tareas son o las bases de datos (como Access o MySQL de acceso libre) o las basadas en indexación como SOLR y su conexión con formas de visualización soportadas por páginas web, sistemas de geolocalización y otras.

Como se ha indicado en el capítulo 2, el almacenamiento y la gestión de datos más usado hasta la primera década del siglo XXI son las bases de datos relacionales. El proyecto MUSIVARIA HD usa este modelo de base de datos, en concreto usa el gestor de bases de datos MySQL (de libre acceso). Pero cuando los datos no tienen una estructura definida, como es el caso que se presenta, los conceptos de tablas, campos, registros, de las bases de datos relacionales pierden su razón de ser y se necesitan otros conceptos de almacenamiento basados en la indexación de diversos objetos.

SOLR (*Searching On Lucene Replication*) es un motor de búsqueda basado en Apache y en la librería LUCENE, escrito en JAVA y de código abierto (https://solrtutorial.es/). Entre sus características principales se destaca la posibilidad de restringir las búsquedas mediante filtrado, la búsqueda por facetas que ofrece sugerencias de filtrado, la clasificación de los

resultados de las búsquedas, las búsquedas por sinónimos, o la integración con bases de datos. Solr funciona recorriendo los documentos seleccionados e incorporándonos a un índice, indexado, añadiendo las palabras clave de los documentos. Este índice acepta datos de muchas fuentes, tales como archivos XML, CSV, archivos Word o PDF. SOLR en lugar de buscar en el texto mismo, realiza la búsqueda de la palabra clave del índice, y a continuación indica en qué documentos se encuentra dicha palabra clave.

El trabajo realizado para el proyecto ARQUEOGRIEGOS en sus etapas de gestión y almacenamiento de la documentación y desarrollo de un sistema de búsqueda para la visualización de los documentos, permite tener una comparación experimental entre ambas tecnologías para proyectos en Humanidades Digitales. Se ha realizado una implementación con ambas tecnologías del sub-corpus seleccionado para el estudio inicial incluye todos los datos sobre las regiones de Ática y Tracia. Los objetos son documentos de texto (en formato .doc), documentos de hoja de cálculo (en formato .xls), archivos de imagen (en formato .jpg) y archivos AutoCAD (en formato .dwg). Una vez probada la viabilidad de la propuesta, en el futuro, se podrán incluir más regiones.

5.1.1 Implementación

Se ha implementado una base de datos relacional en la que insertar la información de los documentos con las siguientes fases.

Diseño de la estructura de la base de datos, atendiendo a la distribución de los documentos entregados por el humanista sobre la región Ática se puede observar una estructura bien definida en cada uno de los documentos de Word que hay por yacimiento. Esta estructura está compuesta por los campos que podemos ver en la Tabla 8 - Estructura de los documentos de Ática

Estos campos serán llevados a la estructura de la base de datos. Así se creará una relación llamada Yacimientos en la que se crearán campos de identificación como ID, CODIGO, REGION, ARCHIVO y otros campos como INTRO, ACCESO, HISTORIA, MITOLOGIA, YACIMIENTO y MUSEO que serán los que contendrán la información textual como tal. Además de esta relación se crean otras tres relaciones llamadas FOTOS, PLANOS y DOCS que contendrán las rutas de acceso de los demás recursos aportados por el humanista como son las fotografías obtenidas desde cámaras fotográficas, planos digitalizados y otros documentos transformados a formato .jpg

Formato de los datos a insertar en las relaciones, los datos aportados por el humanista necesitan ser adaptados para poder ser insertados en las relaciones de la base de datos

relacional que hemos creado para tal efecto. Así, los documentos .doc se convierten a texto plano, o a .xml. Para insertarlos hay que tener cuidado con el formato del tipo de codificación para que los textos se mantengan de manera íntegra y no se produzcan sustituciones de unos caracteres por otros. Seguidamente se recopila toda la información en un archivo común (.csv) que permite importar los datos a la base de datos. En el caso de las imágenes .jpg de las fotografías, planos y documentos hay que adaptar los nombres de los archivos para que sean accesibles desde enlaces de hipervínculo.

Para habilitar el acceso a la base de datos en primer lugar necesitamos una plataforma que la almacene y posteriormente nos permita acceder a los datos contenidos en ella, para ello se han habilitado dos opciones que se detallan a continuación:

Servidor local, mediante la aplicación XAMPP (Apache, MySQL y PHP) del lado del servidor se dispone del motor de bases de datos MySQL para crear y consultar la base de datos (ver Figura 2 - Vista de la Relación YACIMIENTOS en servidor local).

