Universidade de Aveiro
2011

Departamento de Electrónica, Telecomunicações e Informática

António Marcos Rodrigues de Pinho

Mensageiros do Jazz – Sistema de catalogação

Mensageiros do Jazz – Cataloguing System

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Dr. Joaquim Manuel Henriques de Sousa Pinto, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Apoio financeiro da FCT e do FSE no âmbito do III Quadro Comunitário de Apoio.

A ti Filipa *

“Because maybe
You're gonna be the one that saves me
And after all
You're my wonderwall…”

o júri

presidente	Prof. Dr. Joaquim Arnaldo Carvalho Martins Professor Catedrático da Universidade de Aveiro
vogais	
	Prof. Dr. Fernando Joaquim Lopes Moreira Professor Auxiliar do Departamento de Inovação, Ciência e Tecnologia da Universidade Portucalense
	Prof. Dr. Joaquim Manuel Henriques de Sousa Pinto Professor Auxiliar da Universidade de Aveiro (orientador)

agradecimentos

Queria deixar o meu agradecimento:

- Ao Professor Joaquim Sousa Pinto, o meu orientador
- Ao Professor Rui Raposo pelas discussões e ajuda na componente gráfica do portal
- À bibliotecária Cristina Cortês pela ajuda prestada na compreensão das normas de catalogação
- Ao Professor Cláudio Teixeira pela ajuda prestada no “deployment” do portal
- À minha bela e querida namorada Filipa Aguiar
- À Masmorra e à Cave pelos momentos de boa disposição passados e noites das quais nunca me vou esquecer
- A todos os meus amigos que fizeram com que estes últimos anos tenham sido certamente os melhores anos da minha vida
- Aos meus familiares que ao longo do meu percurso académico sempre me apoiaram

palavras-chave

Sistemas de informação, Mensageiros do Jazz, DotNetNuke, UNIMARC

resumo

Com a evolução da WWW (World Wide Web),os portais tornaram-se os principais distribuidores de serviços aos utilizadores. Esses serviços podem assumir várias formas, tais como: e-mail, notícias, blogs, fóruns e sites de busca, entre outros. Além de oferecer todos esses serviços aos utilizadores, estes também servem para armazenar todas as informações relacionadas com esses serviços.

A existência de WCMS (Web Content Management Systems) é muito importante porque vai permitir que qualquer utilizador normal (sem experiência em programação) possa criar, modificar, publicar e compartilhar os seus próprios conteúdos e torná-los visíveis a um grande número de utilizadores através da Internet. Este conteúdo pode ser qualquer tipo de dados, por exemplo, pode-se partilhar arquivos, documentos científicos, vídeos, fotos, entre outros.

O objetivo deste trabalho é construir um portal que permitirá aos utilizadores não-programadores armazenar todo o tipo de arquivos sobre os Mensageiros do Jazz Português e permitir efetuar pesquisas através de uma interface Web amigável. Isso permitirá a publicação de documentos históricos sobre a história e contribuição dos Mensageiros do Jazz Português.

palavras-chave Information Systems, Jazz Messengers, DotNetNuke, UNIMARC

abstract

With the evolution of the WWW (World Wide Web), Web Portals have become the main distributors of services to users. These services can take many forms such as: e-mail, news, blogs, forums and search engines among many others. Besides offering all these services to users, Web Portals have to store all the information related to these services.

In order to create Web Portals it is necessary to have some knowledge on Programming Languages.

The existence of WCMS (Web Content Management Systems) is very important because they will allow any regular user (without programming background) to create, modify, publish and share its own contents and make it visible to a large number of users through the Internet. This content can be any type of data, for example it can be regular files, scientific documents, videos, pictures, etc.

The objective of this work is to build a WCMS to support the Jazz messengers project that will allow non-programmers users to store and search all types of media using a friendly web user interface.

This will allow the publishing of historical documents about the history and contribution of these Jazz Messengers to the Portuguese Jazz.

Contents

List of Images

List of Tables

List of Acronyms

ASP.NET – Active Server Pages .NET
C# - C Sharp
CeSA – Centro de Estudos sobre África e do Desenvolvimento
CMS – Content Management System
CSS – Cascading Style Sheets
DAAB – Data Access Application Blocks
DAL – Data Access Layer
DBMS – Database Management System
DeCA – Department of Communication and Art
DETI – Department of Electronic, Telecommunications and Informatics
DNN – DotNetNuke
FCT – Fundação para a Ciência e aTecnologia
GUI – Graphical User Interface
HTML – HyperText Markup Language
ID - IdentificationIFLA – International Federation of Library Associations and Institutions
IIS – Internet Information System
IT – Information Technology
LAMP – Linux/Apache/MySQL/PHP
MSSQL Server– Microsoft SQL Server
OS – Operating System
PHP – Hypertext Preprocessor
SQL – Structured Query Language
T-SQL – Transact SQL
UML – Unified Modeling Language
UNIMARC– Universal *Ma*chine *R*eadable *C*atalogue or *C*ataloguing
URL – Uniform Resource Locator
W3C – World Wide Web ConsortiumWCMS– Content Management System
WCMS – Web Content Management System
WISA – Windows/IIS/SQL Server/ASP.NET
WWW – World Wide Web
WYSIWYG – What You See Is What You Get
XML – Extensible Markup Language
XPath – XML Path Language
XSL – Extensible Stylesheet Language
XSLT – Extensible Sytlesheet Language Transformation

Chapter 1 - **Introduction**

1.1 Context

From ancient times, man has always tried to find a way to create and keep historical records so they can be preserved through time. The usual way to do this is through the writing of books and documents. With the evolution of computers, new ways to store data were created. Not only this will allow us to store huge amounts of data, but it will also make this data available to be searched and shared to millions of users through the Internet.

Web Archiving is being used to store all kinds of data in large online catalogues that will allow the use of electronic research tools. Google has already started to archive huge amounts of data from documents, pictures and media from many libraries and private institutions. In fact recently, Google has created the biggest online catalogue with pictures from the Holocaust [1]. This is just one example of what effort has been done lately, in order to collect all the historical information that is available worldwide and archive it into online catalogues.

The creation of these online catalogues will allow an easy way for everyone to make research and search for information about a certain event or historical age.

The storage and representation of this historical data must be done following specific cataloguing standards.

Although these online catalogues will allow the storage of huge amounts of data, they will also bring new security threats, which must be treated with care so that confidential data will remain secure and won't be compromised.

1.2 Objectives

The storage of historical/cultural records must follow several cataloguing and bibliographic standards. Usually the cataloguing experts lack the technical knowledge to develop complex information systems that store and make that information available.

The "Jazz Messengers" project is being developed in partnership with the Department of Communication and Art of University of Aveiro (DeCA) and the Department of Electronics, Telecommunications and Informatics (DETI). Its objective is to develop a web platform that will support the cataloguing activities over several documents related to the Portuguese Jazz Messengers.

The web portal should allow any user to search and view some of the contents available. In order to achieve this, the development of an entire back office must be done so that the management and creation of these contents may be done.

Besides having people related to the music area this project also counts with the support of designers, cataloguing experts and IT experts.

1.3 Motivation

"Mensageiros do Jazz" [10] stands for Jazz Messengers and it is an research project funded by FCT (Fundação para a Ciência e a Tecnologia). The general goal of this project is to start a central area for understanding the phenomenon of jazz in Portugal in the twentieth century and to create a seminal project within Jazz Studies.

Jazz studies are an emergent academic domain in Portugal, just like in other European countries and especially in the U.S. It is urgent to learn about the social, political and musical universe associated with this type of music. Jazz emerges from the U.S. in the beginning of the XX century. Therefore it has survived many years in a "marginal" social space and because of this multiple debates have come to recognize the importance of theJazz Messengers. They are understood as the intermediaries between the concepts/behaviours, associated with Jazz, and the public/institutions.

In Portugal these Jazz advisers were crucial for the inclusion of this type of music in the country's musical background. They include journalists, music critics, music lovers, concert/festival organizers and founders of Jazz music schools. It is mainly because of them that Jazz has become well known and spread across Portugal. This has lead to the creation of the School of Portuguese Jazz.

The project's specific goals are to produce analytical knowledge based on disperse documentation about the presence of jazz in Portugal in the twentieth century. This disperse documentation is to be stored in an online digital platform that was developed in this work. The importance of this is to show how its promoters established jazz along the country and to place Portugal among the international community, which is dedicated to the production of knowledge about the reception of Jazz in Europe during the twentieth century.

1.4 Key Topics

In order to develop the Jazz Messengers information system it is necessary to learn some key concepts. These concepts go from similar information systems like digital libraries to the difference between data, information and knowledge and finally a cataloguing standard.

1.4.1 Digital Libraries

Digital Libraries consist in an Information System (IS) that allows the storage and retrieving of digital documents. These documents can be of all types of data from media to scanned documents. Many institutions have created Digital Libraries so they can store valuable information. The data store in digital libraries can belong to any subject depending on the institution behind the Digital Library and its purpose.

The DELOS Digital Library Reference Model [5] defines a digital library as:

"An organization, which might be virtual, that comprehensively collects, manages and preserves for the long term rich digital content, and offers to its user communities specialized functionality on that content, of measurable quality and according to codified policies"

Marco Fernandes(2010) states that a digital library is:
"An information system which provides online search, selection and dissemination of structured collections of digital services and objects (globally known as resources), and promotes the preservation and integrity of those resources)." [6]

Summarizing Digital Libraries are IS that not only consists on the storage and preservation of information. They also have a well-defined structured model for organizing all the information stored. Moreover it will be possible to relate all the information in the system and even generate knowledge from the fusion of it. Finally it will also allow searches performed by users to become even more accurate.

1.4.2 Knowledge

It is certain that in an IS has data. It can also have information but the existence of knowledge is not guaranteed. Starting with data, almost every IS has a database that stores the system data. This data alone has no meaning at all and needs to be given meaning before it can be treated as information.

The processing of the existent data into useful and meaningful format will lead to the creation of information. Information is easily created from a set of data by giving it meaning so that users can use and search for it.

Knowledge on the other hand is far more difficult to create. It takes more than the processing of information to lead to knowledge. From [7]:
"Knowledge refers to the practical use of information. While information can be transported, stored or shared without many difficulties the same can't be said about knowledge. Knowledge necessarily involves a personal experience. Referring back to the scientific experiment, a third person reading the results will have information about it, while the person who conducted the experiment personally will have knowledge about it."

A person's experience and expertise in addition to information will lead to knowledge.

1.4.3 Universal Machine Readable Catalogue or Cataloguing (UNIMARC)

With the evolution of computers there is a need to store bibliographic information in a standard way so that institutions can easily exchange their bibliographic records with each other through a computer network.

UNIMARC provides ways to allow automated information retrieval and citation display.

UNIMARC stands for "Universal Machine Readable Catalogue or Cataloguing" and its main objective is to define labels that can be handled by computers.

"The primary purpose of UNIMARC is to facilitate the international exchange of data in machine-readable form between national bibliographic agencies". It defines how the data should be stored and represented in computers [2].

"Initially, UNIMARC was used for the exchange of records on magnetic tape but has since been adapted for use in a variety of exchange and processing environments." [2]. Nowadays the Internet is used as a wide tool of information dissemination and sharing. This standard when used online will allow the bibliographic records to be shared and searched worldwide. This standardization will allow IT experts to create applications that will provide new services to be used by users when searching for information related to bibliographic records of an institution. Furthermore since these applications share the same representation it would be very easy to join/merge all the bibliographic records in a single online catalogue and to allow the use of external services that could access this catalogue.

For example imagine several institutions that hold important historical records about a certain event. If all these historical records are stored in the UNIMARC format, then it could be created a catalogue that would aggregate, all this information.

1.5 Methodology

For meeting the projects' objectives the needs of the users had to be listened so that the system requirements and functionalities could be found. Since the "Sistema Integrado para bibliotecas e Arquivos Digitais" (SInBAD) and Memória de África are widely known systems by the users their first request was to develop a system alike them. The main requirements found for the system were:

- An online portal;
- Storage, cataloguing and indexing of periodical publications and articles;
- Search and visualization of those documents;
- Allow users to make comments about catalogued documents.

In order to perform the cataloguing of those documents a standard must be used. For choosing the best-cataloguing standard for the system needs some bibliographic experts were contacted and consulted. After some meetings their advice was that it should be used the UNIMARC standard with an Authority index.

1.6 Document Outline

The rest of this document is structured as follows:

- Chapter 2 reviews the relevant state of art of Digital Libraries, technologies used to develop the online portal, examples of other similar projects and describes UNIMARC.
- Chapter 3 explains the portal's general architecture and how the module developed was built.
- Chapter 4 explains and shows the developed Jazz Messengers portal.
- Chapter 5 gives a full review of the work developed and the final conclusions as well as what could be done in the future to improve the portal.

Chapter 2 - **State of the art**

2.1 Introduction

This chapter gives an introduction to the subjects that were studied in order to develop this work. It starts with the description about a metadata standard called Dublin Core.

The Extensible Stylesheet Language (XSL) was used to develop style sheets to XML. A description of the available web platforms and a comparison between some Web content management systems (WCMS) is also made.

DotNetNuke (DNN) was the framework used to develop this work so it will be given a small introduction to it and to other available .NET web content management systems.

