Universidade de Aveiro
2022

Rafael Gomes de Sá

Infraestrutura de Dados Espaciais para Arqueologia

Spatial Data Infrastructure for Archaeology

Universidade de Aveiro
2022

Rafael Gomes de Sá

Infraestrutura de Dados Espaciais para Arqueologia

Spatial Data Infrastructure for Archaeology

Relatório de Projeto apresentado à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Doutor José Manuel Matos Moreira, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Engenheiro Luís Jorge dos Santos Gouveia Marques Gonçalves, Professor Adjunto da Escola Superior de Tecnologia e Gestão de Águeda, com colaboração do Doutor Jorge Manuel Pessoa Girão Medina, Professor Auxiliar do Departamento de Geociências da Universidade de Aveiro.

Este trabalho foi parcialmente financiado pelo Fundo Europeu de Desenvolvimento Regional (FEDER), Programa Operacional Competitividade e Internacionalização no contexto do projeto Odyssey - Platform for Automated Sensing in Archaeology, ALG-01-0247-FEDER-070150.

o júri / the jury

presidente / president	Prof. Doutor ... professor associado da Faculdade ...
vogais / examiners committee	Prof. Doutor ... professor associado da Faculdade ...
	Prof. Doutor ... professor associado da Faculdade ...

agradecimentos / acknowledgements

Um grande obrigado,
Ao meu pai, por todo o esforço e sacrifício para tornar tudo isto possível.
À minha mãe, que mesmo não estando aqui, contribuiu para que eu conseguisse atingir os meus objetivos.
À minha irmã, por todo o apoio incondicional.
À Daniela, por todo o apoio e paciência ao longo destes anos.
Ao David e ao Francisco, pelo apoio, companheirismo e amizade ao longo destes anos.

Um agradecimento especial aos meus orientadores, os professores José Moreira, Luís Jorge Gonçalves e Jorge Medina, por todo o apoio, confiança e capacidade de motivar e ensinar.

Agradeço aos membros do projeto Odyssey, pelo apoio na elaboração deste trabalho.

Por último, agradeço à Universidade de Aveiro por me acolher, e aos professores com quem me cruzei por todo o conhecimento transmitido.

Palavras Chave

Arqueologia, Deteção Remota, Infraestrutura de Dados Espaciais, Sistema de Informação Geográfica Web

Resumo

O inventário português de património arqueológico é caracterizado por lacunas na informação, sendo tendencialmente escassa, omissa ou até mesmo errada. Para além disso, algumas bases de dados patrimoniais possuem um número reduzido de sítios arqueológicos. A informação existente para cada sítio é muitas vezes insuficiente, faltando por exemplo informação geográfica. Isto deve-se à dificuldade, ou até mesmo impossibilidade, de acesso a alguns locais, tornando os levantamentos patrimoniais pouco exaustivos. Para além disso, os processos arqueológicos são tradicionalmente demorados e com grande intensidade de recursos humanos, tornando esses processos mais caros e contribuindo também para a falta de informação patrimonial.

Este projeto focou-se no desenvolvimento de uma infraestrutura de dados espaciais com funcionalidades de um sistema de informação geográfica e capaz de gerir informação relativa a sítios arqueológicos, tendo sido integrada com algoritmos de *machine learning* destinados à deteção remota de sítios arqueológicos. Esta plataforma visa ainda dar suporte às atividades de validação durante o trabalho de campo, contribuindo para a melhoria dos processos de anotação de dados no terreno e, consequentemente, para a melhoria da qualidade da informação e dos algoritmos. A implementação desta plataforma foi baseada no GeoNode, uma *framework* para o desenvolvimento e implementação de infraestruturas de dados espaciais.

A avaliação preliminar da plataforma demonstrou que a plataforma corresponde às expectativas das partes interessadas do projeto. A plataforma irá permitir levantamentos patrimoniais e arqueológicos mais abrangentes, que de outra forma seria inviável, resultando num aumento de sítios arqueológicos identificados e contribuindo para uma melhor qualidade da informação arqueológica. Para além disso, vai permitir uma melhor gestão dos recursos humanos e financeiros, uma vez que as prospeções arqueológicas serão mais direcionadas às zonas com maior potencial arqueológico, previamente identificadas utilizando as ferramentas disponíveis na plataforma.

Keywords

Archaeology, Remote Sensing, Spatial Data Infrastructure, Web Geographic Information System

Abstract

The Portuguese inventory of archaeological heritage is characterized by shortcomings in information, which tends to be scarce, missing or even wrong. Furthermore, some heritage databases have a reduced number of archaeological sites. The existing information for each site is often insufficient, lacking geographic information, for example. This is due to the difficulty, or even impossibility, of access to certain locations, making heritage surveys not very exhaustive. In addition, archaeological procedures are traditionally time-consuming and human-resource intensive, making these procedures more expensive and also contributing to the lack of heritage information.

This project focused on the development of a spatial data infrastructure with geographic information system features and capable of managing information related to archaeological sites, which has been integrated with machine learning algorithms for the remote sensing of archaeological sites. This platform also aims to support validation activities during fieldwork, thereby contributing to the improvement of data annotation processes in the field and, consequently, to the improvement of the quality of the information and the algorithms. The implementation of this platform was based on GeoNode, a framework for the development and deployment of spatial data infrastructures.

The preliminary evaluation of the platform has demonstrated that it meets the expectations of the project's stakeholders. The platform will allow more extensive heritage and archaeological surveys, which would otherwise be unfeasible, leading to an increase in the number of archaeological sites identified and contributing to a better quality of archaeological information. Moreover, it will allow a more efficient management of human and financial resources, as the archaeological surveys will be more targeted on the areas with greater archaeological potential, previously identified using the tools available in the platform.

Contents

List of Figures

List of Tables

Glossary

ADS Archaeology Data Service
AOI Area of Interest
ARIADNE Advanced Research Infrastructure for Archaeological Dataset Networking in Europe
CAOP Official Administrative Map of Portugal (*Carta Administrativa Oficial de Portugal*)
COS Land Cover Land Use Map (*Carta de Ocupação de Solo*)
CRUD Create, Read, Update and Delete
CSV Comma-Separated Values
CSW Catalog Services for the Web
DBMS Database Management System
DTM Digital Terrain Model
EU European Union
FAIMS Field Acquired Information Management System
FOSS Free and Open-Source Software
GDAL Geospatial Data Abstraction Library
GEMET General Multilingual Environmental Thesaurus
GeoCMS Geospatial Content Management System
GeoTIFF Geographic Tagged Image File Format
GIS Geographic Information System
HTTP Hypertext Transfer Protocol
INSPIRE Infrastructure for Spatial Information in Europe
ISO International Organization for Standardization
LiDAR Light Detection and Ranging
LRM Local Relief Model
ML Machine Learning
MVT Model View Template
NIR Near-Infrared
OAuth Open Authorization
OGC Open Geospatial Consortium
OMT-G Object Modeling Technique for Geographic Applications
ORM Object-Relational Mapper
OSGeo Open Source Geospatial Foundation
PRAGIS Puuc Region Archaeological Geographic Information System
RGB Red Green Blue
SDBMS Spatial Database Management System
SDI Spatial Data Infrastructure
SITAR Geographic Archaeological Information System of Rome
SOA Service-Oriented Architecture
SQL Structured Query Language
SRS Spatial Reference System
TIFF Tag Image File Format
UML Unified Modeling Language
URL Uniform Resource Locator
WFS Web Feature Service
WFS-T Transactional Web Feature Service
WKT Well-Known Text
WMS Web Map Service
WMTS Web Map Tile Service
XML Extensible Markup Language

CHAPTER 1

Introduction

The development and the increase of the population has contributed to the acceleration of transformations in the landscapes, related to the expansion of urban centers, creation of access roads, among others. Cultural heritage, as a representation of our history and legacy, is relevant to modern societies, so its preservation or documentation is essential. During interventions in the territory, obtaining detailed information about the heritage in the affected area is crucial for a rigorous planning and management of the project and to reduce uncertainties in the work to be done [1].

However, the Portuguese inventory of archaeological heritage is characterized by shortcomings in information, which tends to be scarce, missing or even wrong. The information available in Portuguese heritage databases is often inconsistent among themselves and, besides that, they have a reduced number of archaeological sites with insufficient information for each site, lacking for example geographic information [1].

This situation is due essentially to the unreliability of information from heritage surveys, and the use of traditional processes and intensive labor. The lack of reliability stems from the difficulty or impossibility of accessing to certain locations, making traditional prospection work economically unviable, which leads to incomplete surveys. This unreliability causes a high number of archaeological sites to be incorrectly identified, and in many cases the actual sites are not previously identified, causing high financial costs and waste of resources in territorial interventions. Despite the technological advances that have been applied to archaeology, this is still a very traditional sector in Portugal, with very time-consuming and labour-intensive procedures, which makes the processes more expensive and less accurate, also contributing to the lack of quality of information about the archaeological heritage [1].

1.1 Context

This project is part of the research project "Odyssey - Platform for Automated Sensing in Archaeology". The Odyssey project consortium is composed by ERA - Arqueologia, the University of Aveiro and the University of Maia. ERA, as the main stakeholder in the project, aims to contribute to the design and development of innovative, automated and intelligent solutions that allow the remote identification and classification of archaeological and heritage sites. Thus, the goal of the Odyssey project is to use image processing and artificial intelligence techniques to detect archaeological sites using data from different sources, automating and complementing human work to improve results, resulting in an innovative platform of geographical information to be used by archaeologists and heritage technicians [1].

The Odyssey project uses several sources of georeferenced information, in particular multispectral imagery and Light Detection and Ranging (LiDAR) data. Multispectral imaging involves a combination of different spectral bands, such as Red Green Blue (RGB) and Near-Infrared (NIR), in order to highlight the visibility of archaeological features [2, 3]. LiDAR is a laser system that allows the creation of point cloud datasets of landscapes and surface characteristics by sending pulses of energy to the earth's surface and collecting them through the sensor [4]. LiDAR-derived products, such as Digital Terrain Models (DTMs), are useful for remote sensing in archaeology. A DTM represents the bare earth surface, removing all natural and non-natural features other than the ground [5], helping to detect archaeological findings that might have been covered by vegetation. These images can then be further processed to highlight patterns or anomalies in the landscape [6]. An example of a visualization technique used for archaeological prospection is the Local Relief Model (LRM), which removes large landscape elements to enhance the visibility of small features [7].

The georeferenced information should be consolidated into the platform, so that a comprehensive and integrated view of the territory and its different archaeological and heritage sites is obtained. Then, the platform should automatically process this information using Machine Learning (ML) algorithms, identifying archaeological sites in a predictive way based on their typology. Finally, the platform should support the ground truthing of the findings obtained by the algorithms, allowing real-time integration of the data collected in the field through a mobile version. This improvement of the field work will contribute to the enhancement of the annotation processes, and therefore to the improvement of the algorithms [1].

1.2 Motivation

The platform aims to improve the efficiency and effectiveness of the entire process of heritage and archaeological surveys, making them comprehensive that would otherwise be unfeasible for operational and economic reasons. With this, better quality information will be available, as a result of the increase in the rate of sites, and their typology, correctly identified [1].

Furthermore, the platform will increase the efficiency of ERA's operations, thus reducing economic costs and improving the management of the human resources. The field prospection will be more targeted to the areas identified as having great archaeological potential, thereby reducing the time needed for this process. This will also reduce the rate of sites detected only during the territorial intervention stage, easing its management and avoiding the waste of resources [1].

1.3 Goals

Considering that the aim of the Odyssey project is to create a platform consisting of a spatial data infrastructure that integrates a geographic information system and remote sensing capabilities, this project has two main goals.

The first goal is to specify the requirements and architecture of the platform. To do this, the platform's usage scenarios, both web and mobile, must be identified, and then used to derive its functional and non-functional requirements. Based on this, a reference architecture must be defined, detailing its components, and the data model required for the platform's operation must be specified. Finally, it should be defined how the integration with ML algorithms will be achieved.

The second goal is to implement the spatial data infrastructure and develop the geographic information system, which will consolidate the georeferenced data and integrate the ML algorithms. For this, a survey of the existing tools should be carried out, selecting those that best fit the needs of the platform. The platform must have the ability to integrate, manage, edit, display and publish geospatial data, maps, metadata and associated documents, following the standards established for this type of systems.

It should be noted that, although in the specification of requirements the web and mobile versions are considered, the definition of the architecture and the implementation of the platform is only done for the web version. The definition of the architecture and implementation of the mobile version are outside the scope of this project. Furthermore, this project focuses only on the integration of the ML algorithms, being their development also out of the scope of this project.

1.4 Document Structure

This document is divided into six chapters, including this introductory chapter. For the remaining chapters a brief description is given next.

Chapter 2 - *Spatial Data Infrastructures and Applications in Archaeology*: provides an introduction to the standards for spatial data and describes the typical architecture of a Spatial Data Infrastructure (SDI) and Geographic Information System (GIS), detailing its components and the main tools available for each of them. In addition, it presents related work relevant to this project.

Chapter 3 - *Platform Requirements and Architecture*: details the usage scenarios and requirements of the platform, as well as its architecture.

Chapter 4 - *Implementation*: contains the implementation details of the platform's functionalities.

Chapter 5 - *Evaluation*: describes the data used to test the platform, and the demonstration made to the Odyssey project stakeholders.

Chapter 6 - *Conclusions and Future Work*: presents the main results and conclusions of the project, and identifies possible future work.

CHAPTER 2

Spatial Data Infrastructures and Applications in Archaeology

This chapter provides an introduction to Web GISs and SDIs, presenting the typical architecture and detailing the components that constitute it and the tools that can be used for its implementation. As the implementation must follow standards defined for the geospatial industry, the most relevant standards for this project are described. Finally, related work is presented, including applications of Web GISs and SDIs to archaeology, mobile solutions, data catalogs and infrastructures, and integration of remote sensing with GIS.

2.1 Technologies and Architecture

A GIS is a system that can store, collect, manipulate, analyze and present geospatial data. The need to make GIS more easily accessible to users has led to the evolution towards Web GIS. Agrawal *et al.* [8] referred to a Web GIS as being a "*web application that provides GIS capabilities*". As such, this type of systems consists of at least a server and a client that uses web technology to communicate between them so that users can access geospatial data [8]. This brings advantages such as better cross-platform capability, a larger number of users, ease of maintenance, and a lower cost relative to the number of users, to name a few [9].

A SDI is defined by Hu and Li [10] as *"the technology, policies, standards, and human resources necessary to acquire, process, store, distribute, and improve utilization of geospatial data, services, and other digital resources"*. This means that an SDI is an extension of a GIS, ensuring the use of standards for interoperability and giving special emphasis to the use of metadata and sharing of geospatial data in an effective way [11, 12].

The implementation of a Web GIS and an SDI should be seen as a process, considering the available technology and the application requirements [13]. As such, the implementation must take into account the standards set by industry, and must be broken down into smaller steps where, for each one, the available tools must be analyzed.

In this section, the relevant standards for this project are presented, and the typical architecture of an SDI and Web GIS is detailed, as well as the main tools available for each of its components.

2.1.1 Standards

The Open Geospatial Consortium (OGC) is an international organization that promotes the development and implementation of standards for the interoperability of geospatial content [14]. The organization defines over sixty standards, of which the most relevant for this purpose can be grouped in categories such as discovery, services, and data models [15].

Concerning the discovery category, the standard for catalog services, in particular Catalog Services for the Web (CSW) [16], stands out. Catalog services allow the publication and searching of collections of metadata, where metadata represent properties about resources that can be queried and presented for future processing, and are essential for sharing and spreading information in an efficient way within the community. For publishing and accessing metadata catalogs for geospatial data and services, this standard defines interfaces for retrieving, discovering, searching, creating, and editing metadata [16].

Regarding services, the standards for Web Map Service (WMS) [17], Web Map Tile Service (WMTS) [18] and Web Feature Service (WFS) [19] are worth mentioning. The WMS standard defines a protocol for producing and requesting georeferenced map images from geospatial databases, which are then returned in a graphic format such as JPEG or PNG. A WMS request is characterized by the geographic layers and the area of interest that will be processed to produce the map [17]. The WMTS standard was introduced as a complement to the WMS standard, and offers a scalable solution for serving maps using predefined or pre-rendered image tiles. This solution is performance driven and allows the use of caching strategies [18]. The WFS standard provides a fine-grained access to geographic information at the feature and feature property level, which means that it is possible to retrieve or change only the information that is intended, rather than getting a file with information that is not relevant. The standard specifies various operations, as discovery or query operations, with focus on transaction operations, also known as Transactional Web Feature Service (WFS-T), which allows the creation, modification, and deletion of features from the data store, thereby supporting the development of collaborative mapping applications [19, 20].

The OGC has standard data formats for representing geospatial information, such as Geographic Tagged Image File Format (GeoTIFF) [21] for georeferenced imagery. The GeoTIFF format dates back to the 1990s [22], but the OGC has recently revised it with the purpose of its standardization. The GeoTIFF standard defines a set of geospatial metadata that enables the georeferencing of information within Tag Image File Format (TIFF) images, obtained from satellite imaging systems, scanned aerial photography, DTMs, and other sources [21]. There are also other formats that, although proprietary, are commonly used, such as the shapefile format [23]. The shapefile format is a geospatial vector data format developed by Esri[1] and stores the geometric location and attribute information of geographic

[1] `https://www.esri.com/` (last acessed Oct. 30, 2022)

features, which can be represented by points, lines or polygons [23].

Geospatial data must be accompanied by its Spatial Reference System (SRS), as only then the data can be displayed and manipulated correctly. An SRS describes unambiguously where resources are located on the Earth's surface through coordinates. The coordinates, and their format, vary depending on the SRS, so it is crucial that the data is provided with the correct SRS to interpret it properly [24]. SRS may be intended for global use, such as the standard WGS84 - EPSG:4326[2], or for local use, like the ETRS89 / Portugal TM06 - EPSG:3763[3] for Portugal.

There is also an European Union (EU) directive that concerns spatial information, the Infrastructure for Spatial Information in Europe (INSPIRE)[4] directive. One of its requirements is that the metadata to be created for spatial datasets and services must follow established themes, ensuring their standardization across the member states [25].

2.1.2 Typical Architecture

A Web GIS system usually follows a three-tier client-server architecture (see Figure 2.1) [26, 27]. The server-side is responsible for receiving, processing and replying to requests, while the client-side is responsible for performing the requests and viewing the retrieved results [28, 29]. This architecture offers, among other advantages, greater flexibility, scalability and reusability [30]. It consists of a data tier, a middle tier, and a presentation tier.

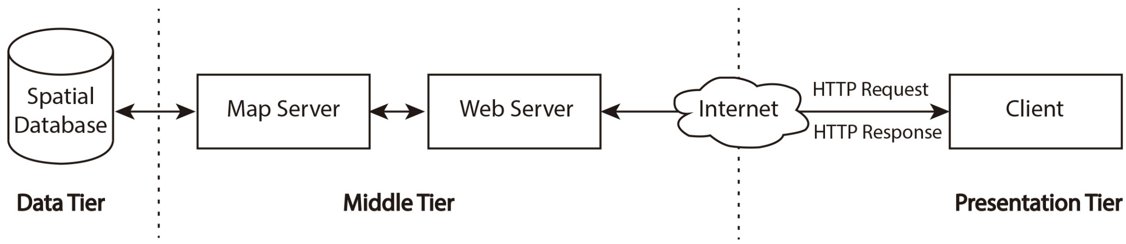

Figure 2.1: Generic three-tier Web GIS architecture [31]

The data tier is where the persistent information is stored and managed through the Database Management System (DBMS) [27]. When it comes to storing spatial data, the concept of geospatial database emerge. A geospatial database is an extension of a conventional database capable of storing, querying, and indexing spatial data [32]. Spatial data structures for data storage are formed as an extension of the basic data types into spatial data types, which are typically the ones defined in the Simple Features standard [33] from OGC, such as points, lines, and polygons [34]. Besides the typical queries in conventional databases, the standard also defines a set of spatial operations that allow, for example, measurements or intersections to be performed [35].