Servidor remoto, se ha habilitado el subdominio https://arqueogriegos.3isi.com para la realización de pruebas de rendimiento en un entorno de producción.

En segundo lugar, para el diseño de un buscador web, se seleccionada la tecnología SOLR en vez de ELASTICSEARCH, porque la primera es una tecnología más orientada a textos y la segunda a búsquedas analíticas, la primera además tiene mayor capacidad para procesar grandes cantidades de dato y es de código libre gratuito. La configuración del sistema implementado ha seguido los siguientes pasos.

Paso 1, creación de la colección: Si los documentos que necesita indexar están en formato binario, como Word, Excel, PDF, etc., Solr incluye un controlador de solicitudes que utiliza Apache Tika para extraer texto para indexarlo en Solr. Para una primera experimentación, se insertan los archivos de la región Ática entregados por el estudioso tal y cual, es decir sin ningún tipo de modificación ni preprocesamiento. Sin embargo, esta solución sencilla, no permite explotar al máximo la organización subyacente. Por ello se decide crear la colección con un esquema personalizado, ARQUEGRIEGOS-ESQ, con un diseño interactivo a través de una interfaz web de SOLR.

Paso 2, selección del modo de esquema, una vez decidida la creación de un esquema personalizado para una colección determinada, la siguiente decisión a tomar es la granularidad de los documentos que se insertaran en la colección para su indexación y es clave pensar en lo que los usuarios del sistema querrán encontrar y ver en las respuestas a sus búsquedas [Granger y Potter, 2014]. En el sub-corpus de trabajo inicial de documentos de las regiones de Ática y Tracia, los textos son grandes y se organizan en apartados como introducción, acceso, historia, mitología, museo...

Esta estructuración subyacente proporciona una base para plantear una granularidad media a la hora de definir los campos del esquema. No sería una estrategia buena la de introducir en un único campo todo el texto de un documento (granularidad alta), al igual que tampoco lo seria definir multitud de campos para clasificar por ejemplo los elementos de características de las construcciones de los yacimientos (tipos de columnas, tipos de edificios…), pues los propios textos dan descripciones muy generales más que detalles. Si fuera al contrario, llevaría a usar otro tipo de granularidad más baja, muy similar a como podría ser un proyecto en el que predominaran datos de tipo eminentemente numéricos, y para el que probablemente hubiéramos escogido otra tecnología de almacenamiento, indexación y búsqueda como *ElasticSearch*.

Será importante también en la definición del esquema diferenciar qué campos se indicarán como indexados y cuáles como almacenados, los primeros son aquellos que son determinantes desde la perspectiva del proceso de búsqueda y los segundos aquellos que no, pero que igualmente son útiles para mostrar los resultados de una búsqueda.

SOLR crea automáticamente el esquema de campos decidiendo los parámetros de cada uno de ellos, ver la Tabla 9 - Tipos y agrupamientos de campos

La asignación automática de los tipos de campo de SOLR, es de cuatro tipos:

- Campo de identificación del documento, en este grupo se ha clasificado a campo ID
- Campos de gestión del documento, en este grupo se encuadrado los campos CODIGO, REGION, ARCHIVO, TIMESTAMP y LANG que son campos que básicamente contienen información relacionada con el propio archivo y que no contiene un texto interesante para el análisis de este
- Campos de información del documento, se encuadran los campos INTRO, ACCESO, HISTORIA, MITOLOGIA, YACIMIENTO y MUSEO que como ya sabemos, o intuimos, son los campos en los que hemos incluido la información textual de los documentos
- Campos numéricos del documento, se encuadran los campos FOTOS, PLANOS y DOCS que contienen datos numéricos sobre el número del elemento de esos tipos (foto, planos y docs) que se entregaron para cada uno de los yacimientos.

En caso de que la asignación automática de campos no fuera considerada la más adecuada, existe la opción de crear los campos manualmente desde el diseñador de esquemas uno a uno y especificando las características para cada uno de ellos. En este proyecto la distribución de la información aportada por el estudioso está muy definida y los formatos en los que se almacena esa información también, con lo que la clasificación automática es óptima y ha detectado automáticamente correctamente todos y cada uno de los campos.

A continuación, se ingresan cada uno de los documentos de la región Ática en el sistema (16). Además, se insertan los 6 documentos de la región Tracia, que permitirán hacer pruebas sobre el sub-corpus de trabajo (habilitado un acceso remoto en la dirección http://solr.dynalias.com:8983).