Some digital libraries projects are overviewed as well as the technology used to implement such systems.

Memória de África is an interesting example since it "…tries to explore the historical thread of memories that unite Portugal with Portuguese speaking countries and thus builds a bridge between our common past, enabling establishment of an identity that is shared by the people of these countries" [8] and has a high cultural, social and historical value for Portugal and its ex-colonies. That project's goals are very similar to the Messengers of Jazz project.

SInBAD is another example of a digital library that aggregates the contents from some other digital libraries.

Since UNIMARC is a very complex standard, it was made the decision to only use a part of it that is mandatory so that the documents stored in this platform can be catalogued.

2.2 Dublin Core

Dublin Core metadata standard is a simple yet effective set of metadata elements for describing and cataloguing a wide range of resources [29][27]. These resources can go from books and text files to music and video files. It has 2 levels: the simple and the qualified.

The simple level consists on 15 useful elements that allow the creation of simple and easy to understand descriptions for information resources. In general these 15 elements are sufficient to describe the majority of resources (Table 1).

"Metadata is a key part of the information infrastructure necessary to help create order in the chaos of the Web (…)" [31].

Metadata consists on the information about data. It is a description of data. "Structured descriptive information about a resource" [28]. This information is normally stored in metadata elements. The objectives of using metadata are:

- Allowing an easier discovery of relevant information;

- Organizing electronic resources;
- Integration and interoperability of resources;
- Cataloguing and preservation of resources.

Element	Description
Contributor	An entity responsible for making contributions to the resource
Coverage	The spatial or temporal information of the resource
Creator	The author responsible for creating the resource
Date	A point/period of time associated with the resource
Description	A textual description of the content of the resource,
Format	The data representation of the resource
Identifier	An unambiguous identifier within a given context
Language	The language of the resource
Publisher	The entity responsible for making the resource available
Relation	Relationship with other resources
Rights	Information about rights held in and over the resource.
Source	Another resource from which the resource is derived
Subject	The topic of the resource
Title	Title of the resource
Type	The nature or genre of the resource.

Table 1 - Dublin Core's simple elements

The qualified level has three more elements and a group of qualifiers [29]. Although the Dublin Core scheme was used to describe the common attributes of all resources, its XML was adapted to include description values from other standards [6]. This proves that Dublin Core is extensible.

2.3 XSL

The XSL is part of the World Wide Web Consortium (W3C) [16] Style sheets Activity. It is a group of 3 languages used to transform and render XML documents. These languages are [17]:

- XSLT (Extensible Stylesheet Language);
- XPath (XML Path Language);
- XSL-FO (XSL Formatting Objects).

As HTML has its own Cascading Style Sheets, there was a need to develop style sheets for XML and so it was created the XSL.

XSLT is used to transform a XML document into another XML document, or another type of document that is recognized by a browser, like HTML and XHTML [18].

In order to navigate through the XML nodes and attributes XSLT uses a special language called XPath which is a query language that may be used to compute values from the content of a XML document [19].

2.4 Web development platforms

A dynamic website has four main components [35] which stand for:

- Operating System (OS);
- Webserver;
- Database Management System (DBMS);
- Scripting Language.

The two main web development platforms are LAMP and WISA:

- Linux / Apache / MySQL / Hypertext Processor (PHP);
- Windows/Internet Information System (IIS) /SQL Server/ Active Server Pages .NET (ASP.NET).

LAMP is known for being a cheap solution and having an open-source community. On the other hand WISA is expensive, corporate and well documented. The differences between both platforms are shown in Table 2.

Platform	OS	Webserver	DBMS	Scripting Language
LAMP	Linux	Apache	MySQL	PHP
WISA	Windows	IIS	SQL Server	ASP.NET

Table 2 - Differences between LAMP and WISA

Since the available server has Windows and IIS installed the choice was the WISA platform.

2.5 WCMS

Navin Nagiah, CEO of DotNetNuke Corporation states that [55]:

“Linguistically, it means any system that helps you manage content – creating, storing, indexing, archiving, publishing, and distributing content”

WCMS is a system that permits the creation and management of web portals in a quick and easy way. Moreover they allow any Internet user, even if non-technical users, to create its own contents and share them. Even grandma would be able to share her most delicious recipes using a WCMS.

"WCM is about managing the publishing and distribution of information" [55].

With a WCMS the user only has to worry about managing the content of the portal itself and not the technical issues that such system arises such as security, consistency, redundancy and availability.

ASP.NET has plenty of web content management systems available. ASP.NET is a server-side scripting language that is part of the Microsoft .NET framework. These WCMS normally have a free and a professional paid version.

2.5.1 Orchard Project

The Orchard team is primarily composed of ASP.NET developers. It is an open-source, community-based project still in an early stage of development. It aims delivering applications and reusable components on the ASP.NET platform. It has 3 main objectives [42]:

- Create .NET applications for end-users, scripters and developers;
- Reuse of components;
- A community that helps defining applications and extensions.

Orchard MS is still not enough stable and widely used to be an option although if it continues to achieve its goals may become a good option in the future [37].

2.5.2 Kentico

Petr Palas created Kentico Software in June 2004. It is a WCMS used for building websites, online stores, intranets and web community sites. It has a browser-based interface for content editing, built-in modules and a customizable API. The administration section is split in two parts:

- CMS Site Manager (Site configurations and development)
- CMS Desk (Site content and data updates)

Kentico offers many different licences but the free versions are very limited compared to the commercial licences. Another disadvantage is its small community when compared to other WCMS. "I've tried a number of CMS systems over the last few years, and I don't think any of them offer this level of sophistication and features for the money." [35]. Kentico CMS is and ideal solution for enterprise websites [39].

2.5.3 Umbraco

A Danish called Niels Hartvig created Umbraco. In the beginning this project was being used by Hartvig to help him building web solutions for his clients [41].

It is designed for being used by developers, which will take a little more effort to get it running. It also takes some time to be learned and understood but after that the possibilities are endless [38]. The main advantages of this WCMS are:

- Very customizable
- Unique tools that will assist the developer solution from staging to live environment
- Training and support

It has two available versions: the free and the professional but both come with the same core product [40].

2.5.4 DotNetNuke

In 2002 Shaun Walker modifies the IBuySpyPortal kit [44] issued by Microsoft that demonstrates how ASP.NET along with Microsoft .NET framework can be used to build intranet and Internet portal applications. Furthermore in 2003 this modified version of the kit becomes the open-source WCMS called DotNetNuke.

DNN is an ASP.NET WCMS that allows normal users to create and edit a portal's content. It is a powerful tool for creating dynamic and interactive web pages using CSS, JavaScript and HTML.

There are two different DNN versions, the commercial and non-commercial one. A portal created with the DNN platform is very extensible as it allows users to add new functionalities in real time through the installation of new modules.

For the design of the website there are skins that allow users that don't have HTML, CSS or Javascript knowledge to make their portals more appealing and user friendly. At the same time it also allows web-designers to create their own skins, even if they don't use Visual Studio as a development tool because skins are written in HTML, CSS and Javascript and can be created using any non-Microsoft software.

2.6 DotNetNuke VS Kentico VS Umbraco

In order to chose the best WCMS for this work these three solutions were compared and reviewed. In Table 3 there are the criteria chosen that will allow knowing which WCMS suits best the system's requirements and user needs.

	DotNetNuke	Kentico	Umbraco
Free version	Lacks some advanced features	Very hard to find on the website and quite limited leaving Kentico logos on websites	Completely open-source
Templates	Has some free skins but has many commercial skins	Has very few commercial templates available	Only a few

	sold in numerous websites		
Standards Compliance	Core is XHTML 1.0 compliant but some modules are still not full compliant	All code rendered is XHTML 1.0 compliant	Completely web standards compliant
Module Availability	It comes with some free modules. Has more than 8000 3rd party add-ons at the snowcovered.com marketplace.	A lot of built-in modules and more resources available at the Kentico's DevNet site	Does not come with the same ready to use modules as other platforms. Contains open-source plugins and advices at our.umbraco.org
Community Support	More than 800.000 registered members many of which are active. Has a very solid commercial community regarding modules and skins	General community still growing and maturing as a whole. Already has a stable commercial community but it still does not possesses the widespread support as DNN or Umbraco	Community still growing rapidly and the commercial community support is on the rise. Already has the centralized community support it needed.
Ease of use	It is designed to be user friendly. Non-technical users can easily understand its User Interface (UI) and modules.	Has a novice-friendly practical interface	It is designed for developers and designers. Users must have some knowledge of .NET, HTML and CSS.

Table 3 - Comparison between different WCMS platforms

From the analysis of these three WCMS platforms it was decided that DotNetNuke would be the best choice for this work since it has been a leader in the ASP.NET WCMS for years. Moreover it proves that it is stable and mature enough to be used. It is also the most downloaded CMS from Codeplex [46] and it also offers a great variety of modules and skins on its online store.

For the purpose of this work it was chose the non-commercial version called "DNN Community Edition" since it is free and open-source, it has the most developed community and it is the WCMS that has more templates.

2.7 DotNetNuke Architecture

DNN uses dynamic page generation to render the correct information needed for each page. It uses the page URL to load the appropriate information [25]. Navigating in a traditional web application implies that an user is taken from a physical page to another. In DNN however there is only a physical page used in the application. Since it holds the information for each page in the database.

When a page is requested the URL is converted into a tabID that will be used to load the needed resources from the database.

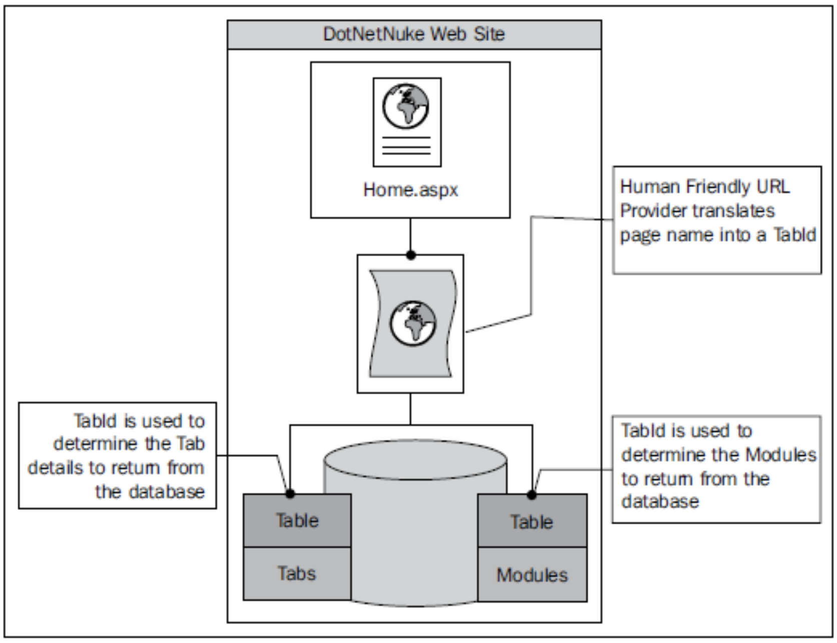

DotNetNuke Web Site
Home.aspx
Human Friendly URL Provider translates page name into a TabId
TabId is used to determine the Tab details to return from the database
TabId is used to determine the Modules to return from the database
Table
Tabs
Table
Modules

Figure 1- DNN dynamic page generation [25]

When an user requests a DNN page (Figure 2) its request sends a Query String with the tabID of the requested page. When the request reaches the web server it will look for the page information in the database (Figure 3).

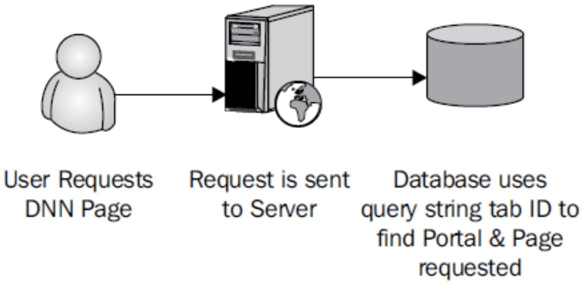

User Requests DNN Page
Request is sent to Server
Database uses query string tab ID to find Portal & Page requested

Figure 2 - User requesting a DNN page [26]

The database stores all the information need to display and create the page. The tabID sent in the request will be used to check which modules make part of the page, to which portal the page belongs to.

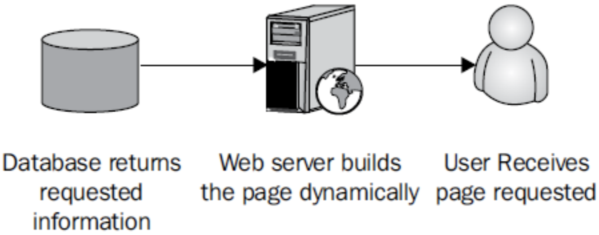

Database returns requested information
Web server builds the page dynamically
User Receives page requested

Figure 3 - DNN web server response [26]

DNN portals have a pre-defined hierarchy of organization elements (Figure 4). The organization elements are parent/child websites, pages and containers [26]. A portal always has a parent website that may contain one or more child websites. Each website has several web pages. These web pages can have any type of content depending on the wanted functionality and design. In older version of DNN the term “tab” was used when referring to web pages. A web page may contain several containers that are used to hold modules. These modules give functionalities to web pages.