The middle tier, or application tier, is where the system logic is implemented. A non-spatial web application usually contains a web server, such as the popular open-source Apache

[2]`https://epsg.io/4326` (last acessed Oct. 11, 2022)
[3]`https://epsg.io/3763` (last acessed Oct. 11, 2022)
[4]`https://inspire.ec.europa.eu/` (last acessed Oct. 13, 2022)

HTTP Server[5], to answer client requests, but in the case of a Web GIS, a map server is also required [36]. This additional server is responsible for handling geospatial data and provides a set of tools for building web mapping applications and web services that are compatible with geospatial data, such as WMS and WFS [27]. Along with a map server, this tier may also include components such as a tile caching server to improve the performance (see Section 2.1.1).

The presentation tier is the interface layer that allows the user to visualize and interact with geospatial data, which can be done through a web browser, desktop application, or mobile application [8].

Apart from three-tier client-server architecture, a Service-Oriented Architecture (SOA) implementation is also used in a number of applications, particularly in SDIs. In SOA, components can play the role of provider, broker, or requestor, and are implemented independently. Generally, the provider publishes its services to the broker, which the requestor will use to discover the required services and consume them (see Figure 2.2) [36].

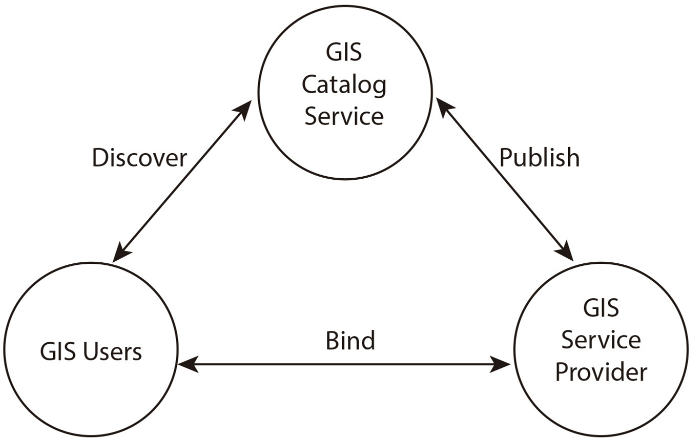

Figure 2.2: GIS Service-oriented architecture [37]

As this is a distributed, loosely coupled architecture which follows the publish-find-bind pattern, it is used as a basis to provide the SDI catalog service for discovering data and services by searching the metadata [10, 36, 37]. The key components for this are geoportals, which consists essentially of a web interface allowing the user to interact and perform searches, metadata, which provides information about the geospatial resources, and search functions, which allows users to find the geospatial resources either through text or map searches. The discovery process is highly dependent on the quality of the metadata and the effectiveness of the search function, since poor metadata or a search function with low performance makes it very hard to find the geospatial resources. Also, as the metadata is the primary source of information about the resources in the catalog, poor quality metadata means that users may not be able, or may find it very difficult, to use the resources [10].

Considering that a Web GIS is part of an SDI, there is a relation between their architectures. The catalog service is a component added to the middle tier, with the data discovered through

[5] `https://httpd.apache.org/` (last acessed Oct. 17, 2022)

the catalog being consumed by users through the services provided by the map server.

2.1.3 Tools

The main tools for each of the components presented in Section 2.1.2 are described next. There are proprietary tools that offer complete solutions, such as Esri's ArcGIS[6]. However, these tools can be costly and less flexible, and for these reasons only Free and Open-Source Software (FOSS) tools are considered.

Data Tier

There are a wide variety of Spatial Database Management Systems (SDBMSs) and DBMSs that support the spatial dimension secondarily. DB-Engines[7] is an initiative that collects information about DBMSs and provides a ranking based on their popularity updated monthly.

Considering the DB-Engines ranking[8] as of October 2022, among the DBMSs that support spatial data, MySQL [38] is the highest ranked open-source DBMS. MySQL implements spatial extensions that comply with OGC specifications, which include the standard geometric data type. The features of the spatial extensions also include spatial indexing and functions for manipulating spatial data types [38]. However, MySQL can be somewhat limited in the functions it provides compared to other SDBMSs.

If only specialized DBMSs are considered, the DB-Engines ranking[9] as of October 2022 distinguishes PostGIS [39] as the top-ranked. PostGIS, an Open Source Geospatial Foundation (OSGeo) project, is a PostgreSQL[10] extension to a spatial database compliant with the OGC's Simple Features standard [40]. PostGIS implements geometric, geographic, raster, and other data types, as well as operators and indexing enhanced for the spatial dimension. PostGIS also has hundreds of spatial functions, which stands out in comparison to MySQL [39]. For storing LiDAR point cloud data, there is the open-source extension pgPointCloud[11] that can be used with PostGIS. In addition, PostGIS has command line tools for importing and exporting data, such as osm2pgsql[12] to import data from OpenStreetMap[13], and shp2pgsql to import shapefiles.

Middle Tier

There are several open-source map servers, and some of the most widely used are OSGeo projects, in particular GeoServer [41] and MapServer [42]. GeoServer is a Java-based server that follows the OGC standards, and allows the visualization and editing of geospatial data. It allows the creation of maps, implementing the WMS standard, and the sharing of data, conforming to the WFS standard. In addition to WFS operations, GeoServer is compliant with the WFS-T specification [41]. MapServer is a geographic data rendering engine written

[6] `https://www.esri.com/en-us/arcgis/products` (last acessed Oct. 29, 2022)
[7] `https://db-engines.com/` (last acessed Oct. 16, 2022)
[8] `https://db-engines.com/en/ranking/spatial+dbms/all` (last acessed Oct. 16, 2022)
[9] `https://db-engines.com/en/ranking/spatial+dbms` (last acessed Oct. 16, 2022)
[10] `https://www.postgresql.org/` (last acessed Oct. 19, 2022)
[11] `https://pgpointcloud.github.io/pointcloud/` (last acessed Oct. 19, 2022)
[12] `https://osm2pgsql.org/` (last acessed Oct. 19, 2022)
[13] `https://www.openstreetmap.org/` (last acessed Oct. 19, 2022)

in C for publishing spatial data and interactive mapping applications to the web. MapServer complies with several OGC standards, such as the WMS and WFS standard, but is not a full-featured GIS system and, as such, does not support the WFS-T specification [42].

When it comes to the catalog service, OSGeo has two open-source projects, GeoNetwork [43] and pycsw [44]. GeoNetwork is a catalog application, compliant with the CSW standard from OGC and the INSPIRE directive, to manage spatially referenced resources. It provides metadata editing, search and harvesting functions, as well as a web map viewer [43]. Pycsw is a CSW server written in Python, which enables the publication and discovery of geospatial metadata. Pycsw is also compliant with the OGC's CSW standard and implements the INSPIRE Discovery Services [44, 45]. The main difference between the two catalog services is that pycsw does not have a graphical user interface like GeoNetwork.

As for caching, two of the main tile server implementations are GeoWebCache [46] and MapProxy [47, 48]. GeoWebCache, an OSGeo community project, is a Java web application that caches map tiles from various sources, such as WMS, acting as a proxy between the client and the server. The fact that the map images are saved as they are requested optimizes the map rendering and avoids unnecessary requests to the server [46]. MapProxy is an open-source proxy for geospatial data that caches tiles and accelerates existing map services to serve them to GIS clients, web or desktop. This tool complies with and supports the OGC standards WMS and WMTS [47]. GeoWebCache has the advantage of already being integrated into GeoServer, although it can also be used independently.

Presentation Tier

OpenLayers [49] and Leaflet [50] are two of the most popular open-source JavaScript libraries for web-based geographic applications [51]. OpenLayers is an OSGeo project and allows to display map tiles, vector data and markers loaded from any source using OGC standards [49]. Leaflet is a light-weight library for mobile-friendly interactive maps, one of the advantages of which is its simplicity and additional plugin-based features [50]. Leaflet is best suited for mobile or simpler applications, however when the number of plugins becomes too large or more complex operations, such as projections, are required, OpenLayers is more appropriate [52].

A desktop GIS application is an application that allows the visualization, query, update and analysis of geographic data and associated information and is personal computer-based [53]. There are a wide variety of open-source desktop GIS applications but QGIS [54], an OSGeo project, is the most popular one [55]. QGIS Desktop is a tool that allows the creation, editing, visualization, analysis and publication of geospatial information and supports various vector, raster and database formats. It supports OGC specifications, such as WMS, WFS, WFS-T, and others [54].

Mobile GIS is the extension of a GIS to the field, thereby allowing data changes directly in the field, increasing its accuracy. Most mobile GIS applications are extensions of desktop GIS applications [56]. QField [57] is an open source mobile application based on QGIS, meaning that it is possible to do a project setup in QGIS Desktop and deploy it in the field through the application. QField's interface is optimized for the tasks that are required to be done in the field, with all other tasks not available in the interface [57].

Geospatial Content Management System

A Geospatial Content Management System (GeoCMS) is a content management system where objects can have a geographical representation. GeoNode[14] is a GeoCMS that includes all components of the three layers described above integrated seamlessly. To the author's knowledge, GeoNode is the only active open source tool of its kind.

GeoNode is a web framework that allows the development and deployment of SDIs. It supports the uploading of vector and raster data, including shapefiles and GeoTIFFs, and handles metadata management, automatically exposing it to the catalog service for search and discovery capabilities. The users responsible for the data can assign permissions, defining which users or groups can view, edit and download the data and metadata. Once the data is uploaded, different layers, including external layers, can be combined and viewed together on thematic maps. All data are accessible through OGC standards such as WMS, WFS and CSW [58].

The GeoNode web application is developed in Django[15], a Python web framework, following the Model View Template (MVT) pattern. In an MVT architecture, as shown in Figure 2.3, Hypertext Transfer Protocol (HTTP) requests are redirected to the appropriate View by the Uniform Resource Locator (URL) mapper. The View is responsible for handling and responding to the HTTP request, accessing the required data through Models, and formatting the response through Templates.

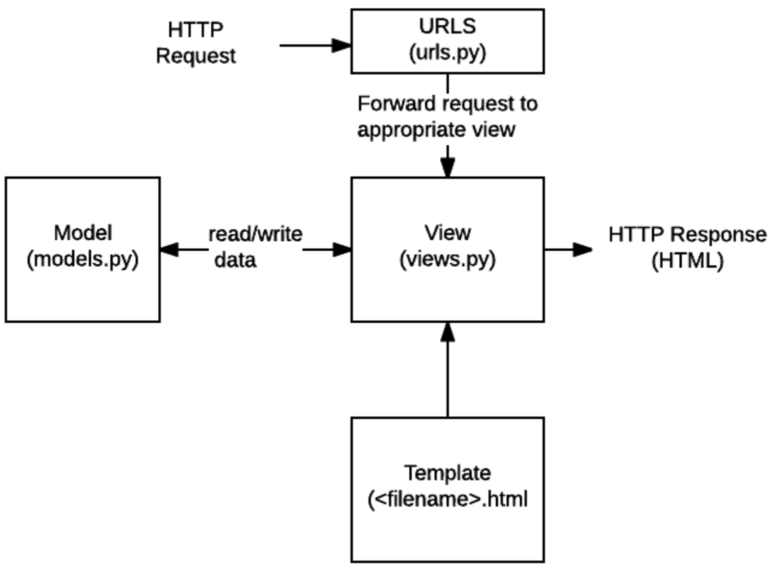

Figure 2.3: Django MVT Pattern [59]

[14]`https://geonode.org/` (last acessed Oct. 17, 2022)
[15]`https://www.djangoproject.com/` (last acessed Oct. 17, 2022)

GeoNode combines the various components described above, namely the spatial database, map server, cache server, catalog service, and the user interface. Figure 2.4 presents the GeoNode architecture. For the spatial database, PostgreSQL with PostGIS is used, although SQLite[16], or other relational DBMS supported by Django, can be used in simpler installations without a spatial database by saving the vector data as shapefiles in the file system. As a map server, GeoNode uses GeoServer, having GeoWebCache already built in as a caching server. The catalog service is implemented using pycsw, although other CSWs such as GeoNetwork can be used. For the user interface, GeoNode uses Django templates, together with JavaScript libraries such as Leaflet and OpenLayers. On top of this, MapStore[17], a web GIS framework, is used to query the data services and visualize them through interactive maps [58].

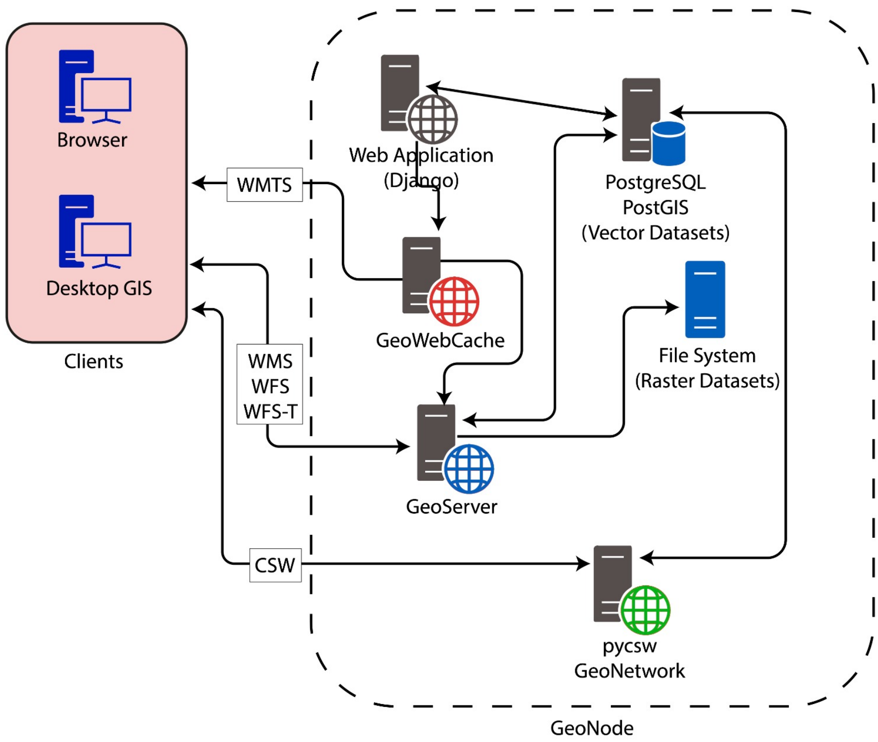

Figure 2.4: GeoNode Architecture [58]

All this combined brings great benefits, as GeoNode has all the needed components, including several functionalities in a user-friendly web application, with a focus on interoperability by implementing the OGC standards. Furthermore, being developed in Django makes it easy to extend and modify its functionalities [60]. On the other hand, GeoNode not being INSPIRE compliant by default is a downside. Also, installing GeoNode may not be a smooth process, and several issues can occur as reported by several users [61].

Nevertheless, the advantages outweigh the disadvantages, making GeoNode a very complete tool on which to base an SDI implementation.

[16] `https://www.sqlite.org/index.html` (last acessed Oct. 17, 2022)
[17] `https://docs.mapstore.geosolutionsgroup.com/` (last acessed Oct. 16, 2022)

2.2 Related Work

GIS applications in archaeology began to be used in the 1980s, initially for modeling surfaces of archaeological interest and managing archaeological resources, and have been gaining a more prominent role due to new applications such as spatial analysis, integrating remote sensing techniques [62]. Initially these tools used desktop applications, which limited the real-time visualization and input of field data. However, this link between archaeological fieldwork and real-time collaboration, both in-field and off-field, is made possible by web GIS applications. The increasing computational performance of mobile devices also enables spatial data, stored in remote repositories, to be easily available in the field [63].

This section presents archaeology-related work that includes web or mobile GIS, catalogs and SDIs,and remote sensing with GIS. The methodology used in the search of the related papers is presented next.

2.2.1 Methodology

A primary search for papers was conducted in the databases Scopus, Science Direct, ACM Digital Library, IEEE Xplore, and SpringerLink. This search was performed on January 15th, 2022, and articles published in journals, book chapters, and conference proceedings were selected. This selection was made based on the title, giving priority to papers in English and with a more recent publication date. Table 2.1 lists the search strings used, as well as the number of papers initially selected.

Table 2.1: Search strings and number of selected papers

Topic	**Search String**	**Selection**
Web and Mobile GIS	archaeology AND (((web OR mobile) AND (gis OR "geographic information system")))	24
Catalogs and Infrastructures	archaeology AND web* AND ("catalog service" OR infrastructure OR (metadata AND retrieval))	5
Remote Sensing	archaeology AND "remote sensing" AND (method* OR technique*) AND gis	14

A more detailed analysis of their abstract and content was conducted to these selected papers. This analysis allowed the exclusion of articles that were not considered appropriate due to having a more archaeological focus, and also some duplicate articles caused by an overlapping of the databases. From a total of 43 initial papers, 29 papers were excluded, remaining 14 papers. A secondary search for information was also carried out, through the references of the papers initially selected and other sources of information, in order to complement the search. Thus, 9 papers were added, making a total of 23 papers for analysis. Not all works are directly presented, since they may be used to complement the information presented or serve as examples. Figure 2.5 shows an overview of the paper selection process.

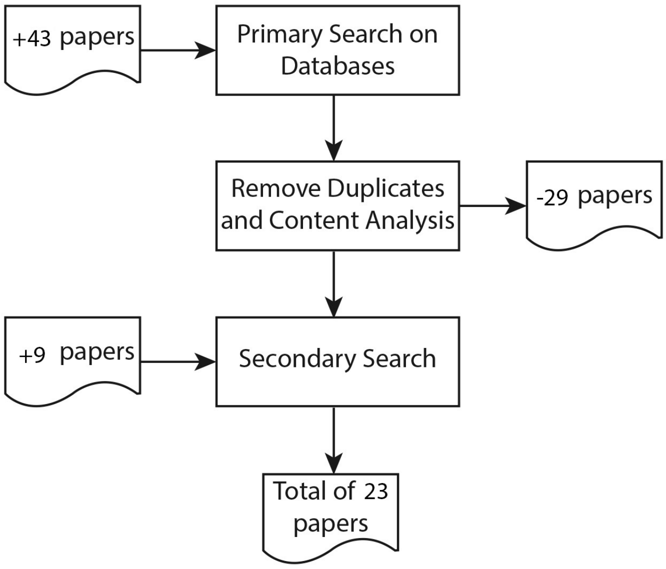

Figure 2.5: Overview of the paper selection process

2.2.2 Web GIS and SDI Applications

The concepts of Web GIS and SDI have evolved over the years, and there may be some inconsistency between the terminology used in the papers that are to be presented and the definitions presented in Section 2.1. Nevertheless, the terms used by the authors of the papers will be kept. There are several projects focused on web mapping applications for publishing, storing and visualizing archaeological data [64–68].

Gallo and Roberto [64] present Anteo, a web GIS platform for sharing archaeological data from Aquileia, Italy. The Anteo architecture is presented in Figure 2.6, and is an example of an application that aggregates the various components described in Section 2.1.2. It was built using only FOSS components, such as OpenLayers, GeoServer, GeoNetwork and PostgreSQL with PostGIS. Anteo has features such as uploading data in shapefile or GeoTIFF format, retrieving data through metadata or geographic bounding box search, and creating maps for viewing layers, where external base layers can also be added via WMS. Special emphasis is given to metadata, with the INSPIRE directive having been tested, and OGC standards, being compliant with standards such as WMS, WFS and CSW [64].