5.1.2 Resumen de las pruebas

Con respecto a la búsqueda "¿cuál es el número de teatros que hay en una región determinada? si utilizamos la herramienta VOYANT (ver figura 2), y manualmente estudiamos los documentos (22), se comprueba que la palabra "teatro" aparece en los yacimientos de THORIKOS, ICARION, RAMNOUS, ANFIAREIO en Ática y ABDERA y MARONEIA de Tracia. Se comprueba que salvo ICARION (uso genérico del término), todos tienen restos de un teatro ver Figura 32 - Capturas de la web de VOYAN TOOLS (Tendencias y Contextos).

Si buscamos la palabra "teatro" en la base de datos implementada, se obtiene la respuesta de la Figura 34 - Consulta del término TEATRO en la BBDD arqueogriegos_sql:

Se producen también 6 coincidencias, es decir el término aparece en 6 yacimientos, a partir de la consulta (que exige conocimiento no básico del lenguaje SQL):

```
SELECT COUNT(*) AS num_coincidencias FROM yacimientos WHERE
(CONVERT(`id` USING utf8) LIKE '%teatro%' OR CONVERT(`codigo`
USING utf8) LIKE '%teatro%' OR CONVERT(`region` USING utf8) LIKE
'%teatro%' OR CONVERT(`archivo` USING utf8) LIKE '%teatro%' OR
CONVERT(`intro` USING utf8) LIKE '%teatro%' OR CONVERT(`acceso`
USING utf8) LIKE '%teatro%' OR CONVERT(`historia` USING utf8)
LIKE '%teatro%' OR CONVERT(`mitologia` USING utf8) LIKE
'%teatro%' OR CONVERT(`yacimiento` USING utf8) LIKE '%teatro%' OR
CONVERT(`museo` USING utf8) LIKE '%teatro%' OR CONVERT(`fotos`
USING utf8) LIKE '%teatro%' OR CONVERT(`planos` USING utf8) LIKE
'%teatro%' OR CONVERT(`docs` USING utf8) LIKE '%teatro%');
```

Para que el usuario pueda entender el contexto encontrado, hay que leer el campo por completo, pero se accede a él de manera más engorrosa que en el caso anterior, que lo mostraba en pantalla.

Finalmente, para el sub-corpus ARQUEOGRIEGOS disponible en un servidor SOLR, accesible desde la https://solr.dynalias.com:8983 (necesario solicitar un usuario y una contraseña), se configura la consulta QUERY con los parámetros de la Figura 36 - Búsqueda en la Colección ARQUEOGRIEGOS_ESQ del término TEATRO

Para mejorar la visualización anterior se ha transformado la respuesta a formato JSON, que posteriormente será visualizado en una aplicación online como

https://jsonviewer.stack.hu. Finalmente se ha implementado una página web para este proyecto con el administrador de contenidos WORDPRESS y con el Plugin WPSOLR https://www.wpsolr.com. La visualización de la pregunta se muestra en la Figura 42 - Buscador de WPSOLR y Resultados para el término TEATRO.

La mejora de visualización con respecto a la visualización de los resultados en una base de datos es notoria, pues la inspección es directa (con o sin página web). También se puede asociar a la base de datos una página web para mostrar los resultados de las búsquedas SQL, pero en SOLR es posible además implementar que se muestren todas las apariciones del término en cada documento. Finalmente hay que indicar que cualquier comparación relacionada con los tiempos de acceso o ejecución de consultas en ambas tecnologías también beneficia la segunda aproximación.

5.2. Posibles mejoras y trabajos futuros

En algunos de los apartados de esta memoria ya se ha puesto de manifiesto que sería muy apropiado desarrollar aplicaciones o herramientas para automatizar algunos de los procesos que se han visto necesarios en el desarrollo de este proyecto. Algunas de las mejoras al proyecto y posibles líneas de trabajo futuras para obtener un producto más completo serían las siguientes:

- Diseño, desarrollo e implementación de una aplicación que genere archivos .XML, u otros formatos como JSON o XML, con un esquema personalizado para su posterior agregación a SOLR. Para conseguir los archivos .XML a partir de la documentación aportada por el humanista para este proyecto, se han realizado una serie de tareas intermedias que si bien no resultan ser especialmente complicada sería muy adecuado crear una aplicación que usara técnicas actuales para capturar la información y traspasarla a los esquemas predefinidos que se agregarían posteriormente en a las colecciones de Apache SOLR.