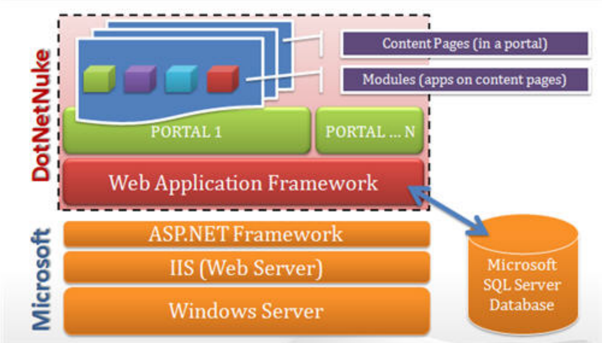

DotNetNuke
Content Pages (in a portal)
Modules (apps on content pages)
PORTAL 1
PORTAL ... N
Web Application Framework
Microsoft
ASP.NET Framework
IIS (Web Server)
Windows Server
Microsoft SQL Server Database

Figure 4 - DNN architecture [34]

DNN module architecture is divided in 3 layers and each layer as its own role in the application (Figure 5). The communication between the UI and the data always passes through the business logic layer. This provides a model for developers to create a flexible and reusable application. By breaking up an application into tiers, developers only have to modify or add a specific layer, rather than having to rewrite the entire application all over again [32]

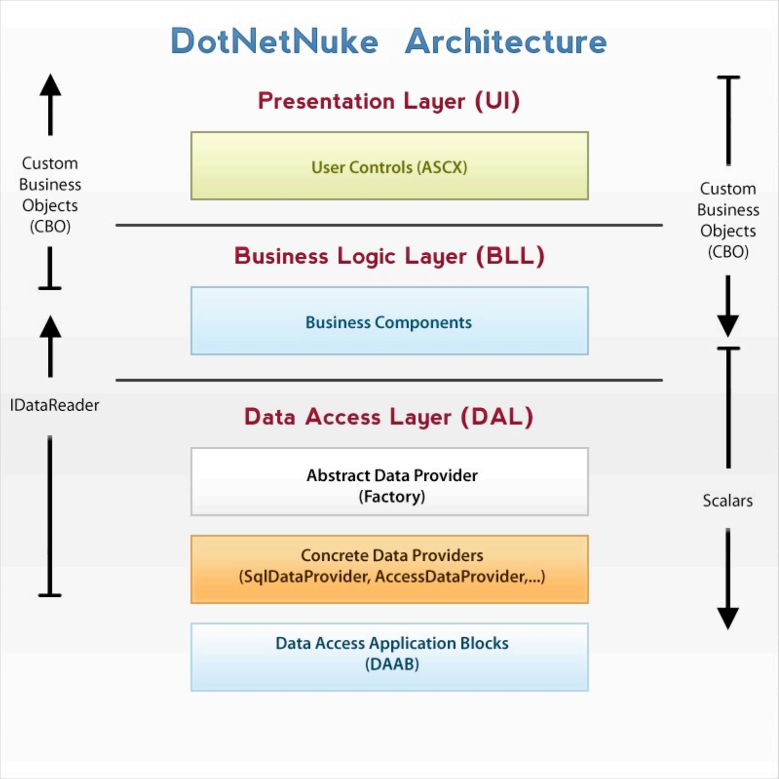

DotNetNuke Architecture
Presentation Layer (UI)
User Controls (ASCX)
Business Logic Layer (BLL)
Business Components
Data Access Layer (DAL)
Abstract Data Provider (Factory)
Concrete Data Providers (SqlDataProvider, AccessDataProvider,...)
Data Access Application Blocks (DAAB)
Custom Business Objects (CBO)
IDataReader
Custom Business Objects (CBO)
Scalars

Figure 5 - DNN 3-tier module architecture [21]

2.7.1 Presentation Layer

The Presentation Layer will display a graphical user interface (GUI) to the user that accesses the portal. It allows the user to interact with the portal and to request its services and contents. All the operations available in the portal will be done through the Presentation Layer that will communicate the user actions to the lower layers. The GUI is built with the following components [4]:

- Web forms
- Skins
- Panes
- Containers
- Module user controls
- Client-side scripts.

The *.ascx files are a collection of user controls that form the GUI of the portal/module.

2.7.2 Business Logic Layer

This layer will connect the upper layer (Presentation Layer) to the lower layer (Data Access Layer (DAL). All the logic and operations available to the user will be using this layer. It will make requests to the database through the DAL and these changes are reflected in the upper layer so that the user notices the portal's current state and what changes were actually made. It also provides several services to the user such as inserting new data, editing data and removing existing data.

All the database tables retrieved from the database will be mapped into a corresponding class. This will allow an object-oriented manipulation of the data.

The Business Components (Figure 5) are all the classes that have the business logic and methods to access and change the database (through the Data Access Layer).

2.7.3 Data Access Layer

The database is accessed through the Data Access Layer. The Business Logic Layer passes all the operations to be performed in the database to the Data Access Layer that will do the required access to the database itself. This layer will also pass any available data to the business layer so it can be displayed to the user via Presentation Layer.

The Data Provider allows developers to create their own implementations to connect to any Database vendor. DNN provides a Microsoft SQL Server implementation of the Data Provider API and this will be used in the implementation of this work.

The Data Access Application Blocks (DAAB) allows the developer to access the database using a helper class to avoid that the code to connect to the database or to execute any operation in the database is repeated over and over again. The DAAB makes the code cleaner and easier to understand.

2.8 Memória de África e do Oriente

"Memória de África e do Oriente" portal is developed by the University of Aveiro and "Centro de Estudos sobre África e do Desenvolvimento" (CEsA). It consists on the gathering of relevant data about the relationship between Africa, Lusophony countries and Portugal [8].

It includes a Virtual Library (which allows the users to check for the physical location of the bibliographic record) and a Digital Library. It also stores relevant data about the past relationship between Africa, Lusophony countries and Portugal [45].

The project main objectives are to avoid the oblivion of Portuguese past experiences in Africa as well as to preserve and centralize all the available historical documents. This will allow these historical documents to be searched by any user through the Internet.

The scan of historical documents is a very delicate issue since its copyrights only expire 70 years after the author's death and that raises many legal issues if a digital library has a non-authorized copy of the material.

This project was developed with the help of IT and bibliographic experts.

2.9 Sistema Integrado para Bibliotecas e Arquivos Digitais (SInBAD)

SInBAD is an integrated system for the digital library and digital archive of University of Aveiro and it is working since 2005. It is a web application with the purpose of allowing the storage, cataloguing, searching and dissemination of academic assets [47].

This IS supports the aggregation of multimedia documents. These documents go from copyrighted materials (books, papers or thesis) to images (posters and photographs) or videos.

Since it holds highly heterogeneous multimedia documents different metadata structures were need. Moreover this diversity made the system to be split in 3 subsystems:

- Libray;
- Archive;
- Jazz.

Each one of these 3 subsystems will store similar types of documents. The Library subsystem stores books, dissertations, scientific papers and posters. The Archive subsystem stores photographs and videos. Finally the Jazz subsystems stores books, magazines and music related to Jazz.

On top of all subsystems is the SInBAD portal. This way each subsystem can be used independently of all the others and if the user wants to perform a general search, the portal will query all the existent subsystems for results.

The system architecture is very modular and web service oriented so new subsystems can be created and easily added to the system. Whenever a new subsystem is added it becomes immediately available to be searched in the portal.

2.10 Authority Index

The objective of an authority index is to solve the problem related to the association of authors with their scientific papers. Usually in the scientific community some members usually use a name as their “scientific name”. The representation of the author’s name sometimes depends on the format defined by the scientific magazines/conferences. For example, a scientific name is “António Pinho” and in a conference X they require that the author’s name in the article to be in the format “António M. R. Pinho”. Not having an authority implies that when a search for articles of “António Pinho” is performed, the article presented in conference X won’t appear in the results.

These formats may not allow the author of a work to be identified by its “scientific name” and therefore not letting that author to be associated with that work.

The introduction of an authority index will avoid this since it is maintained the accuracy of the work's source. It will also allow granting credits to the real author of the work. [12]

2.11 UNIMARC

UNIMARC defines the rules for the representation, storage and cataloguing of bibliographic records. The actual structure of UNIMARC fields/subfields was mapped into a relational database. This structure will be described next. Each bibliographic record has the following structure:

Record Label	Directory	Data Fields	Record Terminator

Table 4 - UNIMARC Bibliographic Record Scheme

2.11.1 Record Label

The record label consists on 24 characters that will be used to identify the record itself. Since all the records were stored in a relational database this record label was not used. Instead it was used a unique Primary Key (PK) that will identify each bibliographic record. This will allow decreasing the amount of information stored in the database.

2.11.2 Directory

The directory consists on a set of entries that represent a field. These entries have the field's tag, its length (number of characters) and its starting position. Since this information can be taken from the database when need it won't be stored in any field. When required it will be retrieved from the database.

2.11.3 Data Fields

The bibliographic information is stored in data fields. Each data field stores specific information depending on its tag. Each field has a tag, 2 indicators and 1 or more subfields.

The tag is a number that gives information about what it is stored in that field and its possible indicators and subfields.

The 2 indicators (Indicator 1 and Indicator 2) supply additional information about the field's contents.

Finally the subfields have an identifier ($x) and holding the bibliographic information as their value. The subfields can be optional or mandatory. For this work it was only used the fields/subfields need to describe both articles and periodical publications.

2.11.4 Fields Used

In the appendix there is a description of all the UNIMARC fields that had to be used in order to catalogue the bibliographic records.

Chapter 3 - **Architecture**

In the following chapter it will be described the portal's architecture (Figure 6) as well as the module's architecture.

3.1 General Architecture

The bibliographic records are the most important piece of the architecture since all the other elements are related or connected to it. When a bibliographic record is created it is catalogued with UNIMARC fields and in order to enrich its content a set of relations can be established. A bibliographic record can have two different types of relations.

The first type is a relation between a bibliographic record and another bibliographic record.

The other type of relation is between an authority and the bibliographic record. This last relation represents the role that the authority has in a bibliographic record and is done through the use of an authority index.

Authenticated users can interact with catalogued bibliographic records by leaving their comments. The possibility of allowing comments to be done by users to a bibliographic record opens a wide number of new possibilities. Therefore it allows the enrichment of the bibliographic record as it provides information from another who is not the same person, which created that bibliographic record. This will lead to collective memory as a group of people is sharing and creating comments based on their own personal experiences and knowledge. These comments can be done in the form of text or key words (tags).

In order to the comments to be shown in the portal, an administrator must validate them so that any inappropriate or offensive content is refused.

Together with the bibliographic record there is the resource in its digital form stored in a document. For example associated with the bibliographic record of a book there can be a digital copy of it. This way the user has access not only to the bibliographic record but also to the actual book enhancing its experience with the system. If a user is doing research in a certain book it may not be sufficient to only have its bibliographic record available.

All of this together will allow a dynamic generation of information that goes beyond the original content of a standalone bibliographic. This result will be outputted to the user.

In order to support the proposed architecture, given the start conditions the setup chosen to perform it will be using an IIS server, a SQL Server database and DNN as the WCMS.

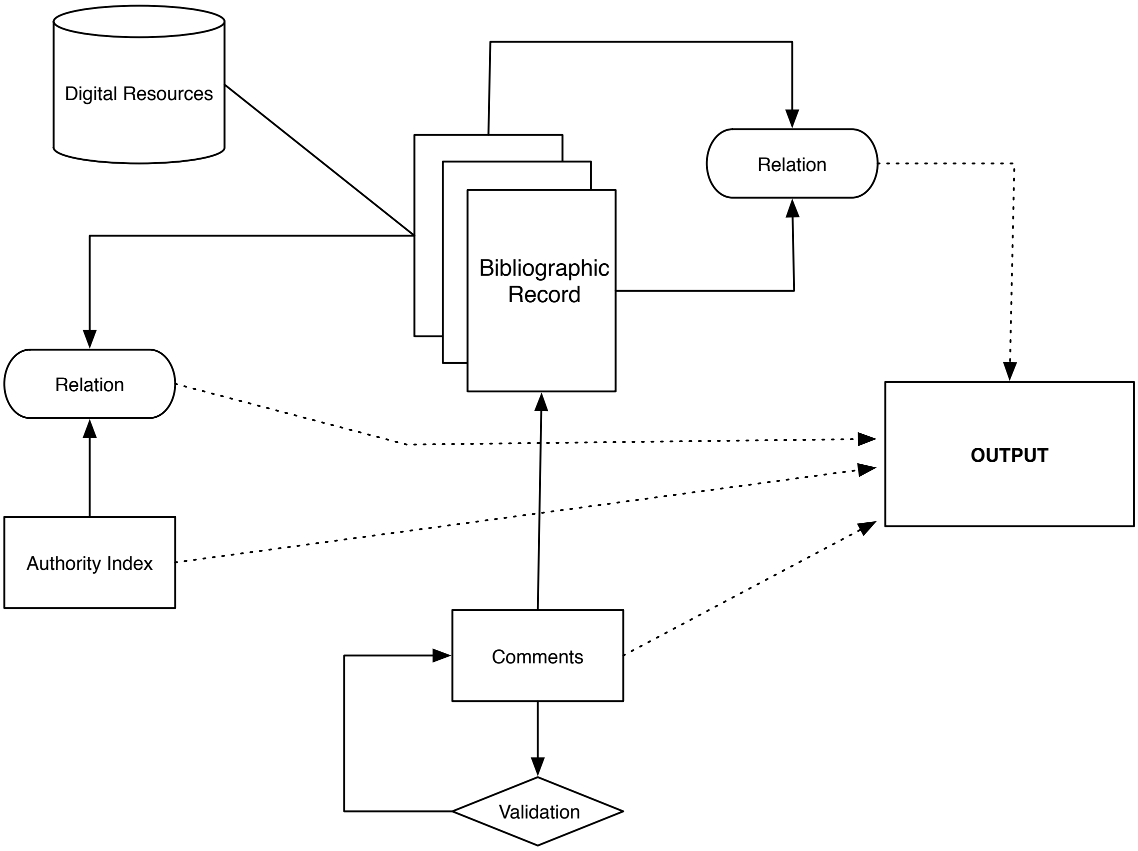


Figure 6 - System General Architecture

The system has 3 different types of users:

- Authenticated users with administration privileges.
- Authenticated users;
- Non-Authenticated users;

Starting with the non-authenticated users they may search and visualize all the information regarding a certain bibliographic record. This information includes:

- Authority index;
- Cataloguing information;
- Comments performed by other users though their username is not shown;
- Digital document;
- Relations regarding that bibliographic record.