McCool [66] presents the Puuc Region Archaeological Geographic Information System (PRAGIS), a web mapping application that allows database querying and visualization of archaeological information, but also offers other functionalities such as drawing graphic elements on maps, distance and area measurements, and downloading user-selected geographic data. Along with archaeological information, the application also displays feature data, using existing WMSs, base maps, and allows user shapefiles to be added. However, unlike Anteo, this application uses Esri's proprietary software, namely ArcMap, ArcGIS Server, and ArcGIS Viewer for Flex, but was not presented as a permanent resource and, as such, is no longer publicly available [66].

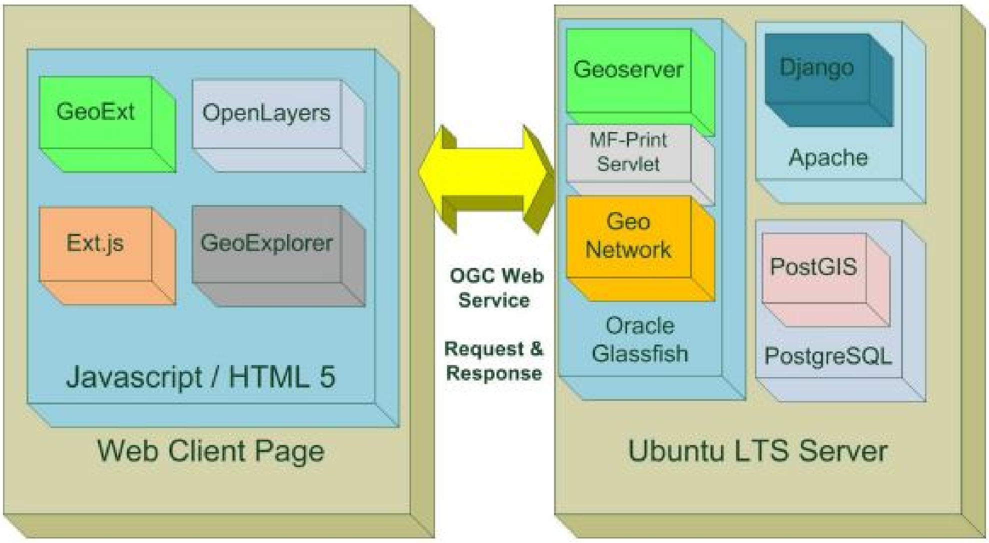

Figure 2.6: Anteo Architecture [64]

Another project is the Geographic Archaeological Information System of Rome (SITAR) [68], an open data infrastructure that, in addition to a web GIS, also has a digital archive. The web GIS enables the visualization of data on a map, and has features to customize the map layers, search and filter the contents, measure areas and distances, and export the map in vector, raster and textual formats. The platform allows spatial data to be downloaded and also provides a RESTful API and WMS and WFS services for sharing the data. However, there is no CSW-compliant service catalog. The archive allows the user to retrieve information stored in the database, such as photographs, reports, and other documents, through a full-text search tool. SITAR uses tools such as PostgreSQL with PostGIS, GeoServer and ElasticSearch[18] as search engine.

Buonanno *et al.* [60] present a paper that, although not related to archaeology, uses GeoNode as the basis for implementing an SDI. The goal of the project was to modify and extend GeoNode to upload and visualize DInSAR-related information, with all new functionalities having been successfully implemented. The authors conclude that GeoNode is an excellent framework for implementing an SDI that can be adapted to the project's requirements, considering one of its greatest strengths to be the use of OGC standards [60].

2.2.3 Mobile Solutions

The proliferation of mobile devices, such as smartphones or tablets, with internet connection and access to location systems has brought new opportunities in archaeology. Much of the work that has been published related to mobile devices in archaeology has focused on excavations and its documentation, giving less emphasis to mobile GIS solutions [69].

Styliaras [70] has reviewed platforms that support archaeological excavations on smartphones and has identified operations commonly used by archaeologists in the field. These include the ability to record data in various formats, such as text, drawings, or photographs,

[18] `https://www.elastic.co/pt/elasticsearch/` (last acessed Oct. 11, 2022)

collaboration and communication between co-workers, and the viewing of georeferenced content, for example on maps, while taking full advantage of the capabilities that mobile devices offer [70].

Mobile systems continue to be developed to address concrete challenges. Recently, Sobotkova et al. [71] presented a mobile system for documenting archaeological surveys that, on top of common requirements such as the annotation of data and metadata, and the sharing of this data through synchronization between devices in the field, present more specific requirements. One of these requirements is the ability to store, manage and synchronize data offline, as internet access in the field would be unavailable or unstable. This poses challenges for large amounts of data, as all the data is stored on the mobile devices without access to an online database. This system was developed using Field Acquired Information Management System (FAIMS) Mobile [72], an open-source platform that generates custom applications for data capture in field surveys of various areas [71].

Mobile GIS solutions are also being implemented, offering significant advantages when, for example, large amounts of data need to be recorded quickly [73]. Mobile GIS solutions like [74] and [69] were based on Esri's ArcGIS Collector, an application with a map-centric interface that allows collecting and updating information in the field, adding photos, and, with internet access, uploading the information to the geodatabase as well as updating the information on other mobile devices [69].

2.2.4 Catalogs and Data Infrastructures

Today, research infrastructures, digital archives and data services are important components of scientific research [75]. Regarding geospatial data, there are a wide variety of international [76, 77], national [78], and regional [79] geoportals that provide the data stored in SDIs. Open-source cataloging software such as GeoNetwork is widely used, as for example in [77, 79, 80]. To ensure high interoperability, for easy sharing and access to data, the SDIs are delivering data through OGC standards, such as WMS and WFS, and, for EU countries, metadata standards such as the INSPIRE directive are also considered.

Throughout the years, archaeologists have accumulated large amounts of fieldwork data, which has a special nature since its creation can result from the destruction of the evidence, making the availability of this data even more important in archaeological research [81]. Efforts are being made to make this data available centrally. At a national level, there are examples such as the well-known Archaeology Data Service (ADS), a digital repository in the United Kingdom that acts as an aggregator of archaeological data from larger and local agencies [81], and more recently work has been undertaken to develop an Italian geoportal focused on archaeological data [82]. On an international scale, the Advanced Research Infrastructure for Archaeological Dataset Networking in Europe (ARIADNE) project aims to create a global access point to archaeological resources by providing, along with the catalog, other services, such as data visualization [75].

Ronzino *et al.* [82], for example, mentions that one of the problems in the creation of the Italian national infrastructure is the lack of homogeneity among geoportals, with only some following the OGC standards and the INSPIRE directive. This highlights even more the importance of using the standards, especially when the data is to be made publicly available.

2.2.5 Remote Sensing with GIS

The use of remote sensing with GIS allows the identification of areas with high potential as archaeological sites, through predictive models [4, 83], and archaeological features, through techniques such as shape detection [84, 85].

Ioro *et al.* [84] used the shape detection technique for the automatic detection of archaeological sites combined with GIS. The results of the shape detection process are layers indicating the probable position of the shape to be searched, which are integrated into a GIS. Then, the user can combine the layers with other information to optimize the analysis. GIS tools can also be used for building predictive models. Nsanziyera *et al.* [83] used the tools provided by ArcGIS, Esri's proprietary tool, to create a model capable of locating areas with high archaeological potential.

Iorio *et al.* [85], also using the shape detection technique to detect coastal underwater archaeological sites, integrated the algorithms and results into a Web GIS, to facilitate user interaction with the layers and information produced. Furthermore, making the data available on the web helped the validation during the on-site campaign.

However, based on the papers that were gathered and analyzed, the integration of the algorithms with GIS or Web GIS is mainly done for visualization and interaction with the data. There is no focus on metadata management or sharing the data through standards such as CSW.

2.3 Concluding Remarks

This chapter started by introducing the main standards used in web GISs and SDIs, as well as the typical architectures on which their implementation is based. With this, it was possible to identify the main components of these systems and infrastructures, and also how they interact with each other. For each component the main FOSS tools were analyzed, of which several OSGeo projects stood out. The main differences of the options for each component were analyzed, to help understand which was the best choice. GeoNode was highlighted in this analysis, as it is a framework that integrates all the components presented in the typical architecture, using in each one of them the tools that stood out individually. Furthermore, GeoNode implements the OGC standards, making this tool an interesting choice.

After that, works related to the scope of this project were analyzed. The analysis of Web GISs and SDIs applied to archaeology allowed to understand the main functionalities found in this type of platforms. It also confirmed the use of the FOSS tools presented and highlighted the importance of the use of standards. Projects such as Anteo [64] provided an insight into what the solution developed in this project will be regarding the functionalities and tools to be used, apart from the remote detection component. The work of Buonanno *et al.* [60] confirmed the capabilities of GeoNode and the possibility of extending and modifying the tool, highlighting it as the best choice to be the basis of this project's platform.

The analysis of mobile solutions pointed out some aspects to be considered, namely the amount of data that has to be stored when there is no access to the online database. The analysis of infrastructures and catalogs that collect geospatial data allowed to emphasize even more the importance of standards to ensure the interoperability of data.

Finally, it was possible to conclude that the remote sensing algorithms are not integrated in SDIs capable of managing and sharing data and metadata. The integration with GIS is mostly intended for visualization, with no automated process to feed the algorithms with the information in the database.

CHAPTER 3

Platform Requirements and Architecture

This chapter describes the functional and non-functional requirements of the platform, as well as its main users, and also to detail the architecture of the solution and its components.

3.1 Platform Requirements

In this section the requirements of the platform are presented. The main functionalities of the platform are introduced, followed by a more detailed description of the actors and use cases. The platform requirements were defined based on the information that was gathered in the several meetings that took place with ERA Arqueologia, the main stakeholder in this project, and on the documentation provided by the company.

3.1.1 User Stories

The main features of the platform are presented from the user's point of view through the user stories presented next:

- As an archaeologist/ heritage technician, I want to aggregate the various sources of heritage and territorial information so that I can get an integrated and multi-layered view of the territory and the different archaeological and heritage sites.
- As an archaeologist/ heritage technician, I want to automatically process the available information so that I can identify archaeological sites or areas with potential archaeological elements.
- As an archaeologist/ heritage technician, I want to add new archaeological sites and/or modify the information on a site record so that I can keep the information up to date.
- As an archaeologist/ heritage technician, I want to integrate into the platform the information obtained in the field in real time, through a mobile version, so that I can have uniform data, which is crucial for collaborative work.

- As an archaeologist/ heritage technician, I want to use the mobile version without Internet connection so that I can carry out field work in remote locations where there is no mobile network connection.
- As an archaeologist/ heritage technician, I want to keep the history of the information uploaded to the platform so that I can restore previous versions.

3.1.2 Assumptions

There are some assumptions to be considered for a proper operation of the platform. It is assumed that the platform receives the data already pre-processed and filtered, so the raw LiDAR point clouds will not be included, but the derived products such as DTMs. Also, the geographic information that is imported into the platform, either shapefiles or GeoTIFFs, uses the official reference coordinate system for Portugal: ETRS89 / Portugal TM06 - EPSG:3763.

During the requirements definition it was necessary to reach an agreement with the other Odyssey project members about the terminology to be used. A definition was reached to describe archaeological sites and archaeological occurrences. An archaeological site is considered to be the area of archaeological intervention, or area of interest, delimited by a surrounding polygon. This site is volatile, meaning that it can evolve over time, increasing or decreasing its area. As for the occurrences, they are the various points of archaeological interest in a given site, where there may be several occurrences for each site. Each occurrence is delimited by a bounding polygon. The definition of these terms was important, as there is no broad consensus within the archaeological community, and the absence of a definition could lead to misunderstandings and ambiguities.

3.1.3 Actors

The users will be mainly archaeologists and heritage technicians, who will use the platform in a very similar way. It is expected that the users have expertise in the use of computers, and especially in the use of GIS tools such as QGIS as the platform operation mode may adopt common practices in this type of systems. Nevertheless, the platform should be user-friendly and intuitive so that any user can benefit from its functionalities. Moreover, the users of the platform will not be exclusively Portuguese, so the platform should be available in Portuguese and English.

The main actors are:

- **User**: Represents the archaeologists and heritage technicians as authenticated users on the platform. This actor has access to most of the functionalities of the platform, which includes adding new data and archaeological sites and managing them, along with the corresponding site record.
- **Administrator**: Represents the user responsible for the administration of the platform and its users. This actor has access to all the functionalities that the normal user has, and is also able to register and delete users from the platform.

3.1.4 Use Cases

Figure 3.1 presents the use case diagram of the platform. Along with more generic use cases, the diagram has two packages that aggregate the use cases related to the archaeological sites and the site records. All use cases assume that the actors they interact with are authenticated by the system and have an active session. In general, all use cases will be shared by the web version and the mobile version, but there are some exceptions. Uploading new data and managing the users of the platform will be exclusive to the web version, while downloading data to be able to work offline will be exclusive to the mobile version.

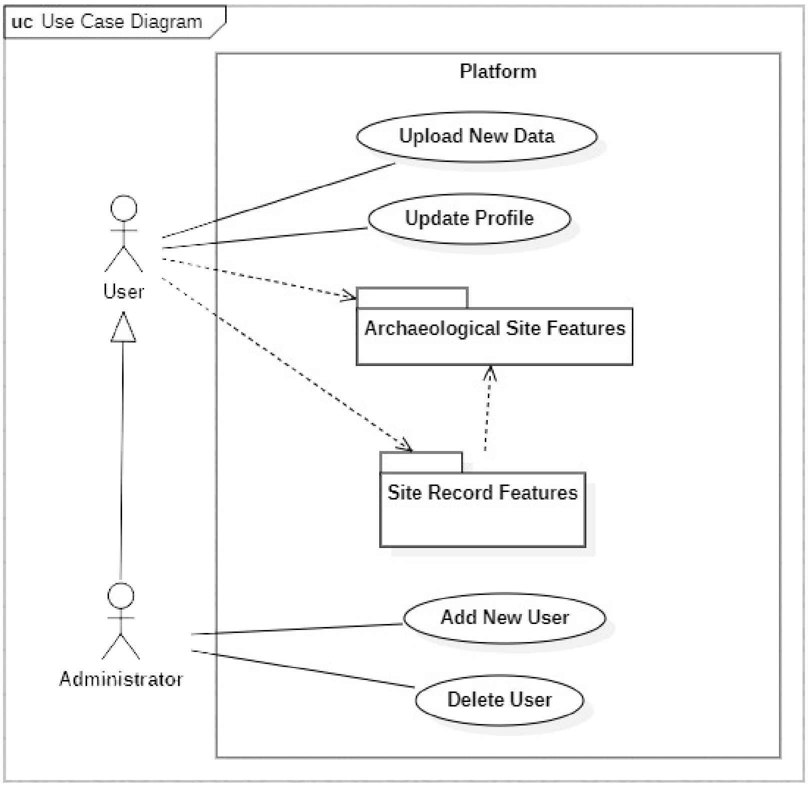

Figure 3.1: Use Case Diagram

Following are the details of the use cases and packages. The most generic use cases are presented next:

- **Upload New Data**: Allows the user to upload new data to the platform. These data come from different sources, namely multispectral images, thermal images, LiDAR-derived products, and different types of maps. The GeoTIFF format shall be supported for raster data, while the shapefile format shall be supported for vector data. The new data is then added to the database.
  Priority: High
- **Update Profile**: Allows the user to change his personal data, such as name and password.
  Priority: High

- **Add New User**: Allows the administrator to register new users on the platform. The basic information of the user, such as name, e-mail and temporary password, are inserted.
  Priority: High
- **Delete User**: Allows the administrator to delete a registered user from the platform.
  Priority: High

Regarding the archaeological sites feature package, Figure 3.2 presents the use case diagram containing the operations related to the archaeological sites.

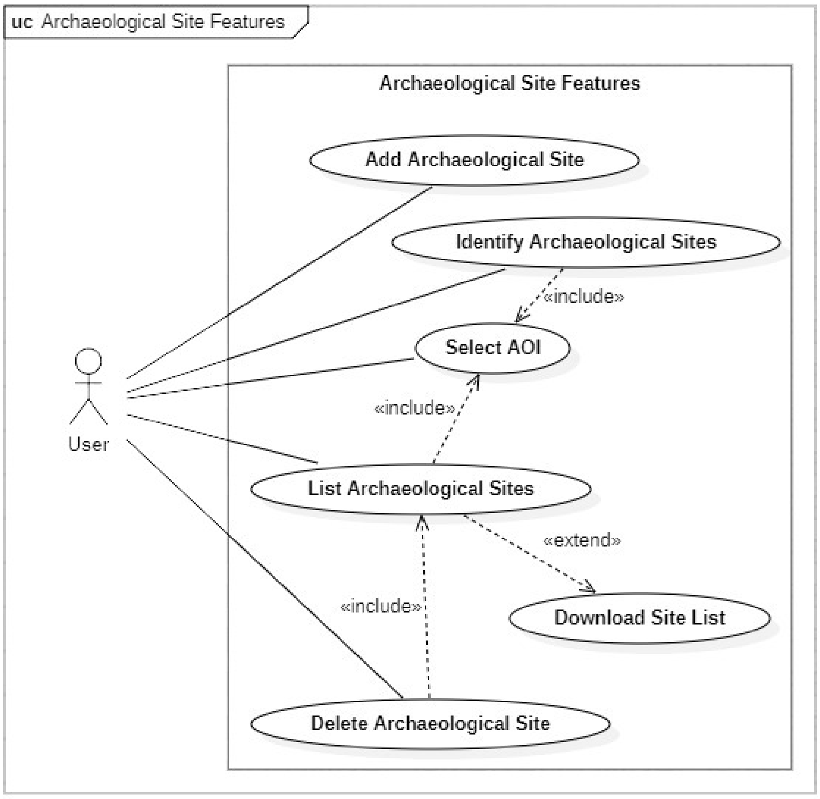

Figure 3.2: Archaeological Site Features

The details of the use cases of this package are presented next:

- **Add Archaeological Site**: Allows the user to add a new archaeological site, including its details and occurrences, to the database. This operation is useful to insert previously known information, and to add sites that have not been automatically created upon the execution of the algorithms. The user must enter information about the site and occurrence, such as name, location, attributes, among others.
  Priority: High
- **Identify Archaeological Sites**: Allows the user to execute the ML algorithms to automatically identify sites with a high potential to contain archaeological occurrences within the Area of Interest (AOI) selected by the user. For each occurrence its bounding polygon is defined and its type assigned, with those occurrences being stored in an

archaeological site created for this purpose. All data are then stored in the database.
Priority: Medium. This use case is dependent on the work of other teams in the Odyssey project, as the goal of this project is solely the integration of the algorithms into the platform.

- **Select AOI**: Allows the user to select the area of the territory they are working on, by selecting a bounding box on a map or by performing a text search that can be fine-tuned with filters.
  Priority: High
- **List Archaeological Sites**: Allows the user to obtain a list of the archaeological sites in the selected AOI. The list contains both the previously known sites and the potential sites identified automatically by the algorithms. The following operations can also be performed.
  Priority: High
  - **Download Site List**: Allows the user to download the list of potential and previously known archaeological sites, and their respective site records, to a mobile device only. The information is stored locally so that it can be possible to work in places without access to mobile network and internet.
    Priority: Low
- **Delete Archaeological Site**: Allows the user to delete a specific archaeological site, and its corresponding site record, stored in the database.
  Priority: High

Figure 3.3 presents the use case diagram that contains the operations performed in the site record feature package.