- Diseño, desarrollo e implementación de una aplicación web que provea una interfaz más amigable para sustituir a la interfaz web de Apache SOLR, una especia de constructor de consultas que provoque una mejor experiencia de usuario. La interfaz web o la API que provee Apache SOLR si bien es muy eficiente y tiene una curva de aprendizaje moderada, si bien puede suponer un hándicap importante para usuarios que vienen del campo de las humanidades y cuyos conocimientos de este tipo de herramientas les pueden suponer una barrera insalvable. Por ello, la creación de una aplicación web que abstrajera esa barrera e hiciera invisibles los detalles de esta podría ser muy beneficioso para los usuarios finales de este tipo de aplicaciones, aunque habría que tener un conocimiento óptimo de las opciones de consulta y gestión de los datos contenidos en las colecciones para saber identificar que opciones tienen valor añadido y cuales no, para implementar la nueva interfaz web que fuera más intuitiva para todo tipo de usuarios.

- Diseño del proceso a realizar para incluir otros elementos en las colecciones de SOLR, en concreto la gestión de imágenes .JPG, archivos .DWG y otros archivos que pudieran ampliar proyectos de este tipo y sería uno de los trabajos futuros.

- Abordar otras aplicaciones de visualización como por ejemplo una APP para smartphone, pues si bien las dos soluciones aportadas, en especial la página web con WORDPRESS, están diseñadas con los nuevos estándares de adaptación de los sistemas a los dispositivos de visualización, la tendencia actual es la de proveer acceso desde las aplicaciones que mejor experiencia de usuario ofrezcan al usuario y al dispositivo que usen en cada ocasión.

Bibliografía

[1] Á. Castellanos González and A. García Serrano, "Representación y Organización de documento digitales: Detalles y Práctica sobre la Ontología DIMH," 2017.

[2] A. Cámara Muñoz, "El Dibujante Ingeniero al servicio de la Monarquía Española. Siglos XVI-XVIII," 2016.

[3] B. Garrido Ramos, "Iconografía musivaria en la Península Ibérica en época romana: investigación y difusión desde el campo de las Humanidades Digitales," 2019.

[4] A. García Serrano and A. Menta Garuz, "Historia del Arte y Humanidades: Organizando datos y textos para su análisis y exploración," 2022.

[5] N. Ortega Rodríguez, "Desarrollos digitales de la Historia del Arte: implicaciones epistémicas, críticas y metodológicas," *Universidad de Málaga,* 2016.

[6] S. Schreibman, R. Siemens and J. Unsworth, A companion to digital humanities, 2004.

[7] S. Schreibman, R. Siemens and J. Unsworth, A new companion to digital humanities, 2015.

[8] C. Fernández, "Catálogo Universal de Museos y Yacimientos Arqueológicos de la Antigua Grecia," 2022.

[9] A. García Serrano and A. Menta Garuz, "La inteligencia artificial en las Humanidades Digitales: dos experiencias con corpus digitales," 2022.

[10] A. García Serrano, "Historia del Arte y Humanidades Digitales: Técnicas y aproximaciones básicas," 2022.

[11] A. G. S. y. A. M. Garuz, "Historia del Arte y Humanidades Digitales: Proyectos en Humanidades Digitales," 2022.

[12] WIKIPEDIA, "WIKIPEDIA," [Online]. Available: https://es.wikipedia.org/wiki/Apache_Cassandra. [Accessed 03 2023].

[13] WIKIPEDIA, "WIKIPEDIA," [Online]. Available: https://es.wikipedia.org/wiki/MongoDB. [Accessed 03 2023].

[14] SOLRTUTORIAL, "SOLRTUTORIAL.ES," [Online]. Available: https://solrtutorial.es/. [Accessed 03 2023].

[15] DATOS.GOB.ES, "DATOS.GOB.ES," [Online]. Available: https://datos.gob.es/es/blog/probamos-spacy-mucho-mas-que-una-libreria-para-crear-proyectos-reales-de-procesamiento-del. [Accessed 03 2023].

[16] W. O. SOLR, "Apache SOLR," 2023. [Online]. Available: https://solr.apache.org/guide/solr/latest/index.html. [Accessed 05 2023].

[17] TUTORIALSPOINT.COM, "TUTORIALSPOINT.COM," [Online]. Available: https://www.tutorialspoint.com/apache_solr/apache_solr_quick_guide.htm. [Accessed 08 2023].

[18] SOLR, "SCHEMA API," [Online]. Available: https://solr.apache.org/guide/solr/latest/indexing-guide/schema-api.html. [Accessed 12 05 2023].

[19] T. Granger and T. Potter, Solr in action, 2014.

[20] "HDH2023," [Online]. Available: https://hdh2023.org/. [Accessed Mayo 2023].