The authenticated users have access to the same contents as non-authenticated users but they may also create comments about a certain bibliographic record.

The last user is the most important type of user since it manages all the back office of the portal allowing it to manage all the portals’ contents:

- Authority index;

- Bibliographic records and their digital resources;
- Registered users.
- Relations;
- User comments and their respective validation;

3.2 Module's Architecture

For creating the "Mensageiros do Jazz" functionalities it was developed a DNN module. This module follows a 3-tier architecture. Each one of the 3 layers will be described in detail in each of its sub-chapters. In Figure 7 it is displayed the module architecture scheme.

The Presentation Layer uses *.ascx controls to display the graphical user interface to the user. All the data need to display the required information will have to be requested to the Business Layer controller class (MensaJazzFormController).

The Business Logic Layer is used to hold the data retrieved from the database and contains all the logic need to perform the requested operations.

The last layer is the Data Access Layer that will be using a data provider class to access the database.

The diagrams used to describe the system details are from Unified Modelling Language (UML).

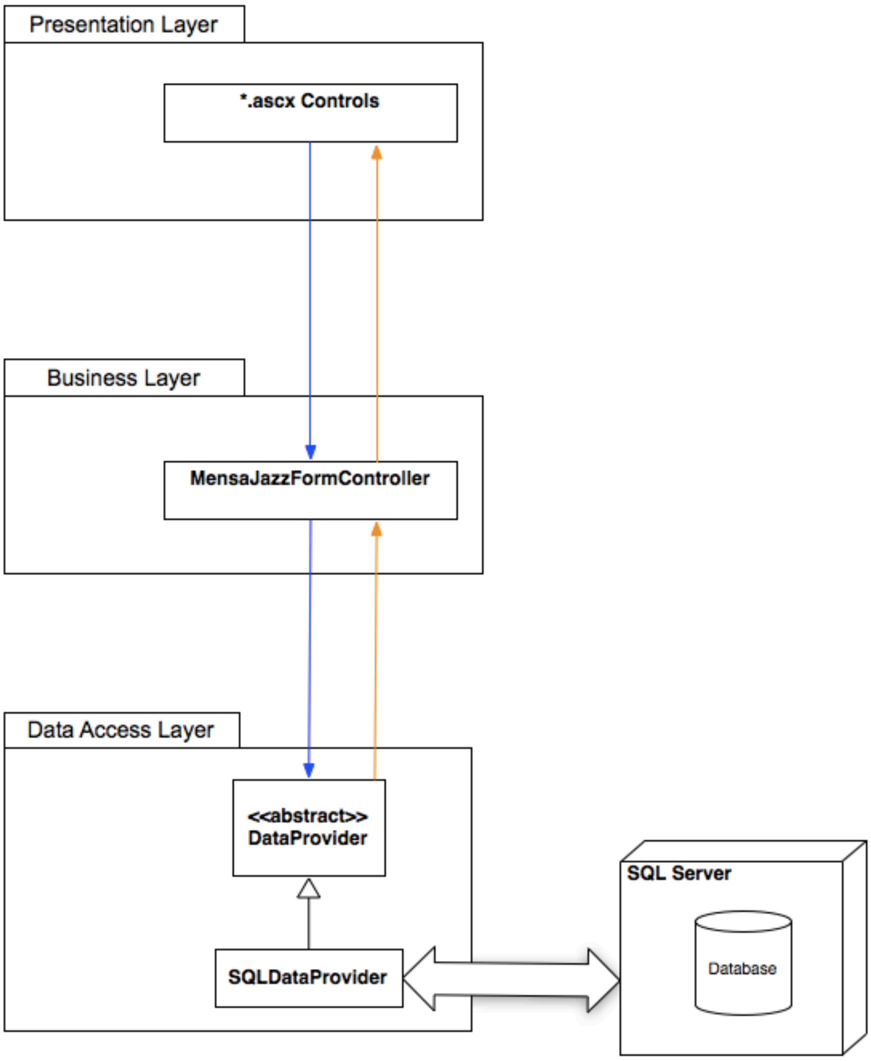


Figure 7 – Module Package Diagram

3.2.1 Data Access Layer

Microsoft SQL Server is a Database Management System (DBMS) developed by Microsoft. It uses Transact-SQL (T-SQL), an extension of the Structured Query Language (SQL), for querying, altering and defining relational databases with statements [20]. The database used in this work was developed with the SQL Server 2008

The first logical part of the database consists on the tables that together form the UNIMARC structure and will be storing the bibliographic information.

The second logical part has the tables that will hold the information on the relations between two distinct records and the relations between an author and a bibliographic record.

The third and last logical part has the tables that will store the comments made by users to specific bibliographic records.

As described previously in chapter 2.11 UNIMARC is a collection of fields. Every field has a tag, two indicators, a mandatory/repeatable flag and one or more subfields.

Field			Subfields	
Tag	**Indicator 1**	**Indicator 2**	**Code**	**Value**
200	1	#	$a	Book of lost things
606	0	#	$a	Mystery
606	0	#	$a	Fiction

Table 5 - UNIMARC Field example

Each subfield has a code and a value. Each field supports a set of possible subfields. In order to avoid having invalid fields and subfields entered in the database it was created the Fields table, which will be filled with all the existent fields, and the SubFields table that will be filled with all possible subfields for each different field.

Every bibliographic record is uniquely identified with he primary key from the Record table (Figure 8). A bibliographic record has one of 2 possible types (article or periodical publication). These types are stored in the RecordType table.

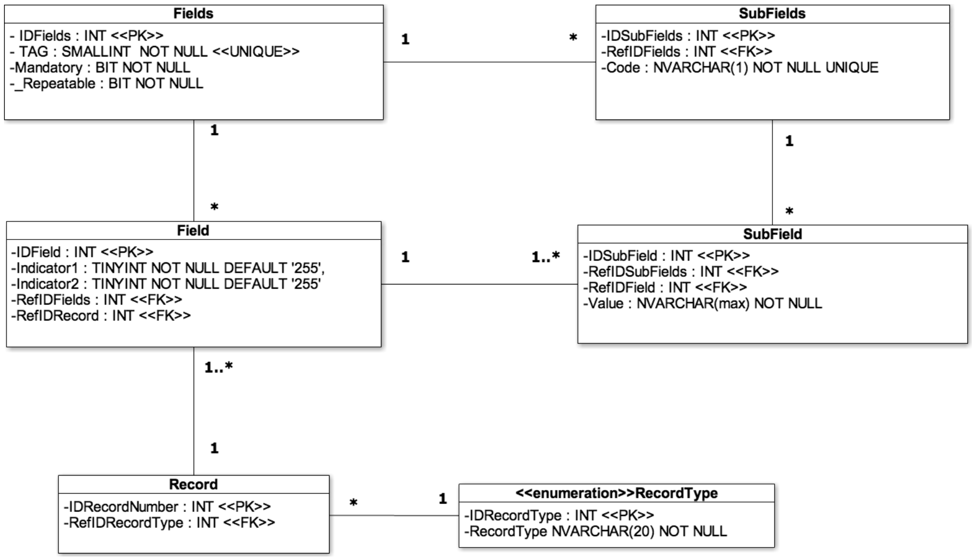

Fields
- IDFields : INT <<PK>>
- TAG : SMALLINT NOT NULL <<UNIQUE>>
-Mandatory : BIT NOT NULL
-_Repeatable : BIT NOT NULL
1
*
SubFields
-IDSubFields : INT <<PK>>
-RefIDFields : INT <<FK>>
-Code : NVARCHAR(1) NOT NULL UNIQUE
1
*
1
*
Field
-IDField : INT <<PK>>
-Indicator1 : TINYINT NOT NULL DEFAULT '255',
-Indicator2 : TINYINT NOT NULL DEFAULT '255'
-RefIDFields : INT <<FK>>
-RefIDRecord : INT <<FK>>
1
1..*
SubField
-IDSubField : INT <<PK>>
-RefIDSubFields : INT <<FK>>
-RefIDField : INT <<FK>>
-Value : NVARCHAR(max) NOT NULL
1..*
1
Record
-IDRecordNumber : INT <<PK>>
-RefIDRecordType : INT <<FK>>
*
1
<<enumeration>>RecordType
-IDRecordType : INT <<PK>>
-RecordType NVARCHAR(20) NOT NULL

Figure 8 - UNIMARC database tables (Part I)

As stated before, UNIMARC has a great number of fields and subfields. In order to help the user there are descriptions for all the fields/subfields. These descriptions are available in both Portuguese and English and are stored in the Fields_PT and Fields_EN tables respectively. The same happens with subfields but their descriptions are stored in the SubFields_PT and SubFields_EN. All those descriptions were taken from the UNIMARC standard [22][23].

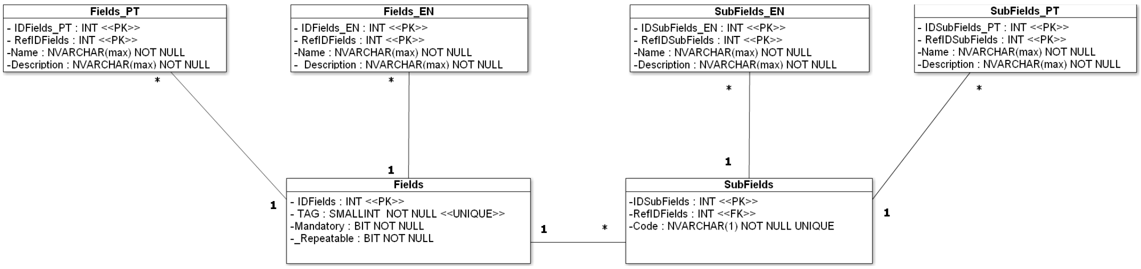

Fields_PT
- IDFields_PT : INT <<PK>>
- RefIDFields : INT <<PK>>
-Name : NVARCHAR(max) NOT NULL
-Description : NVARCHAR(max) NOT NULL
Fields_EN
- IDFields_EN : INT <<PK>>
- RefIDFields : INT <<PK>>
-Name : NVARCHAR(max) NOT NULL
- Description : NVARCHAR(max) NOT NULL
SubFields_EN
- IDSubFields_EN : INT <<PK>>
- RefIDSubFields : INT <<PK>>
-Name : NVARCHAR(max) NOT NULL
-Description : NVARCHAR(max) NOT NULL
SubFields_PT
- IDSubFields_PT : INT <<PK>>
- RefIDSubFields : INT <<PK>>
-Name : NVARCHAR(max) NOT NULL
-Description : NVARCHAR(max) NOT NULL
*
*
*
*
1
1
1
1
Fields
- IDFields : INT <<PK>>
- TAG : SMALLINT NOT NULL <<UNIQUE>>
-Mandatory : BIT NOT NULL
-_Repeatable : BIT NOT NULL
1
*
SubFields
-IDSubFields : INT <<PK>>
-RefIDFields : INT <<FK>>
-Code : NVARCHAR(1) NOT NULL UNIQUE

Figure 9 - UNIMARC Database Tables (Part II)

In order to develop an Authority Index it was created a Person table that has personal information about each authority member and it was also created a table that represents a relation between the bibliographic records and the authority members. For each type of authority it was created a separate table that is linked to the Person's table:

- Author_Colectivity represents a community:
- Author_Family represents a family;
- Author_Person represents an individual author.