The use cases of this package are presented next:

- **View Site Record**: Allows the user to view the information of a given site record, which includes information about the archaeological site, all its occurrences and related information, and the related files. On each site record, the user can perform the following operations.
  Priority: High
  - **Add Annotation**: Allows the user to add new information to a given site record. The user can add new information about the archaeological site and occurrences, add a new occurrence to the site, and also associate files such as photos and videos. To add a set of previously known occurrences, it must be possible for the user to import this information using files in a predefined format.
    Priority: High
  - **Update Annotation**: Allows the user to update the existing information in a given site record. The information can be corrected or confirmed, which is particularly useful for occurrences automatically detected by the algorithms.
    Priority: High
  - **Delete Annotation**: Allows the user to delete information from a given site record, which includes occurrences and files associated with the site.
    Priority: High

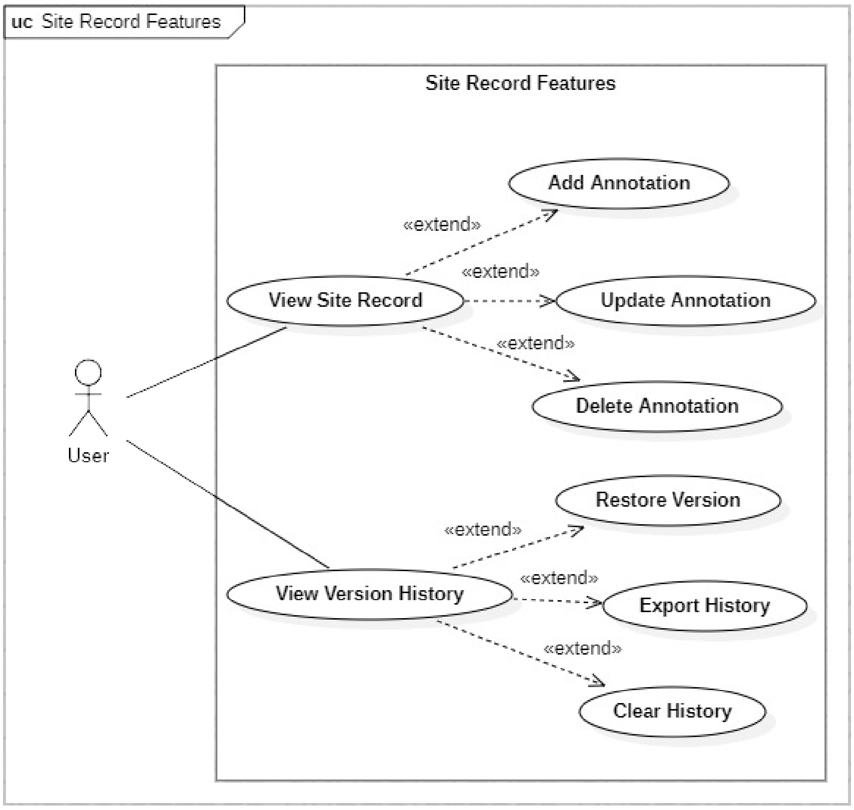

Figure 3.3: Site Record Features

- **View Version History**: Allows the user to view the version history of the site records. All information inserted by users has an identifier and timestamp so that the previous versions can be viewed chronologically. The user can also perform the following operations.
  Priority: Low
  - **Restore Version**: Allows the user to restore a previous version of the site record, based on its version history.
    Priority: Low
  - **Export History**: Allows the user to export historical information from the site records, being able to select which versions to export.
    Priority: Low
  - **Clear History**: Allows the user to delete previous versions of the record, being able to select the ones to be deleted.
    Priority: Low

3.1.5 Non-functional Requirements

The non-functional requirements of the platform are the following:

- **Security**: As the data in the platform will not be public, only authorized users can access the platform. For this purpose, an authentication module should be developed

that only allows access to users registered on the platform. The management of registered users will be done exclusively by the administrator of the platform.
Priority: High

- **Usability**: Although users are expected to have expertise on how to use this type of system, the interface should be simple to use and intuitive, and the functionalities should be self-explanatory. For this, testing and validation of the platform's interface with ERA must be carried out.
Priority: High
- **Reliability**: Since the data will be in a permanent synchronization with the database, it is essential that no information is lost during the process. In addition, users may use the platform in locations without internet access through the mobile version, which means that data can often be synchronized only at a later time, so precautions should be taken to avoid overwriting information.
Priority: High
- **Configurability**: The platform users will not be exclusively Portuguese speakers. For this reason, the platform, both the web and mobile versions, should allow the user to configure the language. The available languages should be Portuguese and English.
Priority: Medium
- **Portability**: The platform should be accessible through the different browsers available and should behave the same way in all of them.
Priority: Medium

3.2 Platform Architecture

This section presents the architecture of the platform, specifying its domain model, and describing its components and how they interact with each other.

3.2.1 Domain Model

The domain model, presented in Figure 3.4, is a conceptual model that describes the main entities of the problem, along with their main attributes, and how these entities relate to each other. The actors, described in Section 3.1.3, are represented by the User entity, which specializes in the Administrator entity. These entities are responsible for creating and managing the archaeological sites and occurrences, which includes their attributes, metrics and files, and for executing the automatic detection algorithms.

Archaeological sites, represented by the Site entity, must have at least one associated occurrence, and can be characterized by several attributes belonging to defined categories. The sites contain a set of information, including the status, that can be either "Not Verified" or "Verified", and the representation through point and polygon.

Similarly, archaeological occurrences, represented by the Occurrence entity, also contain a set of information that includes the status and representation through point and polygon. For the status of occurrences, in addition to the states defined for sites, it can also assume the value of "Verified - False Positive" and "Verified - True Positive". These two extra status

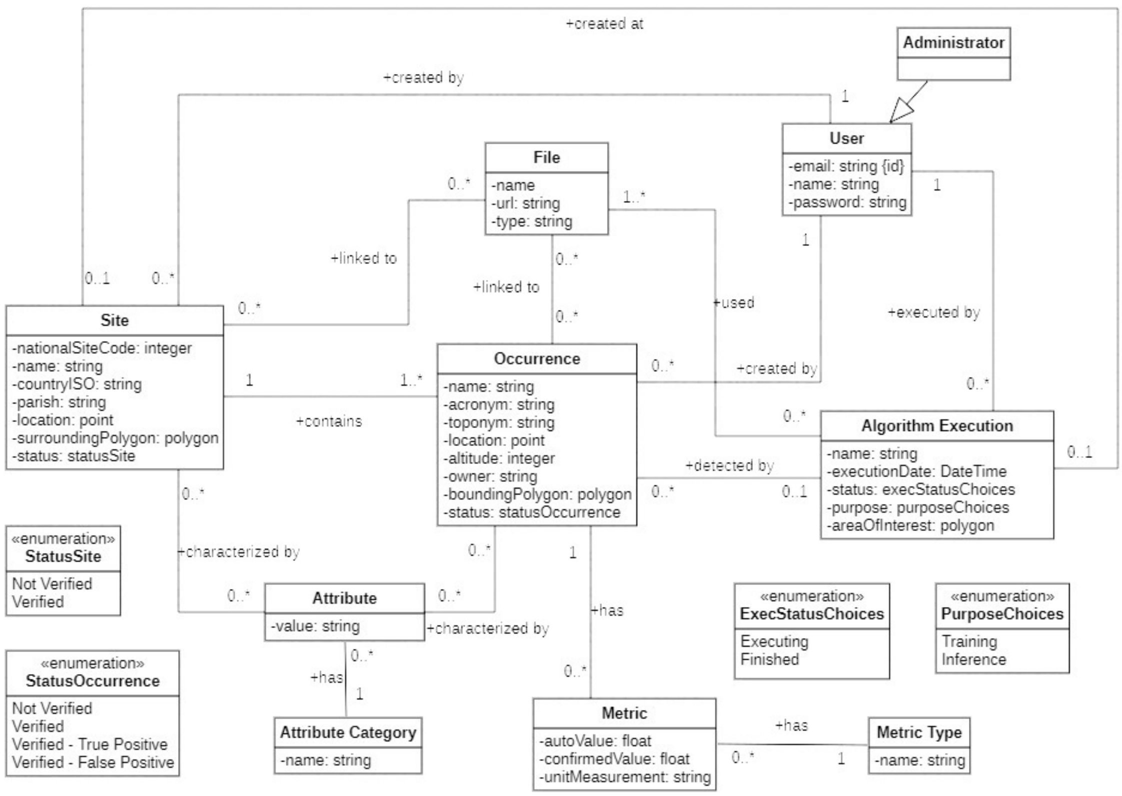

Figure 3.4: Domain Model

are intended for occurrences that are automatically detected by the ML algorithms. The occurrences are also characterized by attributes, and may contain metrics of various types, such as area, diameter, among others.

The files, represented by the File entity, represent the raster and vector spatial data that are uploaded to the platform, and also text files, photos, images, and others. These files can be associated to the sites and occurrences, being the raster data also used in the execution of the ML algorithms. The algorithms' executions are stored so that it is possible to identify the occurrences detected in each execution and the files that were used. The executions can be for training purposes or to infer new detections, and their execution state can be "Executing" or "Finished".

To better understand the geographical relationship between the entities, a model was drawn using Object Modeling Technique for Geographic Applications (OMT-G) (see Figure 3.5), through the OMT-G Designer[1] [86]. The OMT-G model is based on the Unified Modeling Language (UML) class diagram, adding georeferenced classes in addition to the conventional classes, to facilitate the representation of geographic applications. Georeferenced classes describe objects that have a spatial representation and are associated with the Earth's surface, and can be either individualizable geographic objects, such as rivers, or objects that are continuously distributed in space, such as geology. These classes are represented in the model

[1] `http://aqui.io/omtg/` (last acessed Oct. 23, 2022)

in a similar way to a common UML class, adding in the upper left corner the indication of the geometric shape of the representation. It should be noted that, in this model, relationships between classes through simple associations, as in UML, are represented by solid lines, while spatial relationships are represented by dashed lines, making it easy to distinguish visually between the two types of relationships [87].

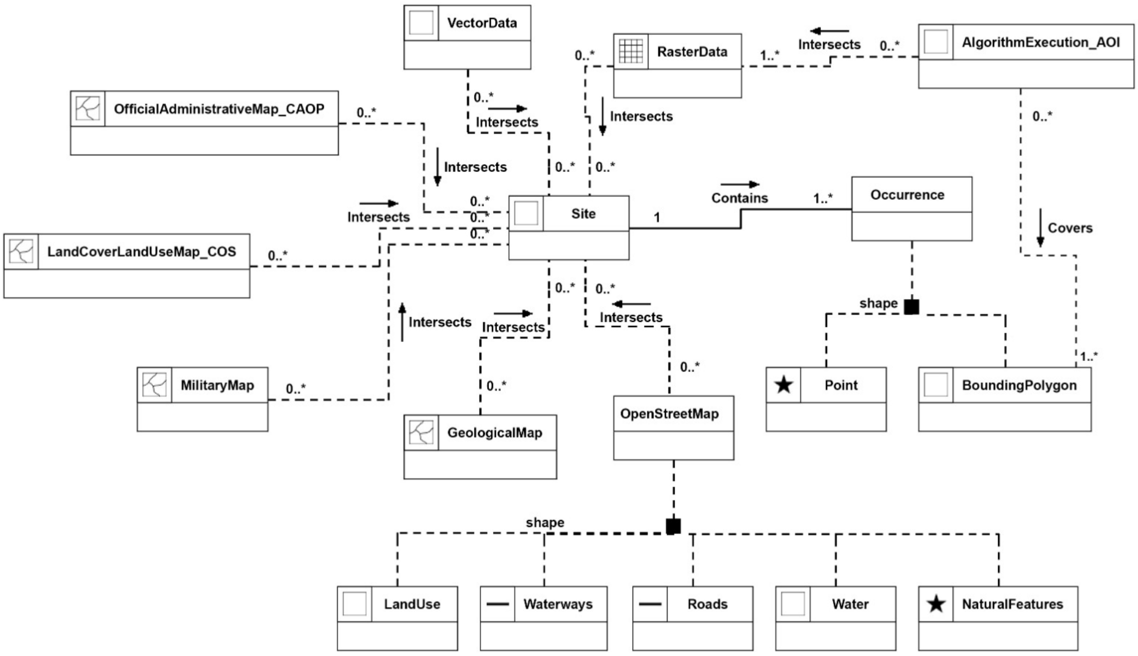

Figure 3.5: OMT-G Model

The sites and occurrences can be intersected with the vector and raster data that is uploaded to the platform. Additionally, ERA requested base layers that also intersect the sites and occurrences and from which additional information can be obtained. These base layers include the Official Administrative Map of Portugal (*Carta Administrativa Oficial de Portugal*) (CAOP), which contains the administrative boundaries of the parishes and municipalities of Portugal, the Land Cover Land Use Map (*Carta de Ocupação de Solo*) (COS), which contains information about the use and occupation of the land, geological maps, military maps and also the information available on OpenStreetMap, being relevant the use of land, water, watercourses, roads and natural features. Both base layers and data are not restricted to a single site or occurrence, and may cover more than one simultaneously.

The execution of the algorithms is represented by a polygon depicting the AOI selected by the user. For the algorithms to be executed, it is first necessary to identify the occurrences whose polygon is covered by the AOI, and the raster data that intersects the AOI, being the existence of both elements required.

It should be noted that, to improve the clarity and understandability of the model, the site is only represented by a polygon although it can also be represented by a point. The same goes for vector data which, in addition to polygons, can also be lines and points. Moreover, all intersections that can be performed with the site can also be performed with the occurrences.

3.2.2 Physical Architecture

Figure 3.6 shows the deployment diagram, where the physical components are represented as well as how they interact with each other. As mentioned in Chapter 2, GeoNode was identified as the best framework on which to base the implementation of the platform, as it already has the components pointed out in the typical architecture. Therefore, the architecture of the platform follows the GeoNode architecture, presented earlier in Figure 2.4. The server-side components should be on the server dedicated to the Odyssey project, allowing the user to interact with the platform through a web browser via HTTP.

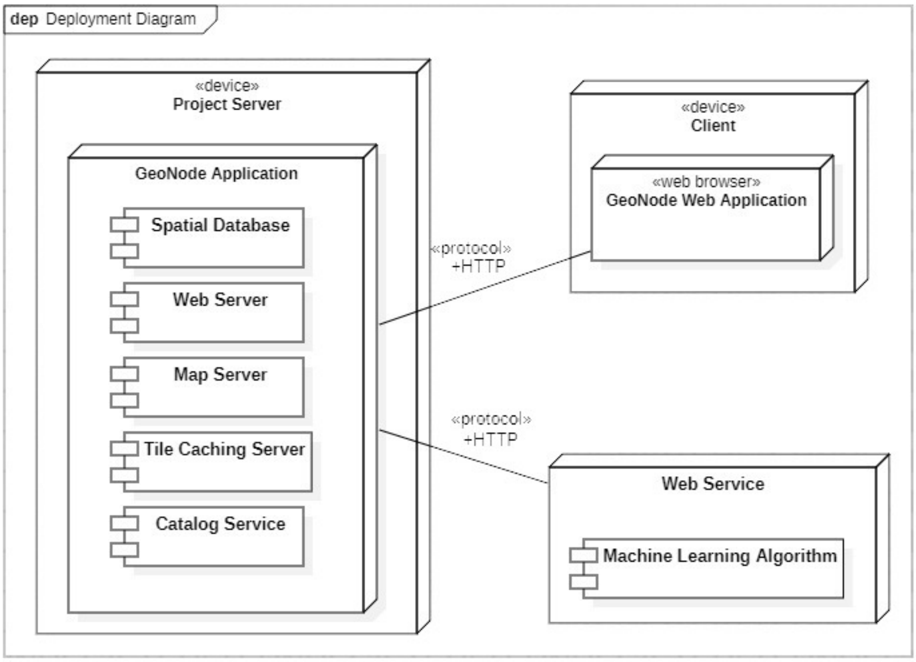

Figure 3.6: Deployment Diagram

For the integration of the platform with the ML algorithms, the Odyssey project member responsible for the algorithms must create a web service, which may or may not be on the same server as the platform, that receives and processes the information, feeds the algorithms, and returns the results obtained, if that is the case. The communication between the platform and the web service is done via HTTP POST requests.

CHAPTER 4

Implementation

This chapter presents the implementation details of the platform. The platform source code can be found in the GitHub repository[1], as well as the installation and configuration instructions.

4.1 GeoNode

GeoNode was chosen as the base platform for the implementation, as it already has the required components and several features. In this project GeoNode version 3.3.x was used, as it is the stable version.

In this section, an overview of the main features of GeoNode is presented, which also matches some of the requirements of the project, and is given an explanation of how the customization of GeoNode is done.

4.1.1 Overview

As mentioned Section 2.1.3, GeoNode is a GeoCMS system that allows the publishing and management of geospatial data in an easy-to-use interface, ideal for non-specialized users.

There are three primary types of resources in GeoNode:

- **Layers**: represent a raster or vector spatial data source. GeoNode supports various data formats, such as shapefiles for vector data and GeoTIFFs for raster data, allowing data to be uploaded using the original SRS. The uploaded layers are published as WMS and WFS, with GeoNode allowing to download the data and metadata. Besides local layers, GeoNode also supports importing remote layers through WMS.
- **Maps**: represent a set of different layers that can be viewed simultaneously on a web map. All layers are automatically transformed to the WGS 84 / Pseudo-Mercator - EPSG:3857[2] reference system, making it possible to view them together with base maps like OpenStreetMap. The maps also have a number of tools, including the ability to sort

[1] `https://github.com/rafaelsa99/odyssey` (last acessed Oct. 27, 2022)
[2] `https://epsg.io/3857` (last acessed Oct. 14, 2022)

and change the transparency of layers, print the map, perform textual and geometric annotations, and search the map by location or name.

- **Documents**: GeoNode allows the upload and publishing of documents in a variety of formats, namely *.txt, .log, .doc, .docx, .ods, .odt, .sld, .qml, .xls, .xlsx, .xml, .bm, .bmp, .dwg, .dxf, .fif, .gif, .jpg, . jpe, .jpeg, .png, .tif, .tiff, .pbm, .odp, .ppt, .pptx, .pdf, .tar, .tgz, .rar, .gz, .7z, .zip, .aif, .aifc, .aiff, . au, .mp3, .mpga, .wav, .afl, .avi, .avs, .fli, .mp2, .mp4, .mpg, .ogg, .webm, .3gp, .flv, .vdo, .glb, .pcd* and *.gltf*. The documents can then be linked to layers and maps.

GeoNode gives great emphasis to metadata, allowing the management of metadata for layers, maps and documents. It allows to upload an Extensible Markup Language (XML) document to automatically populate the metadata, or manually fill in the metadata elements, letting the user to add extra fields dynamically. GeoNode also has search pages for each of these resource, allowing to fine tune the search by text, categories, keywords, region, type, owners and date. It also provides a spatial filter, where it is possible to search by geographical extent.

GeoNode has a built-in authentication and authorization system. GeoNode's authentication is based on Django authentication, managing the users, groups, roles and sessions, while GeoNode's authorization system is based on the Open Authorization (OAuth) 2.0 protocol. Thus, users have the tools to manage their profiles, and assign permissions to each resource to define which users or groups can view, edit or delete the data and metadata.

Other features of GeoNode, such as the creation of dashboards and comment and rating systems, will not be detailed as they are not within the scope of the requirements of this project.

4.1.2 Customization

GeoNode is designed to be extended and customized to meet the requirements of the project. However, the customization of GeoNode should not be done directly in the GeoNode source code, or GeoNode Core, as this introduces problems when upgrading to a new version. To avoid this, the recommended approach is to use geonode-project[3].

Geonode-project is a template for creating custom GeoNode projects, and contains a minimal set of files following the structure of a Django project. GeoNode itself is installed as a requirement of the project, and so it is possible to extend, modify or replace GeoNode components, and even create new Django applications, without touching the original GeoNode code.

Thus, a new Django application was created to implement the project requirements that GeoNode does not have implemented. The customization of GeoNode's layout was done by overriding the existing templates.

[3] `https://github.com/GeoNode/geonode-project` (last acessed Oct. 22, 2022)

4.2 Data Models

The implementation of the data models was done using the Django Object-Relational Mapper (ORM). By using the ORM, there are advantages such as easier code development and maintenance, as it already has built-in tools for migrations, data types, relationships, queries and data validation, while supporting multiple databases without the need to rewrite code. Moreover, it is integrated with other Django features, such as Django Forms, and by avoiding the use of raw Structured Query Language (SQL), the probability of SQL injection vulnerabilities is reduced.