[21] "DRUPAL.ORG," [Online]. Available: https://www.drupal.org/project/search_api_solr. [Accessed Mayo 2023].

[22] "WORDPRESS.ORG," [Online]. Available: https://es.wordpress.org/plugins/wpsolr-free/. [Accessed Mayo 2023].

[23] A. García Serrano and A. Menta Garuz, "Historia del Arte y Humanidades: Proyectos en Humanidades Digitales," 2002.

ANEXOS

Anexo 1: Estructura de la Base de Datos Relacional ARQUEOGRIEGOS_SQL

```
-- phpMyAdmin SQL Dump
-- version 5.1.0
-- https://www.phpmyadmin.net/
-- Servidor: 127.0.0.1
-- Tiempo de generación: 16-05-2023 a las 11:50:24
-- Versión del servidor: 10.4.19-MariaDB
-- Versión de PHP: 8.0.6
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
-- Base de datos: `arqueogriegos_sql`
--------------------------------------------------------
-- Estructura de tabla para la tabla `docs`
CREATE TABLE `docs` (
  `id` int(11) NOT NULL,
  `yacimiento_id` int(11) DEFAULT NULL,
  `nombre` varchar(255) DEFAULT NULL,
```

```
  `ruta` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-- --------------------------------------------------------
-- Estructura de tabla para la tabla `fotos`
CREATE TABLE `fotos` (
  `id` int(11) NOT NULL,
  `yacimiento_id` int(11) DEFAULT NULL,
  `nombre` varchar(255) DEFAULT NULL,
  `ruta` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-- --------------------------------------------------------
-- Estructura de tabla para la tabla `planos`
CREATE TABLE `planos` (
  `id` int(11) NOT NULL,
  `yacimiento_id` int(11) DEFAULT NULL,
  `nombre` varchar(255) DEFAULT NULL,
  `ruta` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-- --------------------------------------------------------
-- Estructura de tabla para la tabla `yacimientos`
CREATE TABLE `yacimientos` (
  `id` int(11) NOT NULL,
  `codigo` int(11) DEFAULT NULL,
  `region` varchar(255) DEFAULT NULL,
  `archivo` varchar(255) DEFAULT NULL,
  `intro` text DEFAULT NULL,
  `acceso` text DEFAULT NULL,
  `historia` text DEFAULT NULL,
```

```
  `mitologia` text DEFAULT NULL,
  `museo` text DEFAULT NULL,
  `fotos` int(11) DEFAULT NULL,
  `planos` int(11) DEFAULT NULL,
  `docs` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-- Índices para tablas volcadas
-- Indices de la tabla `docs`
ALTER TABLE `docs`
  ADD PRIMARY KEY (`id`),
  ADD KEY `yacimiento_id` (`yacimiento_id`);
-- Indices de la tabla `fotos`
ALTER TABLE `fotos`
  ADD PRIMARY KEY (`id`),
  ADD KEY `yacimiento_id` (`yacimiento_id`);
-- Indices de la tabla `planos`
ALTER TABLE `planos`
  ADD PRIMARY KEY (`id`),
  ADD KEY `yacimiento_id` (`yacimiento_id`);
--
-- Indices de la tabla `yacimientos`
ALTER TABLE `yacimientos`
  ADD PRIMARY KEY (`id`);
-- AUTO_INCREMENT de las tablas volcadas
-- AUTO_INCREMENT de la tabla `docs`
ALTER TABLE `docs`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
--
```

```
-- AUTO_INCREMENT de la tabla `fotos`
ALTER TABLE `fotos`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT de la tabla `planos`
ALTER TABLE `planos`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT de la tabla `yacimientos`
ALTER TABLE `yacimientos`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;
-- Restricciones para tablas volcadas
-- Filtros para la tabla `docs`
ALTER TABLE `docs`
  ADD CONSTRAINT `docs_ibfk_1` FOREIGN KEY (`yacimiento_id`) REFERENCES `yacimientos` (`id`);
-- Filtros para la tabla `fotos`
ALTER TABLE `fotos`
  ADD CONSTRAINT `fotos_ibfk_1` FOREIGN KEY (`yacimiento_id`) REFERENCES `yacimientos` (`id`);
-- Filtros para la tabla `planos`
ALTER TABLE `planos`
  ADD CONSTRAINT `planos_ibfk_1` FOREIGN KEY (`yacimiento_id`) REFERENCES `yacimientos` (`id`);
COMMIT;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

"Todo son datos…"

A mi padre

Fin de la memoria: 6 de septiembre de 2023