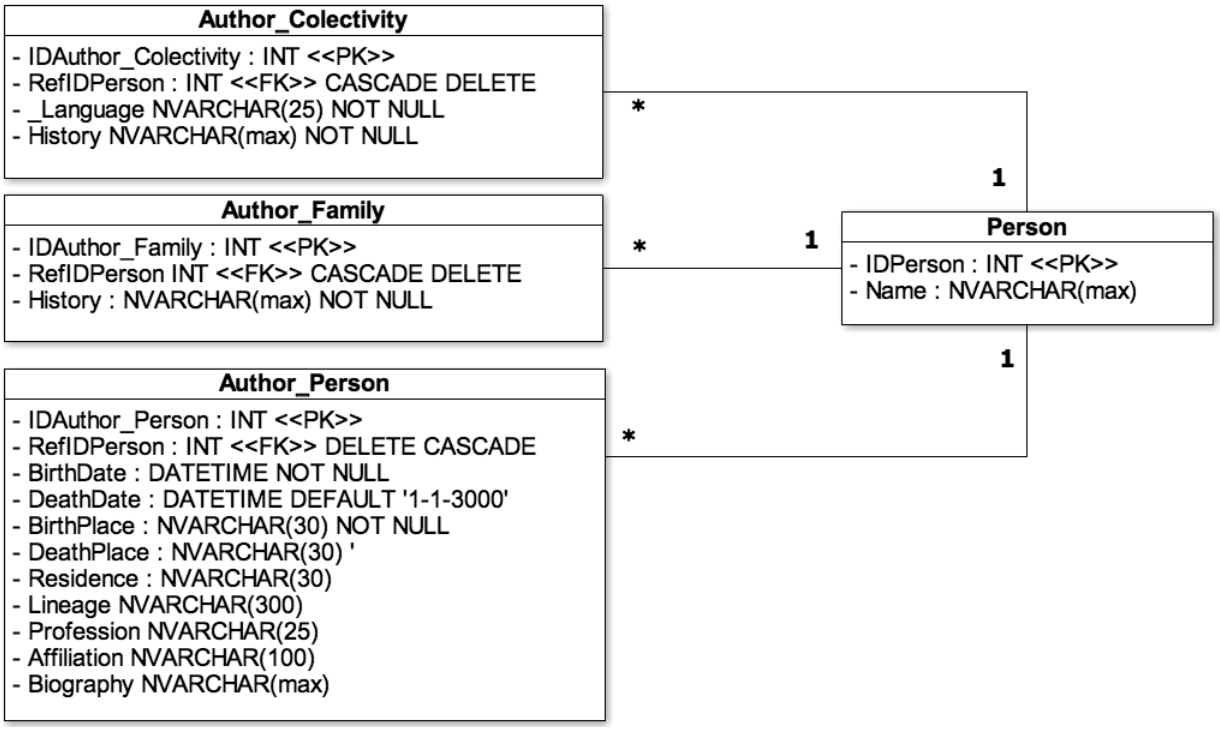

Author_Colectivity
- IDAuthor_Colectivity : INT <<PK>>
- RefIDPerson : INT <<FK>> CASCADE DELETE
- _Language NVARCHAR(25) NOT NULL
- History NVARCHAR(max) NOT NULL
*
1
Author_Family
- IDAuthor_Family : INT <<PK>>
- RefIDPerson INT <<FK>> CASCADE DELETE
- History : NVARCHAR(max) NOT NULL
*
1
Person
- IDPerson : INT <<PK>>
- Name : NVARCHAR(max)
1
Author_Person
- IDAuthor_Person : INT <<PK>>
- RefIDPerson : INT <<FK>> DELETE CASCADE
- BirthDate : DATETIME NOT NULL
- DeathDate : DATETIME DEFAULT '1-1-3000'
- BirthPlace : NVARCHAR(30) NOT NULL
- DeathPlace : NVARCHAR(30) '
- Residence : NVARCHAR(30)
- Lineage NVARCHAR(300)
- Profession NVARCHAR(25)
- Affiliation NVARCHAR(100)
- Biography NVARCHAR(max)
*

Figure 10 – Author's Database Tables

As referred there are two types of relations:

- A relation between two distinct bibliographic records;
- A relation between an authority member and a bibliographic record.

For the relation between bibliographic records the possible names are stored are in the Type_Record_Record table and shown in Table 6.

Relation names			
Version	Resume	Transformation	Complement
Edition	Critic/comment	Imitation	All/part
Translation	Adaptation/modification	Continuation	Part/all
Summary	Improvisation	Supplement	

Table 6 - Possible relation names between distinct bibliographic records

The possible relations between an authority member and a bibliographic record are stored in the Type_Author_Role table. These roles consist on the list from the UNIMARC manual Appendix C [11].

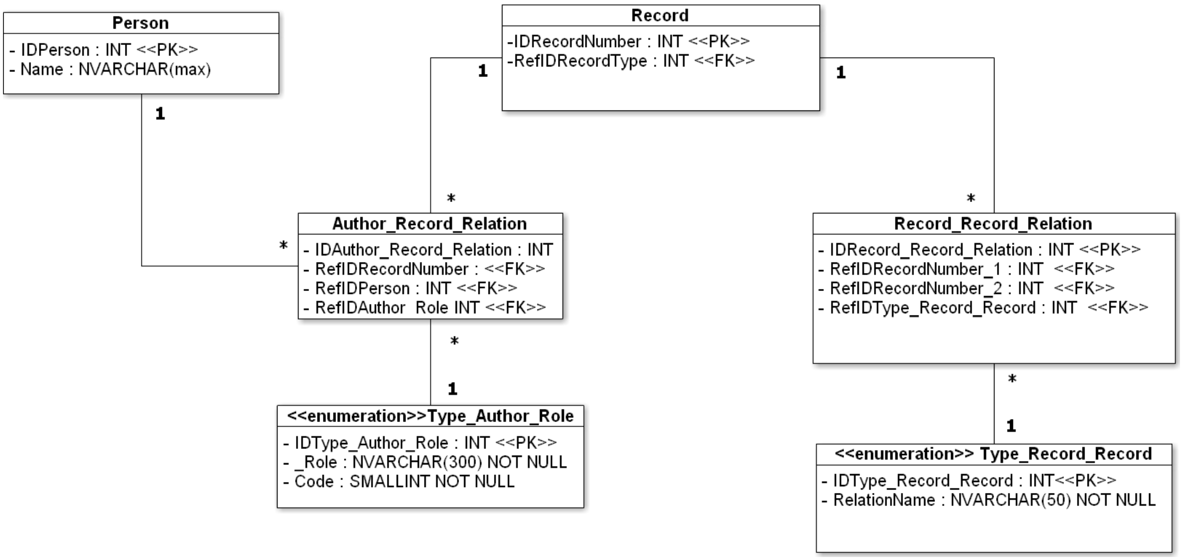

Person
- IDPerson : INT <<PK>>
- Name : NVARCHAR(max)
Record
-IDRecordNumber : INT <<PK>>
-RefIDRecordType : INT <<FK>>
Author_Record_Relation
- IDAuthor_Record_Relation : INT
- RefIDRecordNumber : <<FK>>
- RefIDPerson : INT <<FK>>
- RefIDAuthor Role INT <<FK>>
Record_Record_Relation
- IDRecord_Record_Relation : INT <<PK>>
- RefIDRecordNumber_1 : INT <<FK>>
- RefIDRecordNumber_2 : INT <<FK>>
- RefIDType_Record_Record : INT <<FK>>
<<enumeration>>Type_Author_Role
- IDType_Author_Role : INT <<PK>>
- _Role : NVARCHAR(300) NOT NULL
- Code : SMALLINT NOT NULL
<<enumeration>> Type_Record_Record
- IDType_Record_Record : INT<<PK>>
- RelationName : NVARCHAR(50) NOT NULL
1
*
1
*
1
*
*
1
*
1

Figure 11 – Relation's Database Tables

For each existent bibliographic record, the registered users in the website can make comments about them. An administrator moderates each comment made by an user in order to validate them. Each comment can be in one of three possible states:

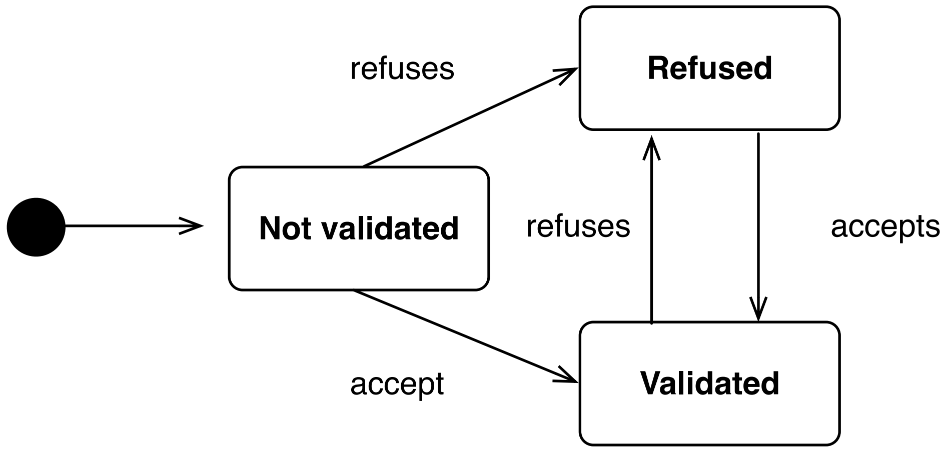


Figure 12 - Comments state diagram

Every comment starts in the “Not validated” state waiting for an administrator to evaluate its content and either validate or refuse it. If the administrator validates the comment its state is changed to “Validated” and it will be shown to users. On the hand if the administrator refuses the comment its state is changed to “Refused” and it won’t be displayed to users (Figure 12). If by any reason the administrator makes a mistake he can always change the state of the comment from Refused to Validated or vice-versa.

The available states are stored in the Type_CommentState. Only logged users may write comments. All users can view the comments but the information regarding the author of a specific comment is only shown to users that are logged in.

All the comments are stored in the Comment table, which is linked to the Record table, and Users tables. This way it is possible to know the content of a comment made to a bibliographic record as well as its user.

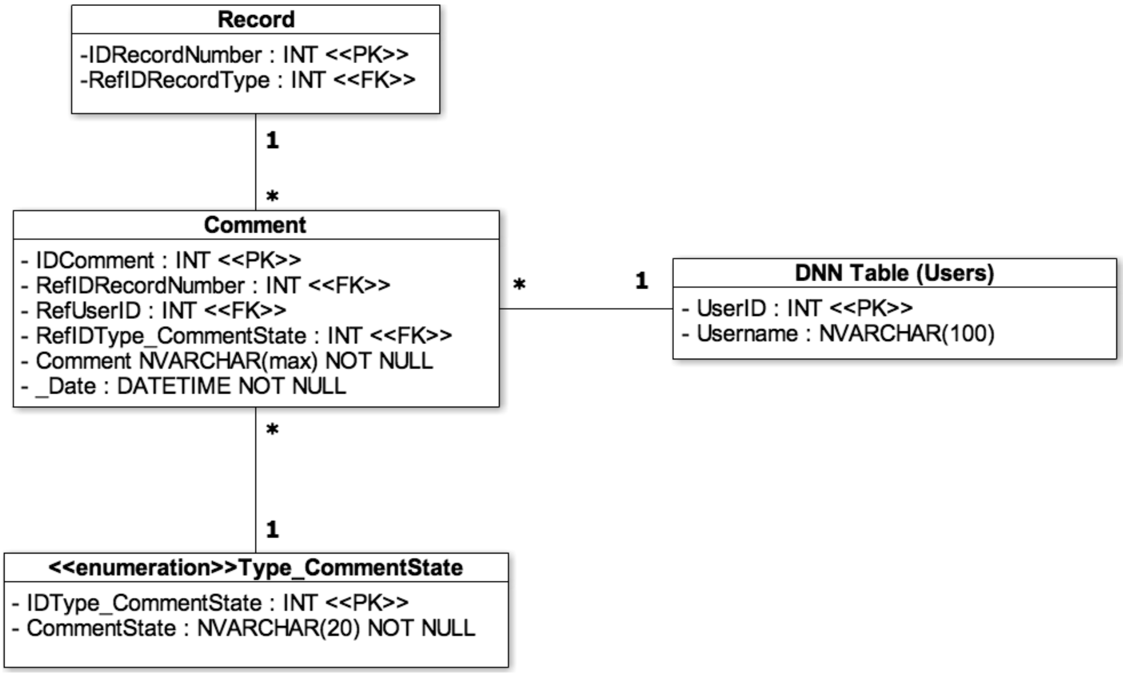

Record
-IDRecordNumber : INT <<PK>>
-RefIDRecordType : INT <<FK>>
1
*
Comment
- IDComment : INT <<PK>>
- RefIDRecordNumber : INT <<FK>>
- RefUserID : INT <<FK>>
- RefIDType_CommentState : INT <<FK>>
- Comment NVARCHAR(max) NOT NULL
- _Date : DATETIME NOT NULL
*
1
DNN Table (Users)
- UserID : INT <<PK>>
- Username : NVARCHAR(100)
*
1
<<enumeration>>Type_CommentState
- IDType_CommentState : INT <<PK>>
- CommentState : NVARCHAR(20) NOT NULL

Figure 13 - Comment Database tables

3.2.1.1 DAL Architecture

The classes that belong to the data access layer are placed in the “Data Access Layer” package. In this package these classes will access the database to retrieve data and returns it to the business layer. The “MensaJazzFormController” class from the Business Logic Layer will communicate with the DataProvider through the “Instance” factory method, which will return the concrete DataProvider class (SQLDataProvider).

The DataProvider is an abstract class from which the instantiated data provider will inherit. Depending on the database vendor used there will be a corresponding instantiated data provider class. In this work the chosen vendor was the SQL Server so the instantiated data provider is the SQLDataProvider class. This class will access the database whenever the controller class requests.