During the creation of the models, an effort was made to use the data models already existing in GeoNode. Figure 4.1 shows a simplified version of GeoNode's entity-relationship diagram, presenting some of its entities and their main attributes. For the *User* model, it was not necessary to create a new model, as GeoNode already has a similar model that is used in the authentication module. Thus, all relationships with the *User* model are made with the GeoNode user model. As for the *File* model, GeoNode's *Document* model was used, since it also has the purpose of storing files such as reports, photos, videos and sounds. This way, uploading and managing the files associated with the sites and occurrence uses the functionality that is available for documents in GeoNode. Similarly, the layers are stored using the existing model for GeoNode layers. This has the advantage of having the layers available through WMS and WFS, having GeoNode handling the metadata, and being integrated with the visualization functionalities through Maps. For the *Algorithm Execution* model, the relationship with the layers that were used is done with the *Layer File* model, which is the model that GeoNode uses to store the information of the layer files that were uploaded.

For the *Site* and *Occurrence* models, and considering their geographic dimension, an attempt to use the *Layers* model was made, since this could facilitate the integration of these data with the features that GeoNode provides. For this, the use of multi-table inheritance[4] was considered. In this type of inheritance there are no abstract models, i. e., each model in the hierarchy is a model by itself, and a one-to-one relationship is automatically created between the models. This type of inheritance would be useful for adding the specific attributes for sites and occurrences to the existing model for layers. However, trying to implement this approach revealed that a detailed analysis of the GeoNode code would be necessary to understand how the implementation is done. Considering that the learning curve would be steep and that time is an important factor in the project, this strategy was not chosen. Furthermore, this approach would create a layer for each site and occurrence, which would not be ideal since the layers should aggregate a set of sites and a set of occurrences.

Alternatively, the models for the sites and occurrences were created from scratch, and their publication as layers was left to GeoServer. This procedure will be further detailed in Section 4.3. The spatial fields of these models are stored using the WGS84 - EPSG:4326 reference system, to be consistent with the SRS that GeoNode uses to store its spatial fields. Since these models need to have a geographic representation, and can have two vector

[4] `https://docs.djangoproject.com/en/4.1/topics/db/models/#multi-table-inheritance` (last acessed Oct. 10, 2022)

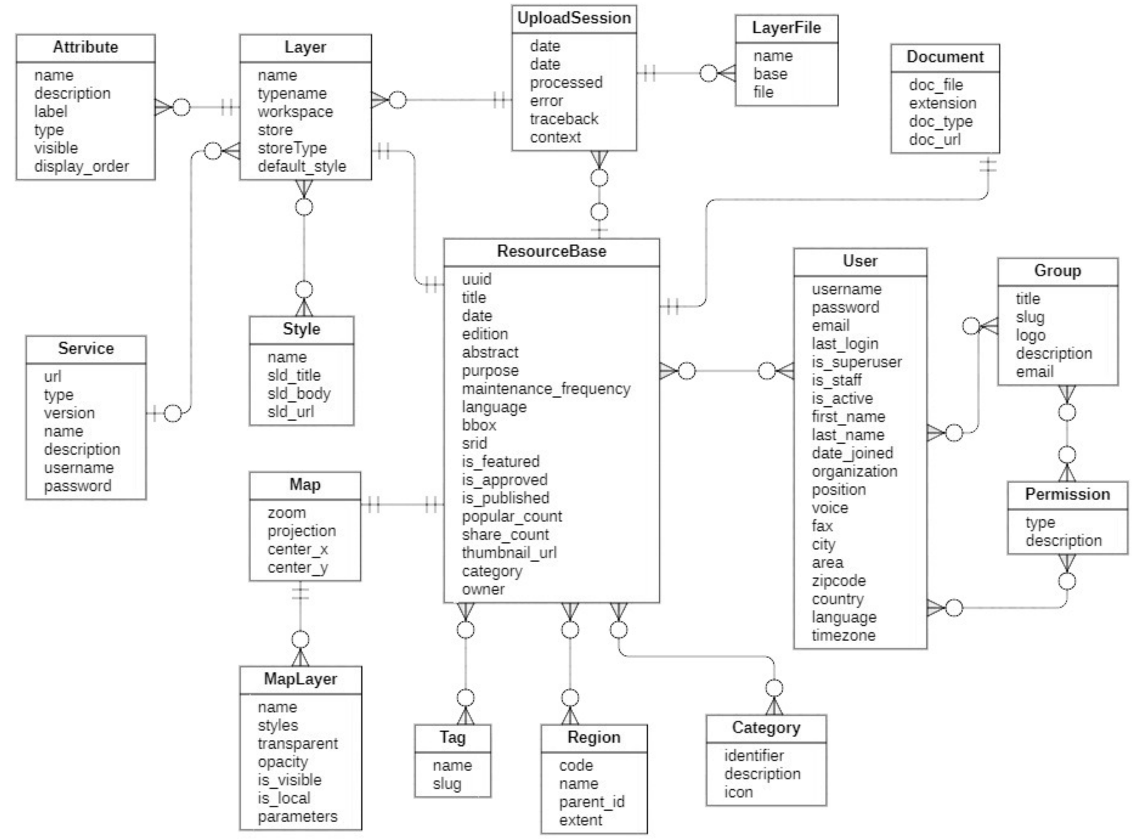

Figure 4.1: Simplified Entity-Relationship Diagram of GeoNode [58]

representations (point or polygon), it is necessary to verify that at least one of these fields is provided. To do this, a model-level verification was created through the *clean* method, a Django method where the validation of the model data is done, raising a *ValidationError* if both fields are null. Additionally, a database-level constraint was created that does the same verification, in case data is entered directly into the database without going through data validation beforehand. These verifications were created for both the *Site* and *Ocurrence* models. The models for the attributes and metrics associated with the sites and occurrences were also created from scratch. GeoNode's *Attribute* model does not have a separation by categories, which is necessary when defining archaeological attributes, and for this reason was not used.

When defining the models, the data types that Django offers were used, except for the site's country field. In this case, *django-countries*[5] was used, a Django application that has a *CountryField* for the models with all International Organization for Standardization (ISO) 3166-1 [88] countries as choices. This field stores the alpha-2 code, which is a two-letter code representing the name of the country. For the fields with predefined options, these were set as the choices for the respective field.

[5] `https://pypi.org/project/django-countries/` (last acessed Oct. 10, 2022)

4.3 GeoServer Configuration

After the creation of the data models for the sites and occurrences, it was necessary to find a way to publish these models as a GeoNode layer, so that it would be possible to benefit from the functionalities that are already available in GeoNode, such as the metadata handling for the layers and the layer viewing functions on the maps. To do so, the solution reached was to first publish the sites and occurrences as layers in GeoServer, and then use the *updatelayers* command that GeoNode provides to synchronize its layers with the GeoServer layers.

For this purpose, it is necessary to first set up a Data Store to fetch the information from the PostGIS database. After the database connection is established, the layers to be published need to be created. However, publishing the Site and Occurrence tables directly as layers is not possible, as each object can be represented by both point and polygon, and GeoServer assumes only one geometric representation. GeoServer addresses this issue and provides several alternatives for publishing the data correctly[6]. The solution used is the creation of views, as it is a less invasive solution that does not require data restructuring and is versatile.

The use of views also allows the selection of data with specific properties and that are not restricted to a single data table. Although, for example, attributes are stored in different tables and related to sites and occurrences through foreign keys, it is relevant that they are also provided in the information for each site or occurrence. Also, as sites and occurrences can have various status, it is useful to publish the information separately for each status. This way, the user can choose to view, for example, only the sites or occurrences that have already been verified in the field, without seeing at the same time those that have not yet been verified. Thus, a total of eight views were created for the occurrences and four views for the sites. Each view presents information from a specific status and with only one of the geometry types, so the total of twelve views created consist of the possible combinations between the available status and geometry types.

The Code Snippet 1 presents the SQL view for the occurrences with the status "Verified" and represented by a polygon. A further analysis of this view shows that in addition to the fields in the occurrences table, information regarding metrics and attributes is also obtained. The view is particularly complex, since the relationship to the attributes is many-to-many and to the metrics is one-to-many, making it necessary to aggregate the whole collection into a single string.

The views are configured in GeoServer by creating a new Data Layer for each view, using the Data Store created earlier. For spatial queries to run properly, the correct SRS must be indicated in the spatial fields. In this case, it is the WGS84 - EPSG:4326 reference system. Once all layers have been created, it is necessary to update the GeoNode layers to include the newly created layers in GeoServer using the *updatelayers* command. This command can be executed with filters so that only specific layers are updated. This is useful as the layers are updated in GeoNode whenever there are changes to the site or occurrences, and it may

[6] `https://docs.geoserver.org/stable/en/user/styling/sld/tipstricks/mixed-geometries.html` (last acessed Oct. 12, 2022)

```sql
SELECT occurrence.id,
       occurrence.designation,
       occurrence.acronym,
       occurrence.toponym,
       site.name AS site_name,
       occurrence.owner,
       occurrence.altitude,
       occurrence.bounding_polygon,
       people_profile.username AS added_by,
       string_agg(metric_values.automatic_values, ', ') AS automatic_metrics,
       string_agg(metric_values.confirmed_values, ', ') AS confirmed_metrics,
       attributes.values AS attributes,
       'Verified' AS status
FROM occurrence
LEFT JOIN
  (SELECT occurrence_attribute_occurrence.occurrence_id,
          array_to_string(array_agg(attribute_category.name || ': ' ||
                          attribute_choice.value), ', ') AS values
   FROM occurrence_attribute_occurrence,
        attribute_category,
        attribute_choice
   WHERE occurrence_attribute_occurrence.attributechoice_id = attribute_choice.id
     AND attribute_choice.category_id = attribute_category.id
   GROUP BY occurrence_attribute_occurrence.occurrence_id)
     AS attributes ON attributes.occurrence_id = occurrence.id
LEFT JOIN
  (SELECT metric.occurrence_id,
          array_to_string(array_agg(metric_type.name || '=' || metric.auto_value ||
                          metric.unit_measurement), ', ') AS automatic_values,
          array_to_string(array_agg(metric_type.name || '=' ||
                          metric.confirmed_value || metric.unit_measurement), ', ')
            AS confirmed_values
   FROM metric,
        metric_type
   WHERE metric.type_id = metric_type.id
   GROUP BY metric.occurrence_id) AS metric_values
      ON metric_values.occurrence_id = occurrence.id,
        people_profile,
        site
WHERE occurrence.added_by_id = people_profile.id
  AND site.id = occurrence.site_id
  AND occurrence.status_occurrence = 'V'
GROUP BY occurrence.id,
         attributes.values,
         people_profile.username,
         site.name;
```

Code 1: SQL View for the Verified Occurrences represented by a Polygon be necessary to update only the layers corresponding to sites, for example. This is done in a thread, as the process is time consuming.

4.4 CRUD Operations

One of the benefits of Django is its administration interface that is created automatically. All the data models that have been created were added to the administration site, which automatically generates the Create, Read, Update and Delete (CRUD) operations based on the fields in the models. Obviously, the use of this administration interface should not be for for general users, but only for use by the administrator of the platform. Therefore, CRUD operations should be created in the front end for the regular users. These features for the sites, occurrences and executions of the algorithms will be presented later in the sections 4.5 and 4.6 respectively. For the attributes and metric types, the CRUD operations are only available in the admin site, as this data will not be changed regularly and should be managed only by the platform administrator. Regarding documents, layers and users, GeoNode already has all the features created, so it was not necessary to create new ones.

The forms for the CRUD operations were implemented based on the data models, as the form fields are strictly bound to the model fields. Django allows the creation of forms through the models using the *ModelForm* class, or *ModelAdmin* class for the Django admin site, avoiding having to duplicate the definition of these fields for the forms. The form fields are rendered in the templates using the default input types. However, for the geographic fields and one-to-many and many-to-many relationships, other widgets were used for better usability. For the geographic fields, to set the point and polygon, a Leaflet widget for Django, was used, through *django-leaflet*[7]. For the one-to-many and many-to-many relationships, widgets from the Select2 library for django were used, through *django-select2*[8] application. The *Select2MultipleWidget* renders the options on the page, with the advantage of including a search functionality which is useful when there are a large number of options. To render the forms in a more appealing way, the *django-crispy-forms*[9] application was used, which allows to control the form rendering without having to write custom form templates.

4.5 Site and Occurrence Features

As mentioned in Section 4.4, it was necessary to create the CRUD operations on the front end for the sites and occurrences. In addition to the fields defined in the models, two more fields have been added to the site and occurrence forms, namely fields to manually enter the latitude and longitude. This is useful for inserting the exact location without having to select the point on the map. On submitting the form, these two fields are converted to a point and its vector representation is saved.

When creating a new archaeological site, it is required to also create a new occurrence, and so the site creation page displays the forms of both models. This constraint is also verified in

[7] `https://django-leaflet.readthedocs.io/` (last acessed Oct. 11, 2022)
[8] `https://django-select2.readthedocs.io/` (last acessed Oct. 11, 2022)
[9] `https://django-crispy-forms.readthedocs.io/` (last acessed Oct. 11, 2022)

the Django admin site, through the *clean* method of a custom formset. For simplicity's sake, when creating a site it is only possible to create one occurrence, without inserting metrics. By default, the sites and occurrences created manually have the "Verified" status. The remaining states are used for sites and occurrences created with the execution of the ML algorithms, as presented further in Section 4.6.

After the creation, it is possible to add more occurrences on the site edit page, and metrics on the occurrences edit page. For better usability, the metrics forms are all displayed inline on the occurrence edit page, with the possibility to add new metrics. To present the forms inline, the *django-crispy-forms* template *table inline formset* was used, by defining a *FormHelper*, as can be seen in Figure 4.2. When deleting a site, the user is notified if there are still occurrences associated with it. If there are, and the user proceeds with the deletion, all associated occurrences are also deleted.

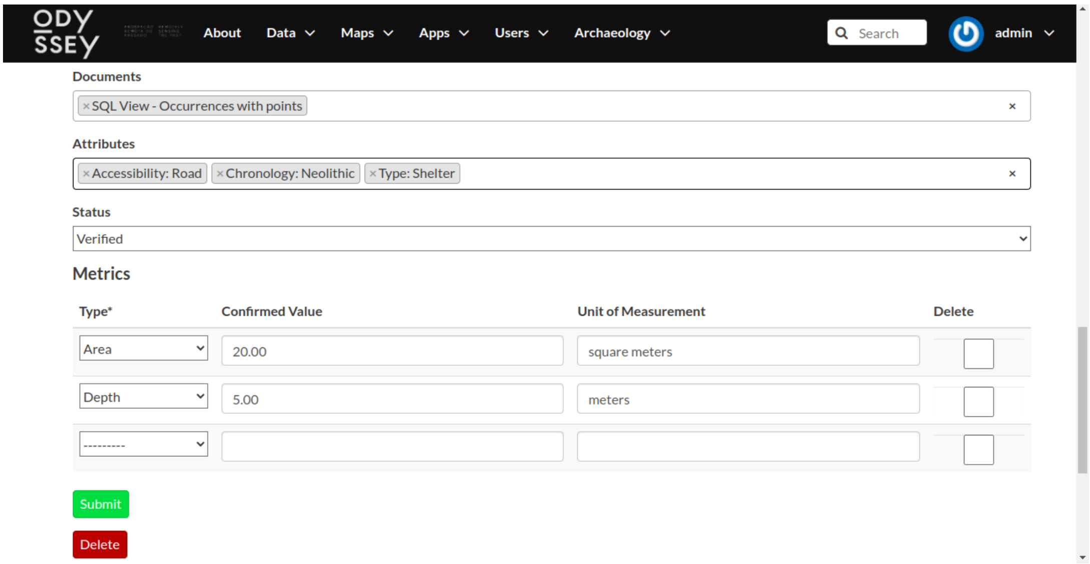

Figure 4.2: Inline Forms for the Metrics of an Occurrence

Whenever sites or instances are created, updated or deleted, the layers are updated as described in Section 4.3. When sites are created and deleted, both site and occurrence layers are updated. However, when it is only editing a site or occurrence, or creating new occurrences, only the corresponding layers are updated.

The following subsections describe in detail two features beyond CRUD operations, the search for sites and occurrences and the occurrences import.

4.5.1 Search Feature

The archaeological sites and occurrences have pages listing all the instances, allowing to search through filters.

For the implementation of the search tool, a first approach was tested using the existing GeoNode search engine. GeoNode uses Haystack[10] as a search plugin with ElasticSearch as search engine. Therefore, the initial idea would be to create new search indexes in Haystack

[10] `https://django-haystack.readthedocs.io/` (last accessed Oct. 11, 2022)

for the sites and occurrences to take advantage of the structure that already exists in GeoNode. That said, the indexes were created and, according to Haystack's documentation, it would be necessary to update or rebuild the indexes to include the newly created indexes. However, the commands needed to perform such operations, *update_index* or *rebuild_index*, are not available in GeoNode. The GeoNode documentation indicates that it is necessary to enable the Haystack Search Backend Configuration via the *haystack_search* property. However, enabling this property caused an ElasticSearch connection error to appear. Since the error could not be fixed, the problem was posted on the GeoNode community forum, but no feedback was received. As the problem persisted, it was necessary to use a new approach for the search tool. It was decided to use the query tool that Django provides, as the search filters would not be too complex and it supports querying spatial data.

Figure 4.3 presents the search page for archaeological sites. The search filters are displayed on the left side of the page, while the results are displayed on the right side. Both the site and occurrence search allow searching by text or by geographic extent. To search by geographic extent, a map is presented where, through zoom-in and zoom-out operations, it is possible to select a bounding box, which in this case is the visible area of the map. In this type of search, the sites and occurrences whose point or polygon intersects the bounding box are listed. For the text search, it is possible to filter archaeological sites by name, national site code, parish, and attributes. As for the occurrences, it is possible to filter by designation, name of the site to which it belongs, owner, attributes and altitude. In alphanumeric filters the search is case insensitive and presents the results that contain the search string. For numeric filters, if the query is for the national site code, it returns the result that corresponds exactly to the code that was specified. If it is the altitude, the search is done within a range between the minimum and maximum altitude inserted. When no search string is entered, all sites or occurrences are listed.

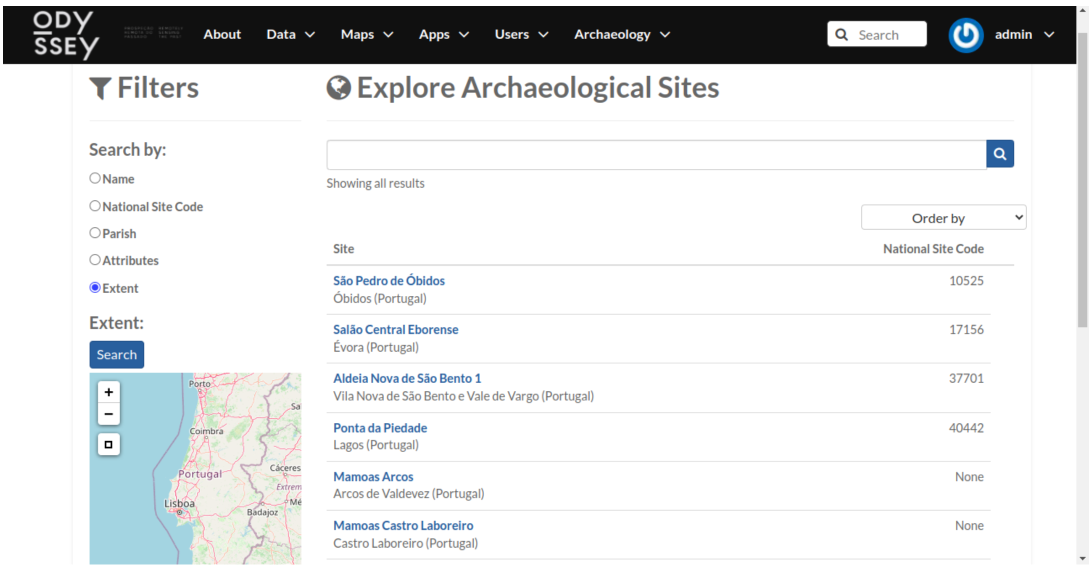

Figure 4.3: Search Page for Archaeological Sites

It is also possible to sort the search results. For the sites it is possible to sort by most or least recent, by name in ascending and descending order, and by national site code, also in ascending and descending order. Similarly, it is possible to sort the occurrences by most or least recent, by name in ascending or descending order, and by the name of the site to which the occurrences belong, in ascending or descending order.