3.2.2 Business Logic Layer

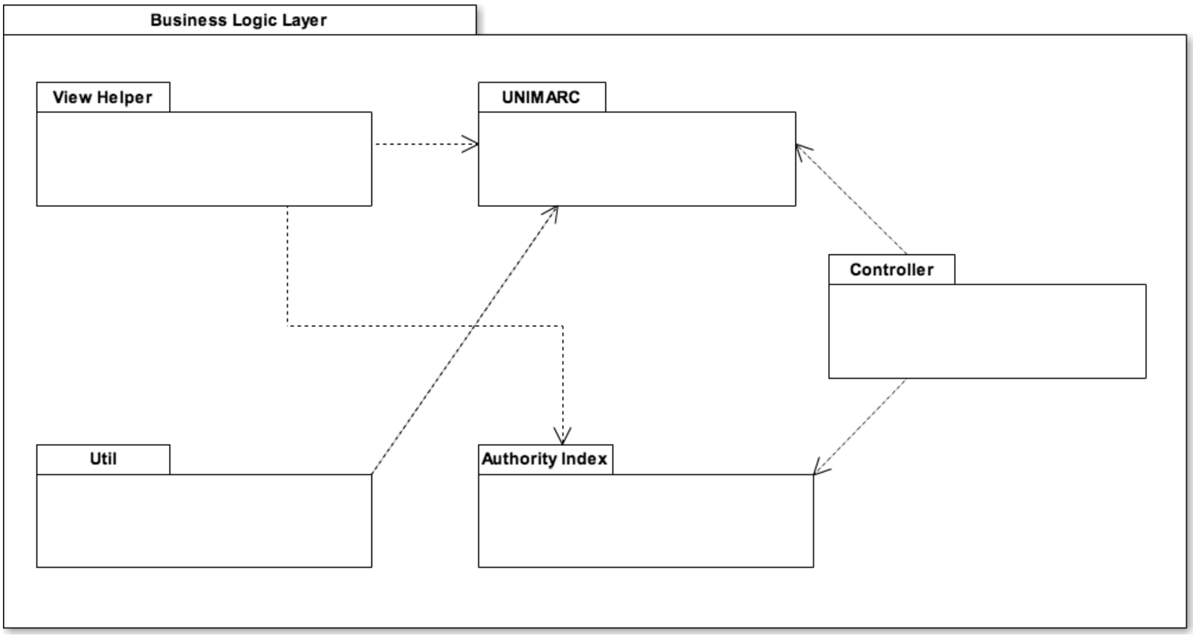


Figure 14 - Packages overview

The module development has lead to 5 main packages. The classes of each package were grouped accordingly to their purpose. The class in the "Controller" package will make the link between the Presentation Layer and the Data Access Layer.

All the package's contents and interaction will be presented in the following sections.

3.2.3 UNIMARC Package

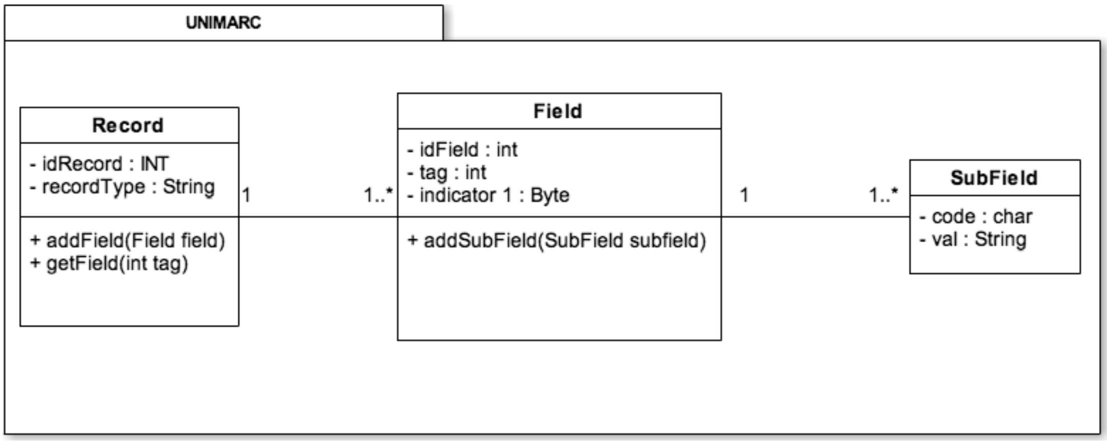


Figure 15 - UNIMARC Package

Therefore it includes 3 classes that together form an UNIMARC bibliographic record. These classes will be used to manipulate all the data retrieved from the database.

They will also be used to store the data entered from the input form. The structure of these classes is very similar to the corresponding tables in the database.

3.2.3.1.1 Record Class

This class represents a bibliographic record and it contains several UNIMARC fields that describe its contents.

Attribute Name	Description
idRecord	Used to store the primary key from the database of that record so that changes can be made to it
recordType	Tells if a record is a periodical publication or an article
fields	List of fields

Table 7 - Record class attributes

3.2.3.1.2 Field Class

This class represents an UNIMARC field. It can have one or more subfields associated with it.

Attribute Name	Description
idField	Used to store the primary key from the database of that field so that changes can be made to it
tag	Identifies which UNIMARC field it is
ind1	Field’s Indicator 1
Ind2	Field’s Indicator 2
subfields	List of subfields

Table 8 - Field class attributes

3.2.3.1.3 SubField Class

This class represents an UNIMARC subfield.

Attribute Name	Description
code	Identifies a subfield
val	Contains the subfield value

Table 9 - Subfield class attributes

3.2.3.2 Util package

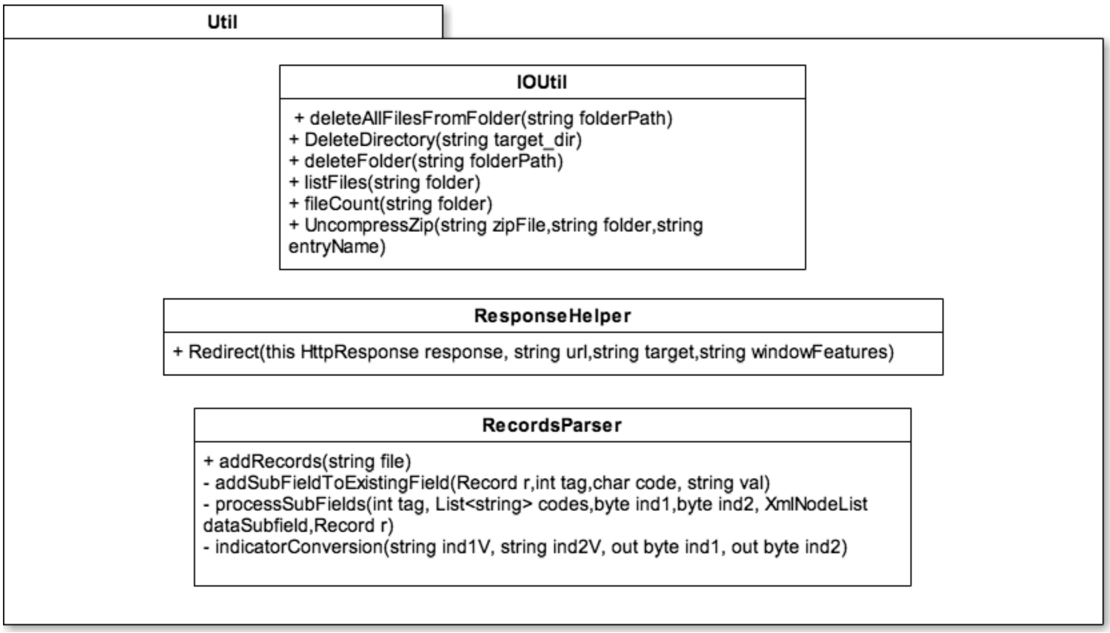

Util
IOUtil
+ deleteAllFilesFromFolder(string folderPath)
+ DeleteDirectory(string target_dir)
+ deleteFolder(string folderPath)
+ listFiles(string folder)
+ fileCount(string folder)
+ UncompressZip(string zipFile,string folder,string entryName)
ResponseHelper
+ Redirect(this HttpResponse response, string url,string target,string windowFeatures)
RecordsParser
+ addRecords(string file)
- addSubFieldToExistingField(Record r,int tag,char code, string val)
- processSubFields(int tag, List<string> codes,byte ind1,byte ind2, XmlNodeList dataSubfield,Record r)
- indicatorConversion(string ind1V, string ind2V, out byte ind1, out byte ind2)

Figure 16 - Util package

The Util package contains auxiliary classes that perform useful IO operations such as:

- Allowing a tab to be opened in a redirect;
- Insertion of bibliographic records saved in a XML file.
- Managing the uploaded files;

3.2.3.2.1 IOUtil Class

It is easier for users to upload several files by creating single .zip file and upload it to the system. For example imaging that a digital document has 1000 images. Instead of having the user uploading those images one by one he can create a zip file and upload it to the system. On receive of the zip file the system unzips the 1000 image files to a folder.

This utility class performs management operations on the file system concerning the files that are uploaded in the system. This class in association with the ICSharpCode.SharpZipLib[14] library will allow the uploaded zip files to the system to be uncompressed.

Method name	Description
deleteFolder(string folderPath)	Deletes the folder “folderPath”
listFiles(string folder)	Retrieves the list of all files within the folder
fileCount(string folder)	Counts the number of files within a folder
UncompressZip(string zipFile, string folder,	Unzips a .zip file “zipFile” to the folder

string entryName)	“folder” with the extracted files named “entryName”

Table 10 - IOUtil class methods

3.2.3.3 *ResponseHelper [13]*

This class is used to allow a Response.Redirect to accept more parameters in order to open a new window. It uses what is called “Extension Methods” which are new to C# 3.0 and allow the user to add new methods to existing types.

Method name	Description
Redirect(this HttpResponse response, string url,string target,string windowFeatures)	Allows the redirection to a new page. The argument windowFeatures allows the user to define the properties of the new window

Table 11 - ResponseHelper class methods

3.2.3.3.1 *RecordsParser*

This class will store UNIMARC records defined in XML files in the database.

Method name	Description
addRecords(string file)	Given a XML file path this method will parse its content and add the result to a List<Record>
addSubFieldToExistingField(Record r,int tag,char code, string val)	Adds a subfield to an existing field of a record
processSubFields(int tag, List<string> codes,byte ind1,byte ind2, XmlNodeList dataSubfield,Record r)	Creates a new field and adds the subfields with codes listed in the “List<string> codes” to it. The new created field is then added to the record.
indicatorConversion(string ind1V, string ind2V, out byte ind1, out byte ind2)	Converts the indicators read from the XML file to a valid format used by the system

Table 12 - RecordsParser class methods

3.2.3.4 *View Helper Package*

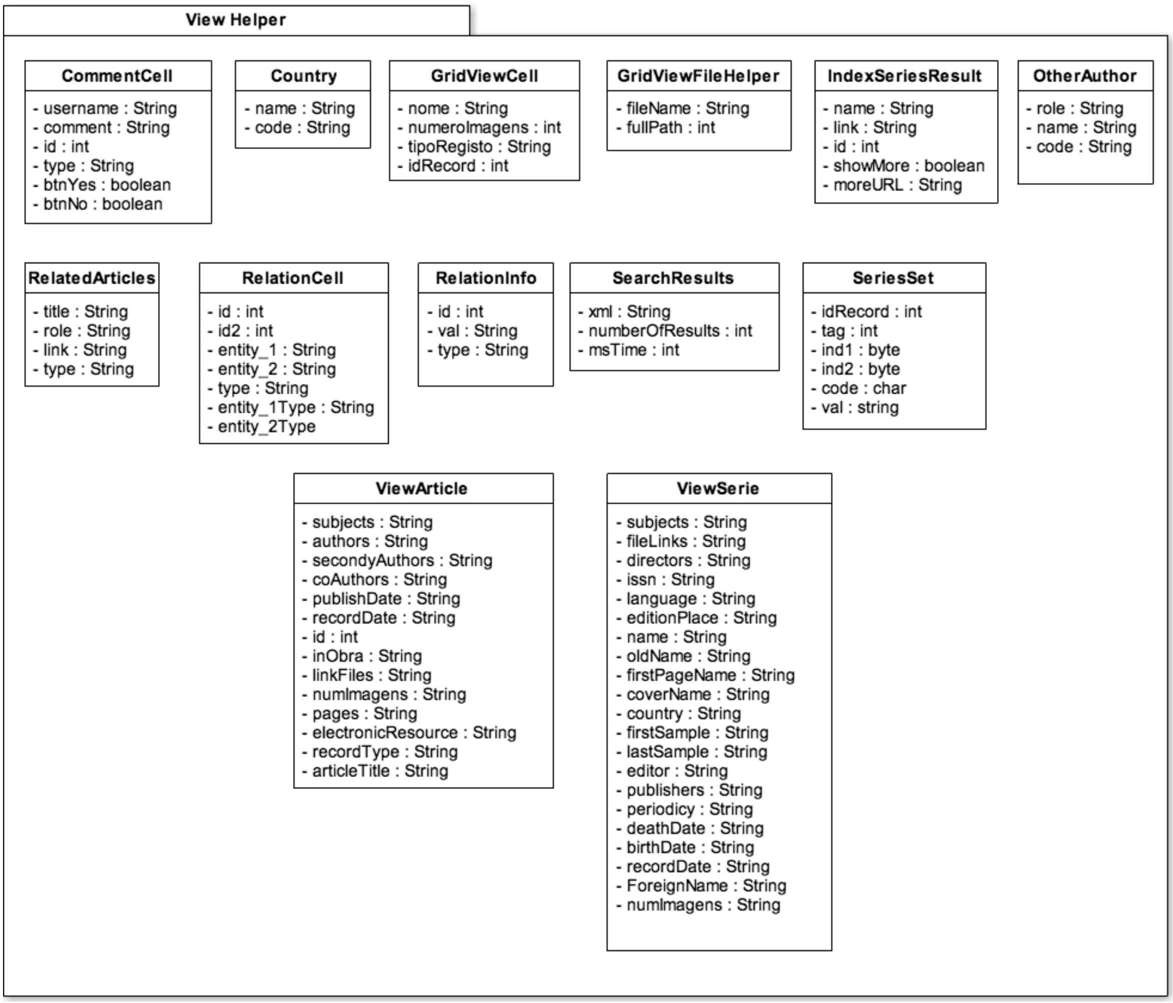


Figure 17 - View Helper

This package contains classes that are used in server-side controls like "GridView" and "ListView" to store and display retrieved data from the database to the user.