For a better usability, the results are paginated using the *Paginator* class that Django provides. Thus, the results are presented with a limit of ten results per page.

4.5.2 Import Occurrences

It is possible to import occurrences through files for a given archaeological site. This functionality is particularly useful as external tools, such as QGIS, can be used to annotate occurrences. The import tool only supports Comma-Separated Values (CSV) files that must be in a predefined format. An example of the format is presented in Code Snippet 2.

```
WKT,Id
"MULTIPOLYGON (((-30482.2483797024 252602.781327305,-30466.7142161098 252603.595021589,-30466.9361327325 252587.690996958,-30482.0264630797 252587.764969166,-30482.2483797024 252602.781327305)))","Mamoa"
"MULTIPOLYGON (((-30052.7657424689 239969.067994009,-30034.2726905728 239970.325521538,-30033.9028295349 239951.314664189,-30052.6917702613 239952.720136133,-30052.7657424689 239969.067994009)))","Shelter"
```

Code 2: Example of a WKT file to import

The first line must contain the header, with the keywords "WKT" and "Id". The first column, "WKT", should contain the polygon(s) delimiting the occurrences or the point(s) indicating their location in Well-Known Text (WKT) format. The polygons must be of MultiPolygon type, and the points of MultiPoint type. This type must be complied with even if there is only one occurrence and, therefore, only one point or polygon. The SRS should be the one used in the Odyssey project, namely ETRS89 / Portugal TM06 - EPSG:3763. The second column, "Id", should contain the type of the occurrence(s) of the respective MultiPolygon or MultiPoint. For proper operation, the Id must match textually with an attribute defined in GeoNode, so that the type of the occurrence is automatically assigned.

To insert the information from the file to the database, it was necessary to use raw SQL, as the Django ORM does not have the specific PostGIS functions that are needed. First, the MultiPolygons and MultiPoints are expanded to Polygons and Points using the PostGIS function ST_Dump. The original SRS of these polygons and points is specified using the ST_SetSRID function, and transformed with the ST_Transform function to the WGS84 - EPSG:4326 reference system.

Then, the type of the geometry is checked, using the function ST_GeometryType, and depending on whether it is of type "ST_Point" or "ST_Polygon" it is stored in the respective field in the database. By default, the name of the occurrences is the Id, followed by a sequential integer number. After importing all occurrences from the file, the occurrence layers are updated.

4.6 Integration with the Machine Learning Algorithms

As mentioned in Section 3.1.4, where the use case to Identify Archaeological Sites was presented, the integration of the platform with the ML algorithms was dependent on the work of other teams of the Odyssey project. Nevertheless, it was possible to perform the integration with the algorithms, in a joint work with the Odyssey project member responsible for the algorithms. The information that needs to be sent to the web service, and its format, was defined collaboratively so that the algorithms could work properly. The workflow of the main activities for the execution of the ML algorithms is represented in the activity diagram of Figure 4.4.

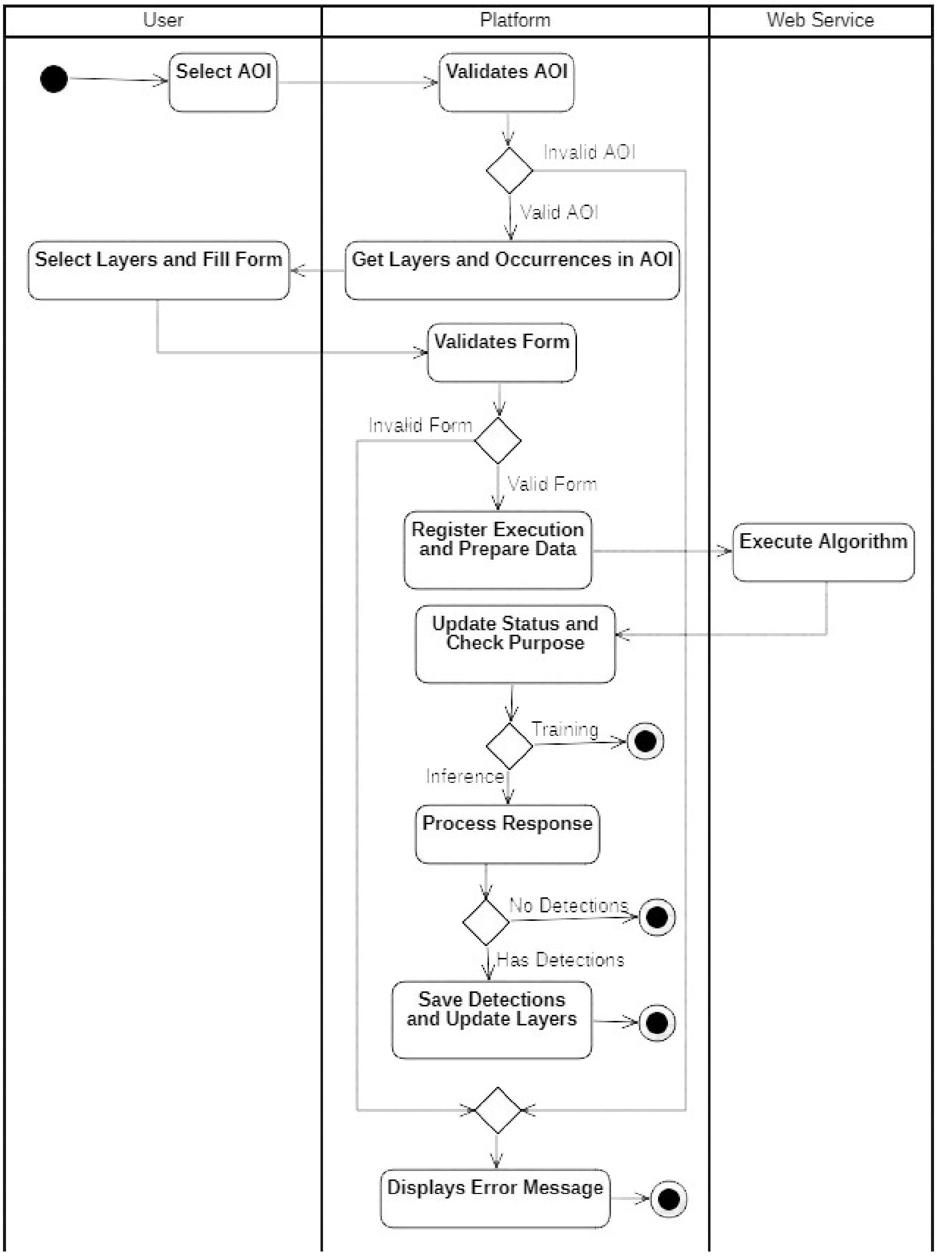

Figure 4.4: Activity Diagram for the Execution of the ML Algorithms

To proceed with the execution of the algorithms, the user must first select an AOI, which will be the bounding box for the execution of the algorithms. The AOI must intersect at least one layer and contain at least one occurrence, as this is the data required to run the algorithms. For the execution of the algorithms, only GeoTIFFs are considered as layers. The layer name must be consistent, following the format *name_type.tif*, the type being the visualization technique or type of processing applied to the layer. For the occurrences, only those that have a defined polygon, have the status "Verified" or "Verified - True Positive" and have the type of occurrence assigned are considered. It is essential to have the polygon defined, as the point does not define the geographical limits of the occurrence and therefore cannot be used by the algorithms. The need to have the occurrence type assigned is because this is how the algorithms distinguishes between occurrences. Finally, occurrences with the status "Not Verified" and "Verified - False Positive" are not considered to not mislead the algorithms. If one of these restrictions is not met, the user is notified and required to select a new AOI. Furthermore, and considering the size of the layers that need to be sent to the algorithms, the size of the AOI should not be excessively large. No programmatic verification of the area size is done, leaving it up to the user to define an AOI of reasonable size.

If the AOI is valid, the user is presented with the list of valid occurrences within the AOI and the layers that intersect it, as shown in Figure 4.5. The user has the possibility to select which layers to send to the algorithms, which can be useful since, depending on the type of processing that the layer has undergone, different results may be obtained. Also, the user must provide a title to identify the execution and indicate the purpose of the execution. If it is "Inference", the algorithms will try to identify new occurrences within the area indicated. Otherwise, if it is "Training", the algorithms will use the information to improve its models.

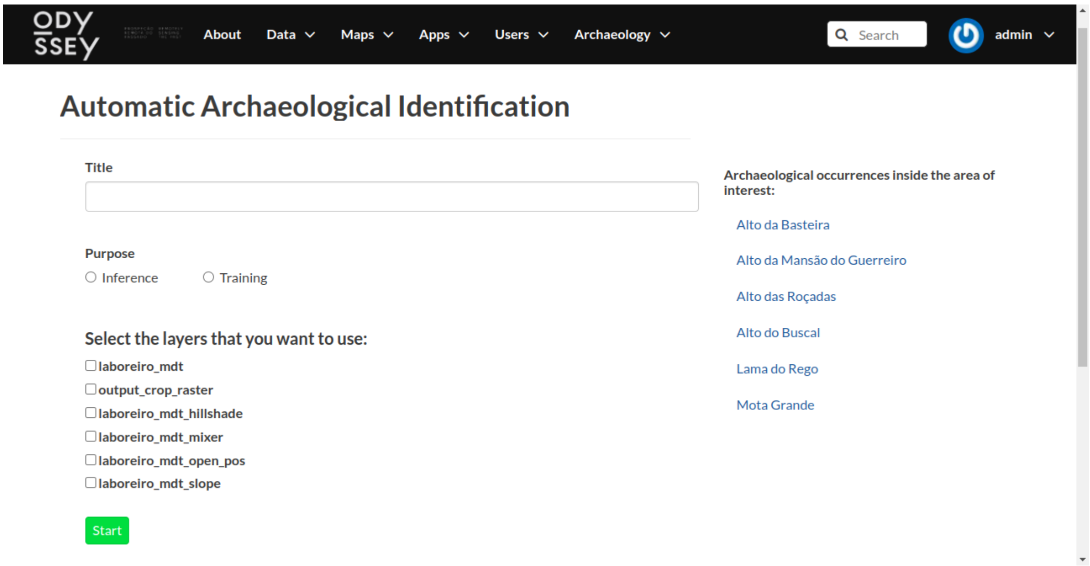

Figure 4.5: Page for Filling in Information and Selecting Layers to Start the Algorithms Execution

Then, the selected layers and occurrences are prepared to be sent to the web service. The information is sent in two JSON objects, one for the occurrences and other for the layers, presented in the Code Snippets 3 and 4, respectively. For the occurrences, the key is the type of the occurrences and the value is the polygons of all the occurrences of that type grouped in a MultiPolygon in WKT format. For the layers, the key is the layer name and the value is a string representing the layer converted to binary format in base64 encoding.

```
{
   "Mamoa":"MULTIPOLYGON (((-29241.2906252581 241680.89583582,-29221.2441570028
   241680.969808027,-29221.8359346635 241662.920589377,-29241.2166530505
   241662.772644962,-29241.2906252581 241680.89583582)), ((...)))",
   "Shelter":"MULTIPOLYGON (((...)), ((...)), ((...)), ((...)))"
}
```

Code 3: Example of a JSON Object with Occurrences for the Web Service

```
{
   "laboreiro_mdt":"SUkqAAgAAAAAAAECsAAAAIKEDAAGAAAAxqoAAK+HAwAgAAAA9qo..."
}
```

Code 4: Example of a JSON Object with Layers for the Web Service

To provide the occurrences in the format that is required, an SQL query is made for each occurrence type, where the bounding polygons of the occurrences of that type are aggregated and converted to a MultiPolygon, using the PostGIS functions ST_Union and ST_Multi, respectively. Then, the MultiPolygon coordinates are transformed to the ETRS89 / Portugal TM06 - EPSG:3763 reference system, which is the SRS of the layers, using the ST_Transform function, and finally converted to WKT using the ST_AsText function.

For the layers, as they have a large size and the bounding box may not cover their whole area, they are cropped by the bounding box and only the relevant portion is sent to the web service. To do this, the coordinates of the bounding box need to be converted from the WGS84 - EPSG:4326 reference system to ETRS89 / Portugal TM06 - EPSG:3763. This is done using the *pyproj*[11] *transform* function, which is a Python interface to the PROJ library for coordinate transformations. Then, a temporary file with the cropped layer needs to be obtained, for which Geospatial Data Abstraction Library (GDAL)[12] was used. GDAL is a library for manipulating geospatial data, and one of its tools is the *warp* function, which allows to define the georeferenced limits of the output file. This way, the file cropped by the AOI that the user defined is obtained and, to provide it in the required format for the web service to process correctly, the file is encoded in base64[13] using the *b64encode* function and is converted to an ASCII string using the *decode* function. In the end, the temporary files created with the cropped layers are deleted.

Once the JSON objects are created, they are sent to the web service and, if the purpose of the execution is "Inference", then a response is expected from the web service. This process is executed within a thread, since sending the information and executing the algorithms are time consuming tasks. Upon receiving the response from the web service, it is verified whether the algorithms were able to detect new occurrences. If it has, a new archaeological site is

[11] `https://pyproj4.github.io/pyproj` (last acessed Oct. 13, 2022)
[12] `https://gdal.org/` (last acessed Oct. 13, 2022)
[13] `https://docs.python.org/3/library/base64.html` (last acessed Oct. 13, 2022)

automatically created, with the title given to the execution of the algorithms by default, and the new detected occurrences are inserted. The format of the web service response is the same as the one that is sent, as shown in Code Snippet 3, and therefore the process to get the information is the reverse. The MultiPolygon is expanded to individual polygons using the ST_Dump function, and are then transformed to the WGS84 - EPSG:4326 reference system to be stored in the database. By default the status of the occurrences detected by the algorithms is "Not Verified", as they need to be verified in the field. After being verified, the status should be changed to "Verified - True Positive" or "Verified - False Positive", depending on whether it was in fact an archaeological occurrence or not. Using these status can be useful for calculating metrics of the algorithms, such as its accuracy.

The algorithm's execution history is saved, and it is possible to check whether a given execution has finished or is still being executed. Consulting the history can be useful to find which layers got the best results. A search tool is also provided, similar to the one presented in Section 4.5.1, where it is possible to search by name of execution or by the geographic extent of the AOI. The search results can be sorted by most and least recent, and by name in ascending or descending order. Additionally, the percentage of detected occurrences with the status "Not Verified", "Verified - True Positive" and "Verified - False Positive" is displayed.

4.7 Metadata

GeoNode uses the ISO 19115: Geographic Information - Metadata [89] schema, so it is not INSPIRE compliant. The INSPIRE directive does have a relationship with a number of ISO 19115 metadata elements, being in some cases more demanding or restrictive, but also has mandatory elements that go beyond those defined in the ISO standard, as described in the technical guidance document [90].

Concerning the INSPIRE directive, the documentation of pycsw, GeoNode's default catalog service, states that it supports INSPIRE Discovery Services by enabling the *metadata_inspire* property. INSPIRE Discovery Services allow for metadata-based search of spatial data and services and the display of the metadata content [45]. This option is enabled by default in GeoNode, which appends an Extended Capabilites section with information required by INSPIRE when performing a *GetCapabilities* request to get the catalog service metadata.

GeoNode also allows the loading of thesaurus, which enables the insertion of the General Multilingual Environmental Thesaurus (GEMET) - INSPIRE Spatial Data Themes[14]. These spatial data themes characterize the layers, defining the subject of the elements they contain. Thus, INSPIRE themes have been added to GeoNode, and a theme can be assigned to the layers in an INSPIRE-compliant manner. However, GeoNode does not make any further reference to the INSPIRE directive, or any means of making it fully INSPIRE-compliant. Yet, community members have had similar issues and posted their insights in the GeoNode forums. The recommendation is to use GeoNetwork as a metadata catalog instead of pycsw, as it is already INSPIRE compliant. Although GeoNode allows the configuration of GeoNetwork as

[14] `https://www.eionet.europa.eu/gemet/en/inspire-themes/` (last acessed Oct. 28, 2022)

an alternative catalog service, community members have reported problems with the metadata download services malfunctioning.

Since configuring GeoNode, and its services, to be INSPIRE compliant does not present a concrete solution, it was agreed that no further effort would be invested in this task, at least at this stage of the Odyssey project, as it is not a crucial task.

4.8 Other Features

The internationalization of GeoNode is done using the i18n module that Django has built-in, which is another of its advantages. The translation strings are obtained by tagging the strings in the templates and using the *gettext* function in the python files, which are then gathered into message files for the languages to be translated. After adding the translations to the files, the messages are compiled and are ready to be used. The GeoNode core already has translations to 32 languages, although in the context of the project, only translations to English and Portuguese are required. The features that were implemented were written in English, and then the translations to Portuguese were created.

Concerning the version history, the proposed solution would be the use of temporal tables, introduced in the SQL:2011 standard [91]. These tables keep the history of data changes, allowing the user to know the state of the data at a given moment in time. Currently PostgreSQL does not support temporal tables natively, but there is an extension[15] available for this purpose. However, it was decided that this feature would not be implemented during this stage of the Odyssey project, as its priority was low and the time available was a limitation.

[15] `https://pgxn.org/dist/temporal_tables/` (last acessed Oct. 24, 2022)

CHAPTER 5

Evaluation

This chapter describes the data that was used to perform early tests on the tool, as well as an actual archaeological site used to test the tool. In addition, it describes the demonstration of the tool given to the Odyssey project stakeholders and the feedback received.

5.1 Test Data

The data initially available for testing during the development of the platform consisted of GeoTIFFs and shapefiles with information about the Alto Minho area. The GeoTIFFs contained LiDAR derived products, such as DTMs and LRMs, while the shapefiles contained manually annotated archaeological occurrences of that area, specifically Mamoas. These data are representative of the type of vector and raster data that will be uploaded into the platform. The files were uploaded to the platform and the layers were visualized on a map using the functionalities already provided by GeoNode. Figure 5.1 shows the data for Castro Laboreiro represented on the GeoNode map.

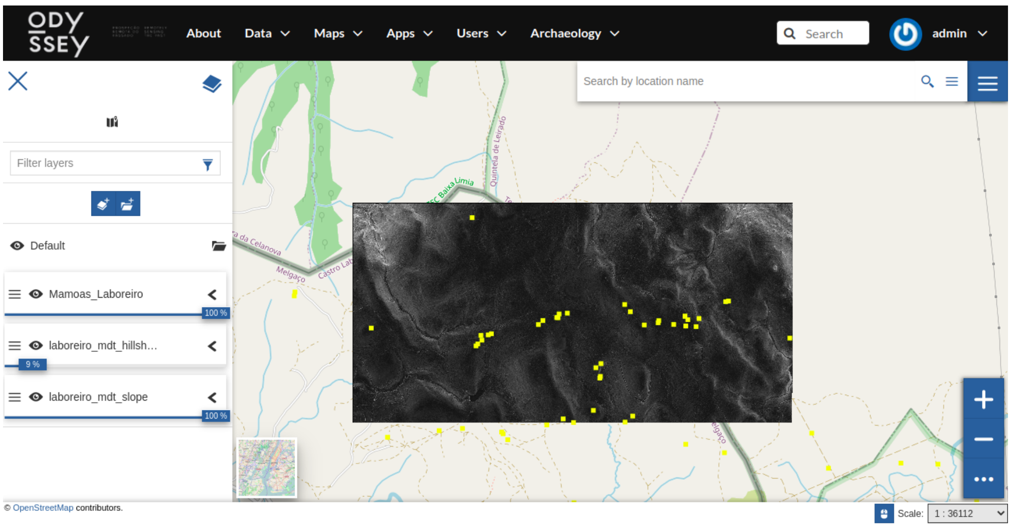

Figure 5.1: GeoNode Map with Layers from Castro Laboreiro

Regarding the attributes to be inserted into the platform, ERA provided a list of attributes and their possible values, adapted from the Endovélico[1] thesaurus, the Portuguese archaeological information and management system. The list of attributes can be found in Appendix A.

As for the metrics, the main types of metrics to be stored are height, length, width, diameter and area. This information was inserted into the platform through the Django administration interface because, as mentioned in Section 4.4, this data does not change regularly. With this, the user has the ability to enter metrics and select attributes for an occurrence as shown in Figure 5.2.

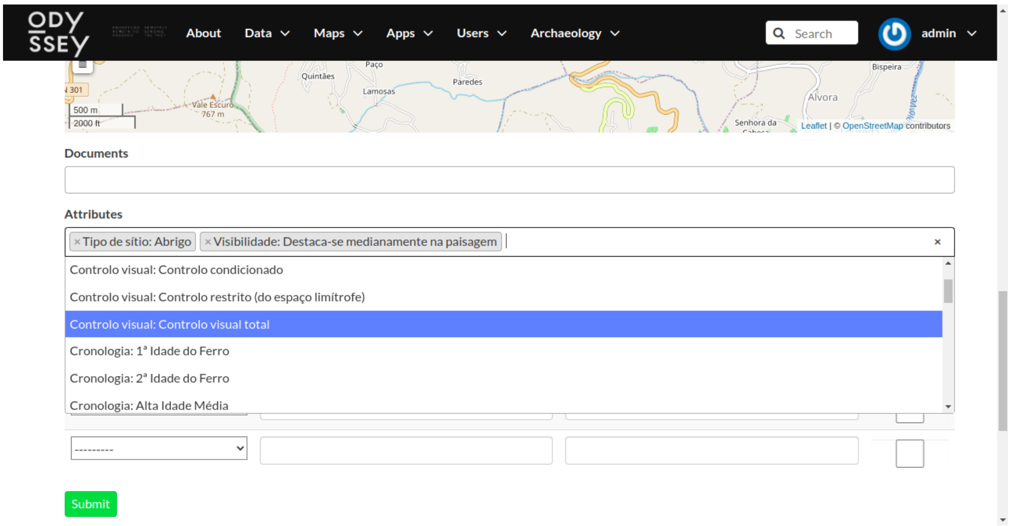

Figure 5.2: Attribute Selection Widget

To test the import of occurrences, the shapefiles with the annotations of Mamoas were converted to WKT format with QGIS desktop tool. By using the generated CSV file, the Mamoas were successfully imported into a test site with their type being assigned automatically.

As base layers, the layers mentioned in Section 3.2.1 were added. The CAOP, COS and Geological Map were added as remote service via WMS. The OpenStreetMap features were added as shapefiles. Although military maps are described in the requirements definition as one of the layers to be added, being these maps paid for, none were provided to be tested.

For a more realistic test, data from a real archaeological site was provided to be uploaded to the platform. The archaeological site is called *Monte Sapo 1/ Herdade da Calçada de Baixo 1* and is located in Santa Clara de Louredo, Beja. The record of the archaeological site, where its information is presented, is available at *Portal do Arqueólogo*[2]. Besides the base information, the aim is also to upload a GeoTIFF with the DTM of the site and a PDF with notes to be linked to the archaeological site. The data was successfully inserted in the platform, but some textual information from the archaeological site description presented in the portal

[1] `http://patrimoniocultural.gov.pt/pt/patrimonio/patrimonio-imovel/patrimonio-arqueologico/endovelico-inventario/` (last acessed Oct. 21, 2022)

[2] `https://arqueologia.patrimoniocultural.pt/index.php?sid=sitios&subsid=3092669` (last acessed Oct. 21, 2022)

could not be included through the existing fields. Although in the requirements definition with ERA the need for a field to provide a longer description of the site or occurrence was not mentioned, the transposition of the information from the *Portal do Arqueólogo* to the platform revealed this need so that no information is misplaced. Nevertheless, this information can always be added in document format and linked to the site or occurrence. Figure 5.3 shows the visualization of the site and occurrences on the map, along with the information of one of the site's occurrences.

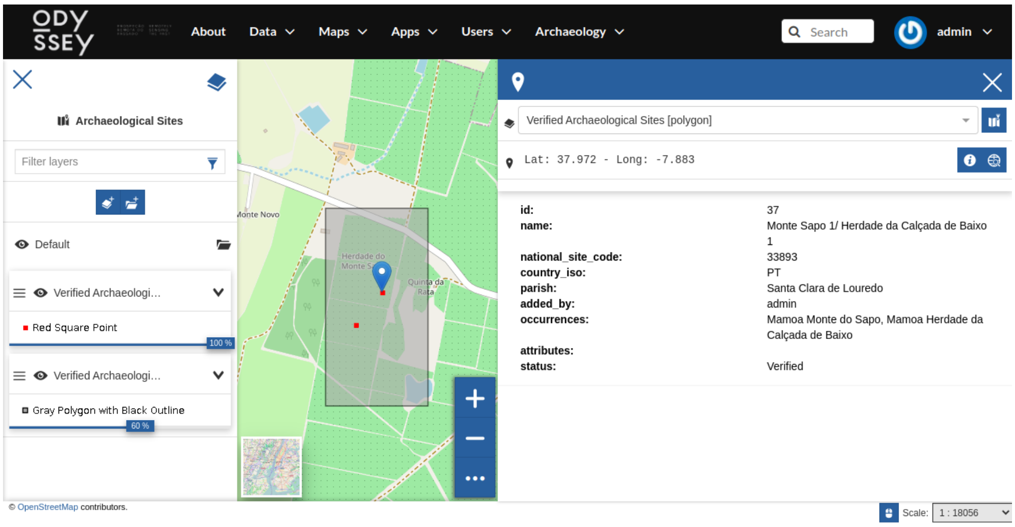

Figure 5.3: Representation of the Site and Occurrences on a Map with Information of an Occurrence

5.2 Demonstration

A demonstration of the tool was made to all parties involved in the Odyssey project, with a special interest from ERA Arqueologia, which is the main stakeholder of the platform. For this demonstration, the test data that was provided was used to present the tool as similar as possible to the real use.

This demonstration allowed a preliminary evaluation of the platform, so that feedback could be obtained. At the moment of the demonstration, the integration with the web service for the ML algorithms had not been implemented yet, and therefore the demonstration consisted in the presentation of the main functionalities of GeoNode, as well as the functionalities already implemented. The feedback obtained through the demonstration was positive, with the functionalities of the platform being in line with ERA's expectations. One of the questions raised was whether it would be possible to connect the geoportal[3] of *Portal do Arqueólogo* as a remote service, so that the information would always be up-to-date. However, this information is available through WFS, and GeoNode only allows adding remote services via WMS. Nevertheless, it is possible to obtain the geoportal information in shapefile format, which can then be uploaded as a layer, with the disadvantage of having to be updated manually.

[3] https://patrimoniodgpc.maps.arcgis.com/apps/webappviewer/index.html?id=5cb4735d7d7743a39a16d7269a753a4a%20 (last acessed Oct. 21, 2022)

Another question was whether it was possible to extract a PDF report with the information of an archaeological site, its occurrences, and all its attributes with the site record template. This functionality was not mentioned during the requirements definition, so it was not considered and is not implemented. However, this would be a useful functionality, as the platform contains all the necessary information necessary for the site record. Furthermore, the creation of this record is part of the archaeologist's work process, and ERA currently fills in this record manually, so automating the process would be an additional benefit.

The next step would be for ERA itself to simulate the actual use of the platform to better evaluate its functionalities and usability. However, it was not possible to perform this simulation with ERA at the time of writing.

5.3 Tests of the Web Service Integration

At the time of writing this document, the web service responsible for executing the ML algorithms was not yet live for testing. For this reason, the web service responses were simulated, with the purpose of verifying that the information received was correctly processed by the platform.

A test was performed with data from the Castro Laboreiro region, where one layer was selected to be sent to the algorithms, with the execution being subsequently stored in the history. The obtained response simulates the detection of two new occurrences, which were correctly added to the new site created automatically.

To verify that occurrences are correctly assigned to the layer depending on their status, the status of one of the detected occurrences was changed to "Verified - True Positive". By visualizing the layers using different styles, namely a solid border for the "Verified - True Positive" and a line pattern for the "Not Verified", it is possible to confirm the representation of the occurrences on the appropriate layers, as shown in Figure 5.4.

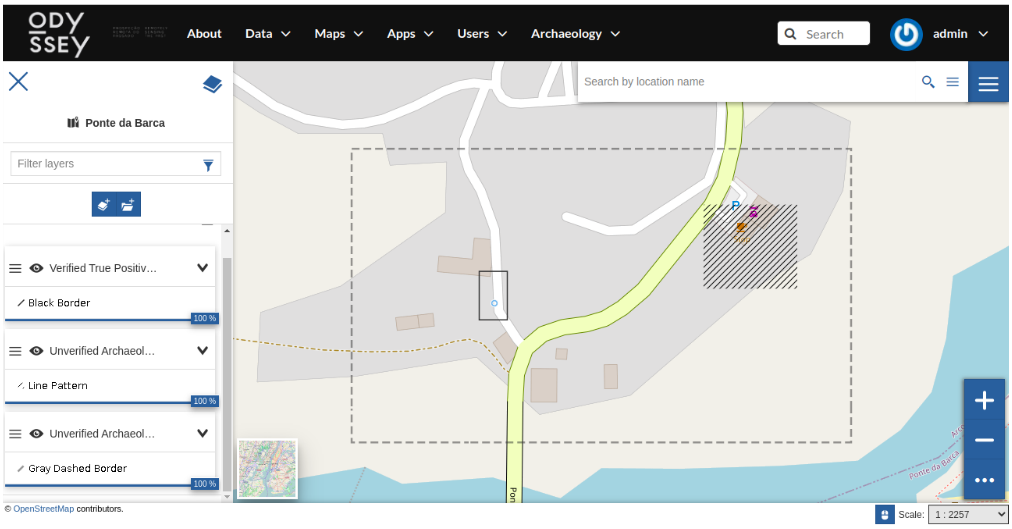

Figure 5.4: Representation of Layers on a Map with Different Styles

CHAPTER 6

Conclusions and Future Work

This chapter provides a summary of the work carried out, presenting its main conclusions. In addition, it also presents possible future work that can be done.

6.1 Conclusions

Archaeology has been taking advantage of technological developments, by using technologies such as LiDAR to obtain georeferenced data that, after being processed, can be used to apply remote sensing techniques for archaeological identifications. However, the archaeological work procedures can still be improved, as there are no tools capable of automating or even replacing human efforts to analyze large amounts of data for the identification of archaeological sites [1].

Thus, this project focused on the need to develop a platform that would be able to consolidate and handle different types of georeferenced data, and to automate the remote sensing of archaeological sites, through ML algorithms.

For this purpose, the main standards were analyzed and the typical architecture of this type of systems was detailed, which helped to identify its main components and the FOSS tools available for its implementation. This analysis highlighted GeoNode as a fully featured tool that could be extended and modified to meet the project's requirements, being this the tool chosen as the basis for the platform's implementation. Additionally, a review of works related to this project was performed, which helped to understand what are the main functionalities commonly found in web GISs and SDIs.

Looking at the platform implementation, and comparing it with the requirements and architecture initially defined, it is possible to verify that the main goals were achieved. The use of GeoNode proved to be a good decision, since it facilitated the development of the platform, having already all the common components in this type of platforms. This allowed focusing the development on the specific functionalities of this project, such as the implementation of the data model and the availability of georeferenced data through normalized services, the implementation of CRUD operations over the data, the development of search and import functionalities of archaeological sites and occurrences, and the integration with the ML algorithms.

Comparing the developed platform, based on the requirements defined and the functionalities provided by GeoNode, with related works analyzed such as Anteo [64], it can be confirmed that the platform incorporates several common functionalities among these works. Additionally, the platform stands out from the others with the integration of ML algorithms, thereby automating the process of archaeological site identification and its consolidation with existing information.

The tests performed to the platform, and the demonstration given to the Odyssey project stakeholders, allowed a preliminary evaluation of the platform. ERA, as the main stakeholder, provided positive feedback, pointing out possible future functionalities to increase the platform's potential.

That said, the platform developed in this project provides a solid basis for the ongoing Odyssey project. Furthermore, it will improve the quality of the information provided and increase the efficiency of the operations carried out by the ERA.

6.2 Future Work

As mentioned in Chapter 5, the next step in the evaluation of the platform is to be the ERA staff itself to interact directly with it by simulating the use in a real context, in order to evaluate the developed functionalities and the usability of the platform.

Regarding the web platform, the requirements initially planned that were not implemented must be met, such as full compliance with the INSPIRE directive, by making changes to the metadata schema, and the version history, to allow previous versions of the data to be consulted and restored. Also, further integrations can be made with the capabilities that GeoNode offers, to maximize the use of the tool. The authorization module that manages permissions in GeoNode can be integrated with the functionalities developed in this project. Moreover, GeoNode provides a REST API, which can be extended to incorporate the developed functionalities.

Finally, as one of the objectives of the Odyssey project is the possibility to use the platform to support data validation in the field, the mobile version of the platform should be developed considering the aspects that were mentioned in the platform requirements.

References

[1] ERA - Arqueologia, “Proposta de Candidatura Projeto Odyssey - Anexo Técnico,” Tech. Rep., 2019.

[2] A. Menéndez Blanco, J. García Sánchez, J. M. Costa-García, J. Fonte, D. González-Álvarez, and V. Vicente García, “Following the Roman Army between the Southern Foothills of the Cantabrian Mountains and the Northern Plains of Castile and León (North of Spain): Archaeological Applications of Remote Sensing and Geospatial Tools,” *Geosciences*, vol. 10, no. 12, p. 485, Dec. 2020, ISSN: 2076-3263. DOI: 10.3390/geosciences10120485.

[3] L. Luo *et al.*, “Airborne and spaceborne remote sensing for archaeological and cultural heritage applications: A review of the century (1907–2017),” *Remote Sensing of Environment*, vol. 232, p. 111 280, Oct. 2019, ISSN: 00344257. DOI: 10.1016/j.rse.2019.111280.

[4] S. H. Parcak, *GIS, Remote Sensing, and Landscape Archaeology*, September. Oxford University Press, Mar. 2017, vol. 1, pp. 1–32, ISBN: 9780199935413. DOI: 10.1093/oxfordhb/9780199935413.013.11.

[5] Q. Zhou, “Digital Elevation Model and Digital Surface Model,” in *International Encyclopedia of Geography: People, the Earth, Environment and Technology*, D. Richardson, N. Castree, M. Goodchild, A. Kobayashi, W. Liu, and R. Marston, Eds. John Wiley and Sons, Ltd, 2017. DOI: 10.1002/9781118786352.wbieg0768.

[6] R. Lasaponara and N. Masini, “Image Enhancement, Feature Extraction and Geospatial Analysis in an Archaeological Perspective,” in *Satellite Remote Sensing. Remote Sensing and Digital Image Processing*, R. Lasaponara and N. Masini, Eds., vol. 16, Springer, Dordrecht, 2012, pp. 17–63, ISBN: 978-90-481-8801-7. DOI: 10.1007/978-90-481-8801-7_2.

[7] R. Hesse, “LiDAR-derived Local Relief Models – a new tool for archaeological prospection,” *Archaeological Prospection*, vol. 17, no. 2, pp. 67–72, Apr. 2010, ISSN: 1099-0763. DOI: 10.1002/ARP.374.

[8] S. Agrawal, D. K. Meena, and R. D. Gupta, “Comprehensive review of GIS and web GIS,” *Discovery*, vol. 41, no. 189, pp. 130–135, 2015, ISSN: 2278-5469.

[9] Y. Sandinska, “Technological Principles and Mapping Applications of Web GIS,” in *Proceedings, 6th International Conference on Cartography and GIS*, T. Bandrova and Konecny M., Eds., Albena, Bulgaria, Jun. 2016, pp. 287–296.

[10] Y. Hu and W. Li, “Spatial Data Infrastructures,” *The Geographic Information Science and Technology Body of Knowledge*, vol. 2017, no. Q2, J. P. Wilson, Ed., Apr. 2017. DOI: 10.22224/gistbok/2017.2.1.

[11] S. Patel, “GIS and SDI in Development Perspective,” *KCG Journal of Multi-Disciplinary*, no. 12, Dec. 2015, ISSN: 2279-0268.

[12] A. K. Tripathi, S. Agrawal, and R. D. Gupta, “Cloud enabled SDI architecture: a review,” *Earth Science Informatics*, vol. 13, no. 2, pp. 211–231, Jun. 2020, ISSN: 1865-0473. DOI: 10.1007/s12145-020-00446-9.

[13] A. Alesheikh, H. Helali, and A. Behroz, “Web GIS: Technologies and Its Applications,” in *Symposium on Geospatial Theory, Processing and Applications*, C. Armenakis and Y. C. Lee, Eds., vol. 34 Part 4, Ottawa, 2002.

[14] Open Geospatial Consortium, *About OGC | OGC*. [Online]. Available: https://www.ogc.org/about (visited on 01/03/2022).

[15] ——, *OGC Standards | OGC*. [Online]. Available: https://www.ogc.org/docs/is (visited on 01/04/2022).

[16] D. Nebert, A. Whiteside, and P. Vretanos, “OpenGIS® Catalogue Services Specification,” Open Geospatial Consortium Inc., Tech. Rep., 2007.

[17] J. de la Beaujardiere, “OpenGIS® Web Map Server Implementation Specification,” Open Geospatial Consortium Inc., Tech. Rep., 2006.

[18] J. Masó, K. Pomakis, and J. Núria, “OpenGIS® Web Map Tile Service Implementation Standard,” Open Geospatial Consortium Inc., Tech. Rep., 2010.

[19] P. A. Vretanos, “OGC® Web Feature Service 2.0 Interface Standard,” Open Geospatial Consortium, Tech. Rep., 2014.

[20] GeoServer, *WFS reference — GeoServer 2.21.x User Manual*. [Online]. Available: `https://docs.geoserver.org/latest/en/user/services/wfs/reference.html` (visited on 01/04/2022).

[21] E. Devys, T. Habermann, C. Heazel, R. Lott, and E. Rouault, “OGC GeoTIFF Standard,” Open Geospatial Consortium, Tech. Rep., 2019.

[22] N. Ritter and M. Ruth, “The GeoTiff data interchange standard for raster geographic images,” *International Journal of Remote Sensing*, vol. 18, no. 7, pp. 1637–1647, May 1997, ISSN: 0143-1161. DOI: `10.1080/014311697218340`.

[23] ESRI, “ESRI Shapefile Technical Description,” Environmental Systems Research Institute, Inc, Tech. Rep., Jul. 1998.

[24] A. Aitchison, “Spatial Reference Systems,” in *Pro Spatial with SQL Server 2012*. Berkeley, CA: Apress, 2012, pp. 1–20, ISBN: 978-1-4302-3492-0. DOI: `10.1007/978-1-4302-3492-0_1`.