Class Name	Description
CommentCell	This class is used to list the comments of a record. The boolean attributes are used to show the accept/deny buttons depending on the comment's current state.
Country	This class is used to bind a dropdownlist with the list of all existent countries. This list is retrieved from the "Type_Countries"

	database table.
GridViewFileHelper	This class stores information about the location of the files stored in the system and belonging to a certain bibliographic record.
GridViewCell	This class stores the information from a record retrieved from the database. This information is then used as a GridView's data source to display the record in a readable way to the user.
IndexSeriesResult	This class will be used for the indexing of records. It displays the name of the bibliographic record and its search link.
OtherAuthor	This class will be used in the Article's form to store other types of authority like designers, illustrators along with their UNIMARC relator code.
RelatedArticles	This class is used to retrieve the related articles of an authority or member of a periodical publication.
RelationCell	This class is used to retrieve information about a relation. There are two types of relation: a relation between 2 records and a relation between a record and an authority member.
RelationInfo	This class retrieves a list of existent relations, or the list of available authority members or bibliographic records to be bound to a server side control.
SearchResults	This class will be used to store the search results in xml format. It also shows how much time it took (in ms) and the number of results that were found
SeriesSet	This class will be used to store the records fields so that they can be used as data sources of server-side controls.
ViewArticle	This class will be used to store information about an article. This information will be displayed when the user wants to see more information about a record (in this case of

	the type article).
ViewSerie	This class will be used to store information about periodical publication. This information will be displayed when the user wants to see more information about it

Table 13 - ViewHelper package classes' description

3.2.3.5 Authority Index Package

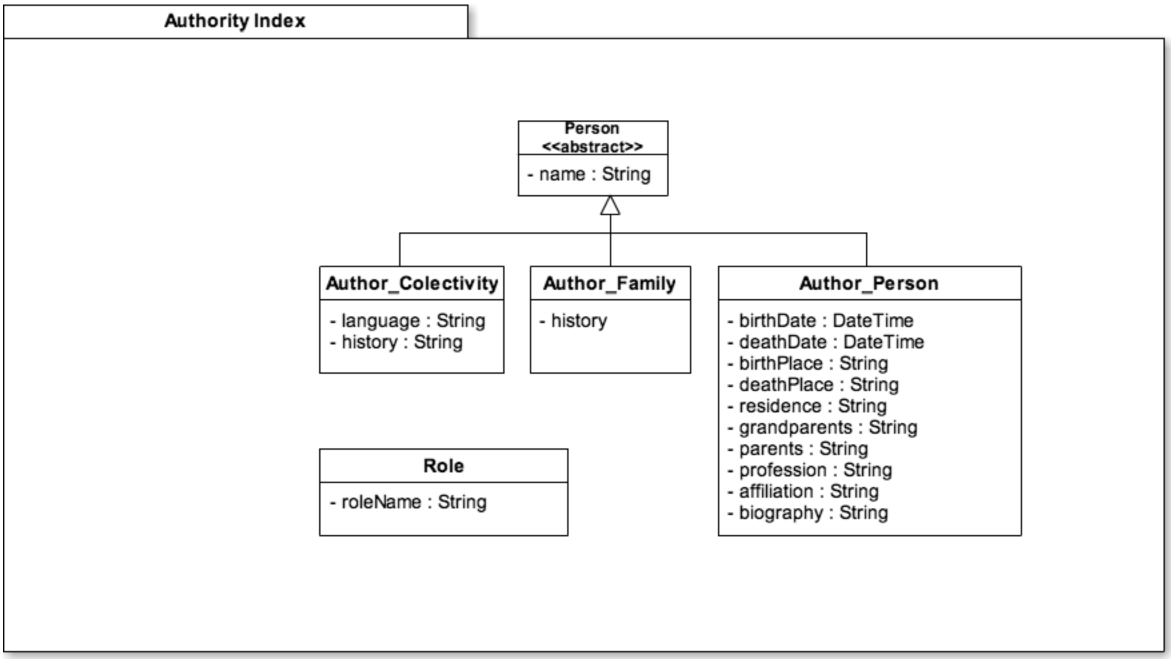


Figure 18 – Authority Index Package

This package contains the classes that represent the authority index. It also has a class that refers to all possible authority members' roles.

3.2.3.5.1 Person class

This abstract class represents a person

Attribute Name	Description
name	Name of the person

Table 14 - Person class attributes

3.2.3.5.2 Author_Colectivity class

This class represents a type of authority that symbolizes a colectivity.

Attribute Name	Description
language	Language of the colectivity
history	History of the colectivity

Table 15 - Author_Colectivity class attributes

3.2.3.5.3 Author_Family class

This class represents a type of authority that symbolizes a family.

Attribute Name	Description
history	History of the family

Table 16 - Author_Family class attributes

3.2.3.5.4 Author_Person class

This class represents a type of authority that symbolizes a person.

Attribute Name	Description
birthDate	Persons' date of birth
deathDate	Persons' date of death
birthPlace	Place where the person was born
deathPlace	Place where the person has passed away
residence	Persons' residence
grandParents	Persons' grandparents
parents	Persons' parents
profession	Persons' profession
affiliation	Persons' affiliation
biography	Persons' biography

Table 17 - Person class attributes

3.2.3.6 Comments Package

This package contains the classes that will be used to display all the information related to comments made to bibliographic records.

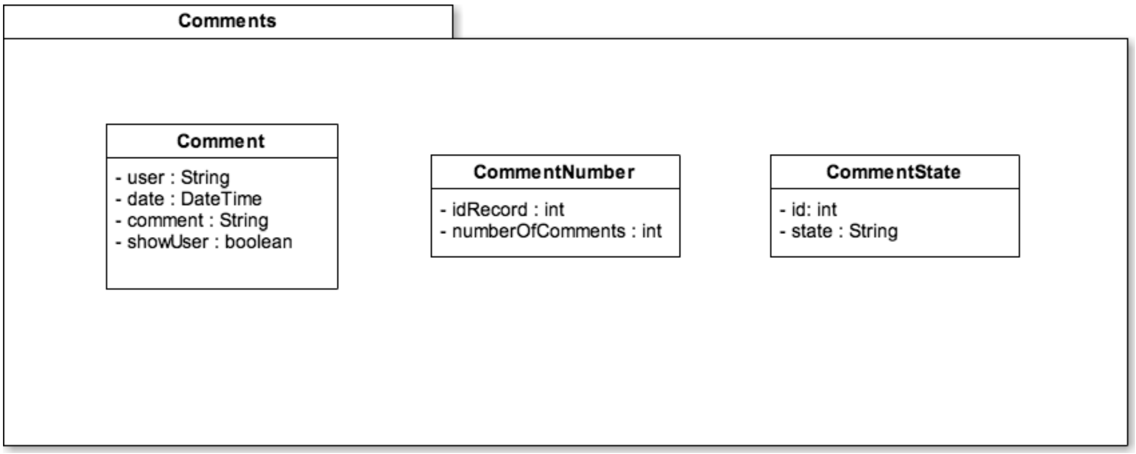


Figure 19 - Comments Package

3.2.3.6.1 Comment class

This class represents a comment made to a specific record

Attribute Name	Description
user	Username of the user who made the comment
date	Date of when the comment was made
comment	Contents of the comment
showUser	Flag used to know if the username is shown or not

Table 18 - Comment class attributes

3.2.3.6.2 CommentNumber class

This class specifies how many comments were made to a specific record

Attribute Name	Description
idRecord	ID of the record
numberOfComments	Number of comments made to that record

Table 19 - CommentNumber class attributes

3.2.3.6.3 CommentState class

This class is used to retrieve the name of the states that a comment may have. It will get the values from the Type_CommentState database table.

Attribute Name	Description
id	ID of the comment's state
state	State name

Table 20 - CommentState class attributes

3.2.3.6.4 Role class

This class is used to get the list of available roles that an authority member may have in a record. These roles are retrieved from the Type_Author_Role database table.

Attribute Name	Description
roleName	Names of the possible roles for an authority

Table 21 - Role class attributes

3.2.3.7 Controller Package

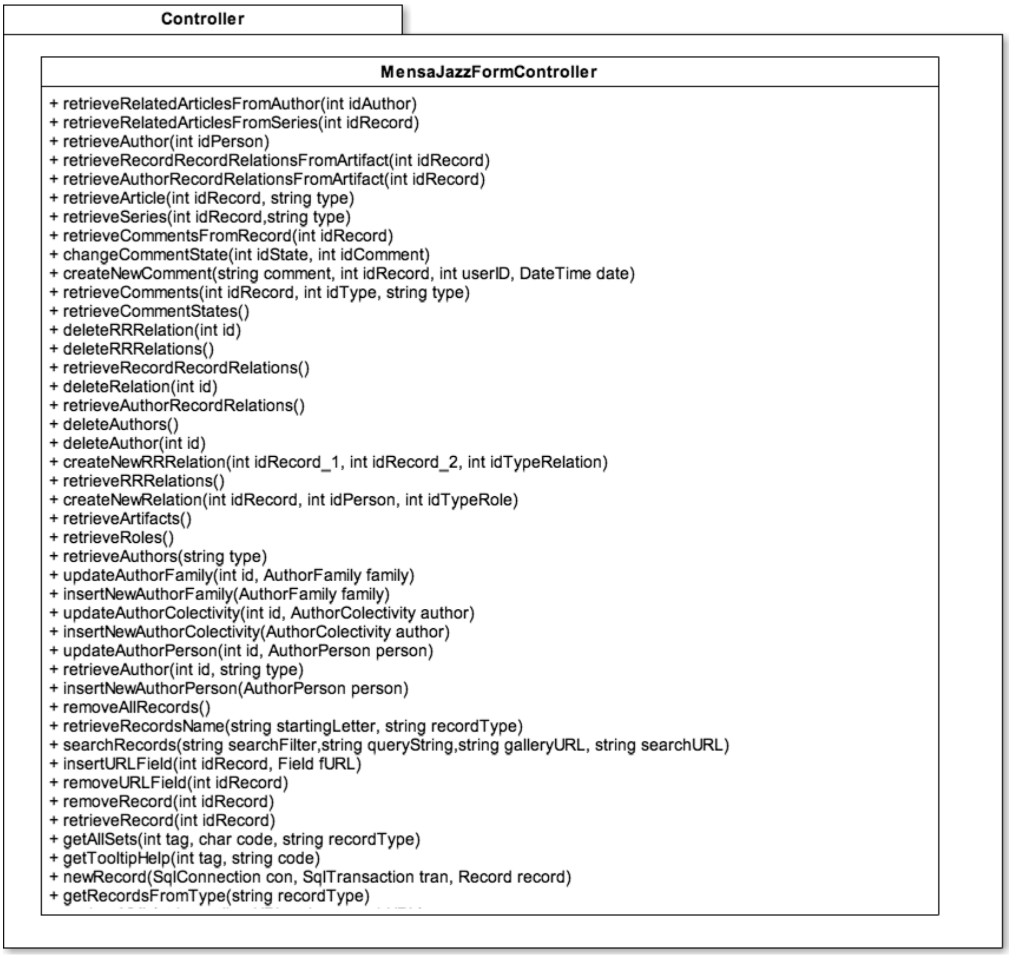

Controller
MensaJazzFormController
+ retrieveRelatedArticlesFromAuthor(int idAuthor)
+ retrieveRelatedArticlesFromSeries(int idRecord)
+ retrieveAuthor(int idPerson)
+ retrieveRecordRecordRelationsFromArtifact(int idRecord)
+ retrieveAuthorRecordRelationsFromArtifact(int idRecord)
+ retrieveArticle(int idRecord, string type)
+ retrieveSeries(int idRecord,string type)
+ retrieveCommentsFromRecord(int idRecord)
+ changeCommentState(int idState, int idComment)
+ createNewComment(string comment, int idRecord, int userID, DateTime date)
+ retrieveComments(int idRecord, int idType, string type)
+ retrieveCommentStates()
+ deleteRRRelation(int id)
+ deleteRRRelations()
+ retrieveRecordRecordRelations()
+ deleteRelation(int id)
+ retrieveAuthorRecordRelations()
+ deleteAuthors()
+ deleteAuthor(int id)
+ createNewRRRelation(int idRecord_1, int idRecord_2, int idTypeRelation)
+ retrieveRRRelations()
+ createNewRelation(int idRecord, int idPerson, int idTypeRole)
+ retrieveArtifacts()
+ retrieveRoles()
+ retrieveAuthors(string type)
+ updateAuthorFamily(int id, AuthorFamily family)
+ insertNewAuthorFamily(AuthorFamily family)
+ updateAuthorColectivity(int id, AuthorColectivity author)
+ insertNewAuthorColectivity(AuthorColectivity author)
+ updateAuthorPerson(int id, AuthorPerson person)
+ retrieveAuthor(int id, string type)
+ insertNewAuthorPerson(AuthorPerson person)
+ removeAllRecords()
+ retrieveRecordsName(string startingLetter, string recordType)
+ searchRecords(string searchFilter,string queryString,string galleryURL, string searchURL)
+ insertURLField(int idRecord, Field fURL)
+ removeURLField(int idRecord)
+ removeRecord(int idRecord)
+ retrieveRecord(int idRecord)
+ getAllSets(int tag, char code, string recordType)
+ getTooltipHelp(int tag, string code)
+ newRecord(SqlConnection con, SqlTransaction tran, Record record)
+ getRecordsFromType(string recordType)

Figure 20 - Controller package

3.2.3.7.1 MensaJazzFormController class

This class performs all the operations that need to access the Database. It is called in the Presentation Layer in order to ask for data from the Data Access Layer.