[25] INSPIRE Maintenance and Implementation Group (MIG), “Technical Guidance for the implementation of INSPIRE dataset and service metadata based on ISO/TS 19139:2007,” Tech. Rep., 2017.

[26] M. Á. Rodríguez Luaces, B. R. Nieves, J. R. Paramá, and J. R. Viqueira, “A Generic Architecture for Geographic Information System,” *Universidad Da Coruña*, no. July, pp. 1–234, 2004.

[27] A. Tiwari and D. K. Jain, “Geospatial Framework For Dengue using Open Source Web GIS Technology,” *Joint International Workshop of ISPRS WG VIII/1 and WG IV/4 on Geospatial Data for Disaster and Risk Reduction November*, pp. 21–22, Nov. 2013.

[28] M. A. Şahin and M. Yakar, “WebGIS Technology and Architectures,” *Advanced GIS*, vol. 1, no. 1, pp. 22–26, Oct. 2021.

[29] L. Luqun, L. Jian, and T. Yu, “The Study on Web GIS Architecture Based on JNLP,” in *Symposium on Geospatial Theory, Processing and Applications*, C. Armenakis and Y. C. Lee, Eds., vol. 34 Part 4, Ottawa, 2002.

[30] A. Naseer, H. I. Aldoobi, and B. Y. Alkazemi, “A service-oriented architecture for GIS applications,” in *2015 10th International Joint Conference on Software Technologies (ICSOFT)*, vol. 2, 2015, pp. 1–5.

[31] P. Fu, *Web GIS: Principles and Applications*, 2012.

[32] J. Lyons, “Geodatabase vs. Geospatial Database: What’s the Difference?” *Safe Software*, Mar. 2021.

[33] Open Geospatial Consortium, “OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture,” Tech. Rep., 2011.

[34] T. Nyerges, “Spatial Database Management Systems,” *Geographic Information Science and Technology Body of Knowledge*, vol. 2021, no. Q1, Feb. 2021, ISSN: 25772848. DOI: `10.22224/gistbok/2021.1.11`.

[35] J. R. Herring, “OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture,” Open Geospatial Consortium Inc., Tech. Rep., 2011.

[36] S. Agrawal and R. D. Gupta, “Web GIS and its architecture: a review,” *Arabian Journal of Geosciences*, vol. 10, no. 23, p. 518, Dec. 2017, ISSN: 1866-7511. DOI: `10.1007/s12517-017-3296-2`.

[37] L. Vaccari, P. Shvaiko, and M. Marchese, “A geo-service semantic integration in Spatial Data Infrastructures,” *International Journal of Spatial Data Infrastructures Research*, vol. 4, pp. 24–51, 2009. DOI: `10.2902/1725-0463.2009.04.art2`.

[38] MySQL, "11.4 Spatial Data Types," *MySQL 8.0 Reference Manual*, 2021. [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/spatial-types.html.

[39] PostGIS, *PostGIS Features List.* [Online]. Available: https://postgis.net/features/ (visited on 01/05/2022).

[40] OSGeo, *Projects Archive - OSGeo*, 2022. [Online]. Available: https://www.osgeo.org/projects/ (visited on 01/06/2022).

[41] GeoServer, *GeoServer User Manual — GeoServer 2.21.x User Manual*, 2022. [Online]. Available: https://docs.geoserver.org/latest/en/user/ (visited on 01/06/2022).

[42] J. McKenna, D. Fawcett, and H. Butler, "An Introduction to MapServer," *MapServer 7.6.4 documentation*, Jul. 2021. [Online]. Available: https://mapserver.org/introduction.html.

[43] GeoNetwork, *GeoNetwork opensource v3.10 GeoNetwork Documentation*, 2021. [Online]. Available: https://geonetwork-opensource.org/manuals/trunk/en/index.html (visited on 01/07/2022).

[44] T. Kralidis, *pycsw 2.6.1 Documentation*, Oct. 2021. [Online]. Available: https://docs.pycsw.org/en/2.6.1/ (visited on 01/07/2022).

[45] IOC Task Force for Network Services, "Technical Guidance for the implementation of INSPIRE Discovery Services," Tech. Rep., Mar. 2011.

[46] GeoWebCache, *GeoWebCache 1.15.2 User Manual*, Jun. 2019. [Online]. Available: https://www.geowebcache.org/docs/current/index.html (visited on 01/07/2022).

[47] MapProxy, *MapProxy — The accelerating web map proxy.* [Online]. Available: https://mapproxy.org/ (visited on 01/11/2022).

[48] R. García, J. P. de Castro, E. Verdú, M. J. Verdú, and L. M. Regueras, "Web Map Tile Services for Spatial Data Infrastructures: Management and Optimization," in *Cartography - A Tool for Spatial Analysis*, C. Bateira, Ed., IntechOpen, 2012. DOI: 10.5772/46129.

[49] OpenLayers, *OpenLayers - Overview.* [Online]. Available: https://openlayers.org/ (visited on 01/07/2022).

[50] Leaflet, *Leaflet - a JavaScript library for interactive maps.* [Online]. Available: https://leafletjs.com/index.html (visited on 01/07/2022).

[51] M. Khitrin, "Comparison of JavaScript Libraries for Web-Cartography," *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control and Radioelectronics*, vol. 17, no. 3, pp. 147–152, 2017, ISSN: 1991976X. DOI: 10.14529/ctcr170317.

[52] Geoapify, *Leaflet vs OpenLayers. Pros and cons of both libraries | Geoapify*, Apr. 2019. [Online]. Available: https://www.geoapify.com/leaflet-vs-openlayers (visited on 01/07/2022).

[53] Esri, "desktop GIS| Definition," *Esri Support GIS Dictionary*, [Online]. Available: https://support.esri.com/en/other-resources/gis-dictionary/term/960670d1-b08f-43b6-bb71-3564b13a4cde.

[54] QGIS, *QGIS Desktop User Guide/Manual.* [Online]. Available: https://docs.qgis.org/3.16/en/docs/user_manual/ (visited on 01/10/2022).

[55] S. Maurya, A. Ohri, and S. Mishra, "Open Source GIS: A Review," *Proceedings of National Conference on Open Source GIS: Opportunities and Challenges*, no. October, pp. 150–155, 2015, ISSN: 0012-821X.

[56] MapASyst, *What is mobile GIS? – Geospatial Technology*, Aug. 2019. [Online]. Available: https://mapasyst.extension.org/what-is-mobile-gis/ (visited on 01/10/2022).

[57] QField, "QField - your mobile [Q]GIS solution," *QField Documentation 1.0*, [Online]. Available: https://qfield.org/docs/index.html#.

[58] P. Corti, F. Bartoli, A. Fabiani, C. Giovando, A. T. Kralidis, and A. Tzotsos, "GeoNode: an open source framework to build spatial data infrastructures," *PeerJ Preprints*, vol. 7, pp. 1–10, Feb. 2019, ISSN: 2167-9843. DOI: 10.7287/PEERJ.PREPRINTS.27534V1.

[59] M. contributors, *Django introduction*, Sep. 2022. [Online]. Available: `https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction` (visited on 10/17/2022).

[60] S. Buonanno *et al.*, “A GeoNode-Based Platform for an Effective Exploitation of Advanced DInSAR Measurements,” *Remote Sensing*, vol. 11, no. 18, p. 2133, Sep. 2019. DOI: `10.3390/rs11182133`.

[61] B. M. F. Amin and A. Georgios, “Implementation of a Geodata Metadata Portal Site,” Stuttgart Technology University of Applied Sciences, Stuttgart, Tech. Rep., 2015.

[62] A. Scianna and B. Villa, “GIS Applications in archaeology,” *Archeologia e Calcolatori*, vol. 22, pp. 337–363, 2011, ISSN: 1120-6861.

[63] F. Galeazzi and H. Richards-Rissetto, “Editorial Introduction: Web-based Archaeology and Collaborative Research,” *Journal of Field Archaeology*, vol. 43, no. sup1, S1–S8, Oct. 2018, ISSN: 0093-4690. DOI: `10.1080/00934690.2018.1512701`.

[64] P. Gallo and V. Roberto, “ANTEO: Sharing Archaeological Data on a WebGIS Platform,” in *Proceedings of the 2nd Workshop on The New Technologies for Aquileia*, L. Fozzati and V. Roberto, Eds., vol. 948, Aquileia: CEUR Workshop Proceedings, 2012, pp. C1–C10.

[65] Y. Matsui, A. Gidding, T. E. Levy, F. Kuester, and T. A. DeFanti, “ArchaeoSTOR map: publishing archaeological geodata on the web,” in *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications - COM.Geo ’12*, New York, New York, USA: ACM Press, 2012, pp. 1–2, ISBN: 9781450311137. DOI: `10.1145/2345316.2345355`.

[66] J.-P. P. McCool, “PRAGIS: a test case for a web-based archaeological GIS,” *Journal of Archaeological Science*, vol. 41, pp. 133–139, 2014, ISSN: 03054403. DOI: `10.1016/j.jas.2013.07.037`.

[67] S. Bernardoni, M. Montanari, and R. Trojanis, “Open History Map,” *Archeologia e Calcolatori*, vol. 28.2, pp. 539–548, 2017, ISSN: 23851953. DOI: `https://doi.org/10.19282/AC.28.2.2017.44`.

[68] M. Serlorenzi, A. D’Andrea, and R. Montalbano, “SITAR: a new open-data infrastructure for a public archaeology of Rome,” in *ArcheoFOSS XIV 2020: Open Software, Hardware, Processes, Data and Formats in Archaeological Research*, J. Bogdani, R. Montalbano, and P. Rosati, Eds., Oxford: Archaeopress, 2021, pp. 108–118, ISBN: 9781803271255.

[69] I. Lindsay and N. N. Kong, “Using the ArcGIS Collector Mobile App for Settlement Survey Data Collection in Armenia,” *Advances in Archaeological Practice*, vol. 8, no. 4, pp. 322–336, Nov. 2020, ISSN: 2326-3768. DOI: `10.1017/aap.2020.26`.

[70] G. Styliaras, “Towards a web-based archaeological excavation platform for smartphones: review and potentials,” *SpringerPlus*, vol. 4, no. 1, p. 311, Dec. 2015, ISSN: 2193-1801. DOI: `10.1186/s40064-015-1115-3`.

[71] A. Sobotkova *et al.*, “Deploying an Offline, Multi-User, Mobile System for Digital Recording in the Perachora Peninsula, Greece,” *Journal of Field Archaeology*, vol. 46, no. 8, pp. 571–594, Nov. 2021, ISSN: 0093-4690. DOI: `10.1080/00934690.2021.1969837`.

[72] B. Ballsun-Stanton, S. A. Ross, A. Sobotkova, and P. Crook, “FAIMS Mobile: Flexible, open-source software for field research,” *SoftwareX*, vol. 7, pp. 47–52, Jan. 2018, ISSN: 23527110. DOI: `10.1016/j.softx.2017.12.006`.

[73] M. Sullivan, P. Hiscock, and P. Hughes, “Three scales: GIS, GPS and digital site and data recording technology in archaeological salvage at olympic dam in arid South Australia,” *Journal of the Anthropological Society of South Australia*, vol. 38, pp. 85–107, Dec. 2014, ISSN: 10344438.

[74] C. J. H. Ames, M. Shaw, C. A. O’Driscoll, and A. Mackay, “A Multi-User Mobile GIS Solution for Documenting Large Surface Scatters: An Example from the Doring River, South Africa,” *Journal of Field Archaeology*, vol. 45, no. 6, pp. 394–412, Aug. 2020, ISSN: 0093-4690. DOI: `10.1080/00934690.2020.1753321`.

[75] C. Meghini *et al.*, “ARIADNE: A Research Infrastructure for Archaeology,” *Journal on Computing and Cultural Heritage*, vol. 10, no. 3, pp. 1–27, Aug. 2017, ISSN: 1556-4673. DOI: `10.1145/3064527`.

[76] Infrastructure for Spatial Information in the European Community, *INSPIRE Geoportal*. [Online]. Available: `https://inspire-geoportal.ec.europa.eu/index.html` (visited on 01/20/2022).

[77] Food and Agriculture Organization of the United Nations, *FAO Map Catalog.* [Online]. Available: https://data.apps.fao.org/map/catalog/srv/eng/catalog.search#/home (visited on 01/20/2022).

[78] Direção-Geral do Território, *Registo Nacional de Dados Geográficos.* [Online]. Available: https://snig.dgterritorio.gov.pt/rndg/srv/por/catalog.search#/home (visited on 01/20/2022).

[79] Comissão de Coordenação e Desenvolvimento Regional do Centro, *CCDRC - IDECENTRO.* [Online]. Available: http://geon.ccdrc.pt/geonetwork/srv/por/catalog.search#/home (visited on 01/20/2022).

[80] European Environment Agency, *EEA geospatial data catalogue.* [Online]. Available: https://sdi.eea.europa.eu/catalogue/srv/eng/catalog.search#/home (visited on 01/20/2022).

[81] J. D. Richards, "Archaeology Data Service (ADS)," in *Encyclopedia of Global Archaeology*, C. Smith, Ed., 2nd ed. Cham: Springer International Publishing, 2018, pp. 671–673, ISBN: 978-3-319-51726-1. DOI: 10.1007/978-3-319-51726-1_2121-2.

[82] P. Ronzino, V. Acconcia, and A. Falcone, "Towards the integration of spatial data through the Italian Geoportal for Archaeological Resources," in *2018 3rd Digital Heritage International Congress (DigitalHERITAGE) held jointly with 2018 24th International Conference on Virtual Systems and Multimedia (VSMM 2018)*, IEEE, Oct. 2018, pp. 1–5, ISBN: 978-1-7281-0292-4. DOI: 10.1109/DigitalHeritage.2018.8810132.

[83] A. Nsanziyera, H. Rhinane, A. Oujaa, and K. Mubea, "GIS and Remote-Sensing Application in Archaeological Site Mapping in the Awsard Area (Morocco)," *Geosciences*, vol. 8, no. 6, p. 207, Jun. 2018, ISSN: 2076-3263. DOI: 10.3390/geosciences8060207.

[84] A. Di Iorio *et al.*, "Automatic detection of archaeological sites using a hybrid process of Remote Sensing, Gis techniques and a shape detection algorithm," in *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*, IEEE, Apr. 2008, pp. 1–7, ISBN: 978-1-4244-1751-3. DOI: 10.1109/ICTTA.2008.4529910.

[85] A. Di Iorio, D. Biliouris, R. Guzinski, L. B. Hansen, and M. Bagni, "Innovation technologies and applications for coastal archaeological sites FP7-ITACA," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 40, no. 7W3, pp. 1367–1373, Apr. 2015, ISSN: 16821750. DOI: 10.5194/isprsarchives-XL-7-W3-1367-2015.

[86] L. E. O. Lizardo and C. A. Davis, "OMT-G Designer: A Web Tool for Modeling Geographic Databases in OMT-G," in *Advances in Conceptual Modeling. ER 2014. Lecture Notes in Computer Science*, vol. 8823, Springer, Cham, Oct. 2014, pp. 228–233, ISBN: 978-3-319-12256-4. DOI: 10.1007/978-3-319-12256-4_24.

[87] K. A. V. Borges, C. A. D. Junior, and A. H. F. Laender, "Modelagem conceitual de dados geográficos," in *Bancos de Dados Geográficos*, M. A. Casanova, G. Câmara, C. A. D. Junior, L. Vinhas, and G. R. de Queiroz, Eds., Curitiba: Mundogeo, May 2005, ch. 3, pp. 93–142.

[88] ISO/TC 46 Information and documentation, "ISO 3166-1:2020 Codes for the representation of names of countries and their subdivisions," ISO, Tech. Rep., 2020.

[89] ISO/TC 211 Geographic information/Geomatics, "ISO 19115-1:2014 Geographic information — Metadata," ISO, Tech. Rep., 2014.

[90] European Commission Joint Research Centre, "INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119," Tech. Rep., 2013.

[91] ISO/IEC JTC 1/SC 32 Data management and interchange, "ISO/IEC 9075-1:2011 Information technology — Database languages — SQL," ISO, Tech. Rep., 2011.

Appendix A - List of Attributes based on the Endovélico Thesaurus

Atributo	Valores Possíveis
Tipo de sítio	Abrigo, Achado Isolado, Alcaria, Alinhamento, Anfiteatro, Aqueduto, Arte Rupestre, Arranjo de Nascente, Atalaia, Azenha, Balneário, Barragem, Basílica, Calçada, Canalização, Capela, Casal Rústico, Castelo, Cais, Cemitério, Cetária, Chafurdo, Cidade, Circo, Cista, Cisterna, Complexo Industrial, Concheiro, Convento, Criptopórtico, Cromeleque, Curral, Depósito, Edifício com interesse histórico, Eira, Ermida, Escultura, Estrutura com interesse histórico, Fonte, Forja, Forno, Fortificação, Forum, Fossa, Gruta, Hipocausto, Hipódromo, Igreja, Indeterminado, Inscrição, Lagar, Laje Sepulcral, Malaposta, Mancha de Ocupação, Marco, Menir, Mesquita, Miliário, Mina, Moinho de Maré, Moinho de Vento, Monumento Megalítico Funerário, Mosaico, Muralha, Muro, Nicho, Nora, Oficina, Olaria, Palácio, Paço, Pedreira, Pelourinho, Poço, Pombal, Ponte, Povoado, Povoado Fortificado, Recinto, Represa, Salina, Santuário, Sarcófago, Sepultura, Silo, Sinagoga, Talude, Tanque, Teatro, Templo, Termas, Tesouro, Torre, Tulhas, Via, Viaduto, Moinho de Água, Monte, Laje com Covinhas, Pias, Villa, Açude e Dique, Espigueiro, Quinta, Alminha, Cruzeiro
Cronologia	Paleolítico Inferior, Paleolítico Médio, Paleolítico Superior, Epipaleolítico/Mesolítico, Neolítico, Neolítico Antigo, Neolítico Médio, Neolítico Final, Calcolítico, Calcolítico Final, Bronze Pleno, Bronze Final, Idade do Ferro, 1ª Idade do Ferro, 2ª Idade do Ferro, Romano, Romano Republicano, Romano Império, Romano Alto Império, Romano Baixo Império, Idade Média, Alta Idade Média, Baixa Idade Média, Islâmico, Moderno, Contemporâneo, Pré-História Antiga, Pré-História Recente, Proto-História e Indeterminado
Contexto geológico	Granitos, Xistos, Calcários, Aluviões, Coluviões, Areias, Terraço, Depósitos argilosos, Rochas vulcânicas, Dioritos, Arenitos, Terraço fluvial/cascalheira, Outro
Implantação topográfica	Arriba, Planície, Colina suave, Cerro – topo, Cerro – vertente, Espigão de meandro fluvial, Esporão, Escarpa. Plataforma / rechã, Planalto, Praia, Várzea, Leito de rio ou ribeiro
Visibilidade	Destaca-se bem na paisagem, Destaca-se medianamente na paisagem, Diluído na paisagem, Escondido
Controlo visual	Controlo visual total, Controlo condicionado, Controlo restrito (do espaço limítrofe)
Uso do solo	Agrícola, Agrícola regadio, Baldio, Florestal, Industrial, Pastoreio, Turismo, Urbano, Pedreira, Areeiro, Pântano, Aterro, Caminho
Coberto vegetal	Sem vegetação, Vegetação rasteira, Arbustos ou matos densos, Floresta/mata densa, Floresta/mata pouco densa, Montado
Dispersão de Materiais (em área)	Extensa, Média, Pequena, Pontual
Tipo de dispersão	Contínua, Dispersa, Concentrada, Progressiva
Acessibilidade	Via Rápida, Estrada Nacional, Estrada Municipal, Estradão, Caminho de pé posto, Sem acesso
Trabalhos arqueológicos	Conservação/Valorização, Escavação, Sondagem, Levantamento, Prospecção