Method name	Description
retrieveRelatedArticlesFromAuthor(int idAuthor)	Given the id of an authority member this method will return a list of Articles in which that authority member takes a role
changeCommentState(int idState, int idComment)	This method will change the state of a comment
createNewComment(string comment, int idRecord, int userID, DateTime date)	This method will save a new comment made to a record. This comment will have the state “Not Validated”
createNewRelation(int idRecord, int idPerson, int idTypeRole)	This method creates a new relation between an authority and a bibliographic record
createNewRRRelation(int idRecord_1, int idRecord_2, int idTypeRelation)	This method will create a new relation between two records
deleteAuthor(int id)	Given the id of the authority member it will be deleted
deleteAuthors()	This method deletes all existent authority members
deleteRelation(int id)	Given the relation’s ID that relation will be deleted
deleteRRRelation(int id)	This method will delete a relation between two records
deleteRRRelations()	This method will delete all the existent relations between two records
formatXML(string inXML,string galleryURL,string searchURL)	This method will make some modifications to the XML returned from the database
getAllSets(int tag, char code, string recordType)	This method will return the value of an UNIMARC subfield with the provided field’s tag and subfield’s code

getRecordsFromType(string recordType)	This method returns a list of bibliographic records of a the provided type
getTooltipHelp(int tag, string code)	This method will return the description of an UNIMARC field
insertNewAuthorColectivity(AuthorColectivity author)	This method inserts a new authority member of type colectivity
insertNewAuthorFamily(AuthorFamily family)	This method inserts a new authority member of type family
insertNewAuthorPerson(AuthorPerson person)	This method will insert a new authority member of type person
insertURLField(int idRecord, Field fURL)	This method inserts a new UNIMARC 856 field to an existent bibliographic record
newRecord(SqlConnection con, SqlTransaction tran, Record Record)	This method creates a new bibliographic record
removeAllRecords()	This method removes all existent bibliographic records
removeRecord(int idRecord)	Given the id this method will delete an existent bibliographic record
removeURLField(int idRecord)	This method removes the UNIMARC 856 field from an existent bibliographic record
retrieveArticle(int idRecord, string type)	Given the id of the article this method will return a class with all the information regarding it
retrieveArtifacts()	This method retrieves the title of all the available bibliographic records
retrieveAuthor(int id, string type)	Given the id and type of authority this method will return the corresponding authority member
retrieveAuthor(int idPerson)	Given the id of an authority member this method will return it
retrieveAuthorRecordRelations()	This method will retrieve all

	existent relations between authority members and records
retrieveAuthorRecordRelationsFromArtifact(int idRecord)	Given the id a of a record this method will return all the relations that this record has with authority members
retrieveAuthors(string type)	This methods returns the list of available authority members according to their type
retrieveComments(int idRecord, int idType, string type)	This method will retrieve the comments of a bibliographic record that are in a specific state
retrieveCommentsFromRecord(int idRecord)	Given the id of a bibliographic record this method will return the list of validated comments made to it
retrieveCommentStates()	This method will return a list of available comment states
retrieveRecord(int idRecord)	This method will retrieve an existent bibliographic record
retrieveRecordRecordRelations()	This method will retrieve all the existent relations between any two bibliographic records
retrieveRecordRecordRelationsFromArtifact(int idRecord)	Given the id of a record this method will return all the relations that it has with other bibliographic records
retrieveRecordsName(string startingLetter, string recordType)	This method a list of the bibliographic record which title begins with the provided letter
retrieveRelatedArticlesFromSeries(int idRecord)	Given the id of a bibliographic record this method will return the list of articles that belong to the bibliographic record or have a relation with it
retrieveRoles()	This methods returns the list

	of all the available authority members' roles
retrieveRRRelations()	This method will retrieve all the possible names of a relation between two bibliographic records
retrieveSeries(int idRecord,string type)	Given the id of the periodical publication this method will return a class filled with all the information about it
retrieveXML(string galleryURL,string searchURL)	This method returns the XML with all the available bibliographic records
searchRecords(string searchFilter,string queryString,string galleryURL, string searchURL)	This methods performs a search on the database and returns a XML with the results and the time it took
updateAuthorColectivity(int id, AuthorColectivity author)	This method updates an existent authority member of type colectivity
updateAuthorFamily(int id, AuthorFamily family)	This method updates an existing authority member of type family
updateAuthorPerson(int id, AuthorPerson person)	This method updates an existent authority member of type person
updateRecord(Record Record, string oldSetName,string newSetName)	This method updates an existent bibliographic record

Table 22 - MensaJazzController class methods

3.2.4 Presentation Layer

This layer contains several user controls that together will form the graphical user interface of the website. User controls are *.ascx files and usually a DNN module has only 3 user controls. As the complexity of the module grows there is a need to create more user controls. They may interact with each other and DNN has specific navigation rules for achieving it. Some of the user controls are forms that the user must fill in with cataloguing information.

All the data entered is validated through the use of ASP.NET field validators. For a better user experience each time an user performs a successful or unsuccessful operation the system will show a message.

The description of every user control created for this work will be presented in the following sections.

3.2.4.1 User controls

CreateAuthor.ascx - This user control allows an administrator to create a new authority member. This authority member can be of a certain type:

- A colectivity;
- A family;
- A person.

For creating an authority member of the type colectivity the user must fill in a form with the name, language and history of it. The fields name and language are mandatory. In order to create an authority member of type family the user must fill in a form with the name and history of it. Only the name field is mandatory.

Finally for creating an authority member of type person the user must fill in a form with some personal information about it. This information consists in:

- Affiliation;
- Biography;
- Birthdate (*);
- Birthplace (*);
- Date of death;
- Lineage;
- Name (*);
- Place of death;
- Place of residence;
- Profession;

(*) Mandatory fields

Those mandatory fields are required in order to uniquely identify an authority member.

EditMensaJazzForm.acsx - This user control contains all the forms needed to create and maintain a new bibliographic record. The application supports two different types of bibliographic records: articles and periodical publications; periodical publications may contain several articles.

In this scenario an article always belongs to a specific periodical publication. The link between an article and its respective periodical publication is done through UNIMARC field with the tag 464 subfields $t (Periodical Publication’s title), $p (Range of pages) and $v (Volume).

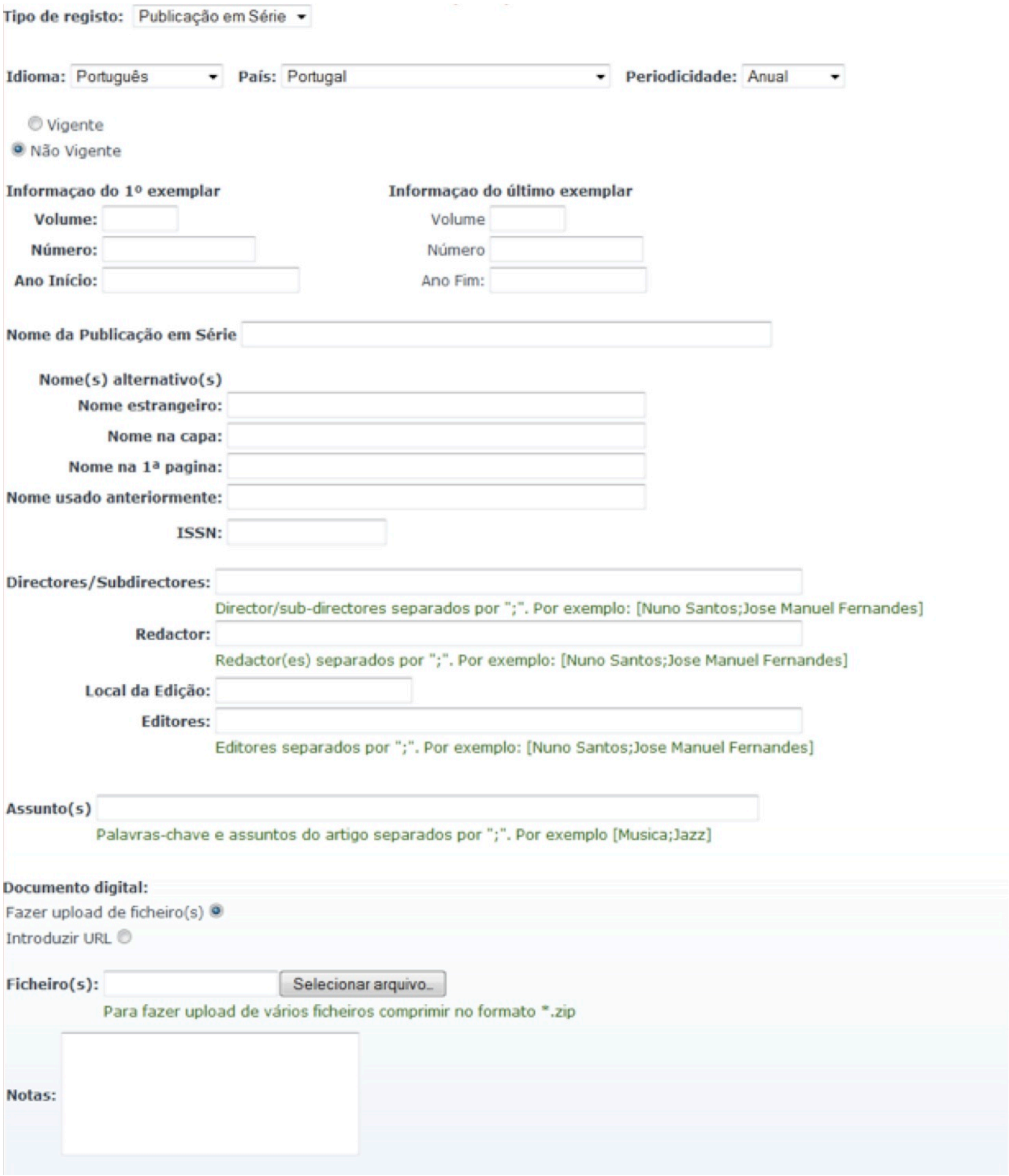
All the information entered in these forms will be placed in the respective UNIMARC fields/subfields and saved to the database.

When the administrator wants to insert new periodical publication it must fill in the following information:

- Language (*);

- Country (*);
- Periodicity (*);
- Choose whether if it still exists or not
- 1st and last copies available;
- Name (*);
- Alternative names;
- ISSN;
- Directors;
- Editors;
- Place of publication;
- Publishers;
- Subjects;
- Notes.

(*) Mandatory fields



Tipo de registo: Publicação em Série

Idioma: Português País: Portugal Periodicidade: Anual

○ Vigente
● Não Vigente

Informaçao do 1º exemplar
Volume:
Número:
Ano Início:

Informaçao do último exemplar
Volume
Número
Ano Fim:

Nome da Publicação em Série

Nome(s) alternativo(s)
Nome estrangeiro:
Nome na capa:
Nome na 1ª pagina:
Nome usado anteriormente:
ISSN:

Directores/Subdirectores:
Director/sub-directores separados por ";". Por exemplo: [Nuno Santos;Jose Manuel Fernandes]
Redactor:
Redactor(es) separados por ";". Por exemplo: [Nuno Santos;Jose Manuel Fernandes]
Local da Edição:
Editores:
Editores separados por ";". Por exemplo: [Nuno Santos;Jose Manuel Fernandes]

Assunto(s)
Palavras-chave e assuntos do artigo separados por ";". Por exemplo [Musica;Jazz]

Documento digital:
Fazer upload de ficheiro(s) ●
Introduzir URL ○

Ficheiro(s): Selecionar arquivo...
Para fazer upload de vários ficheiros comprimir no formato *.zip

Notas:

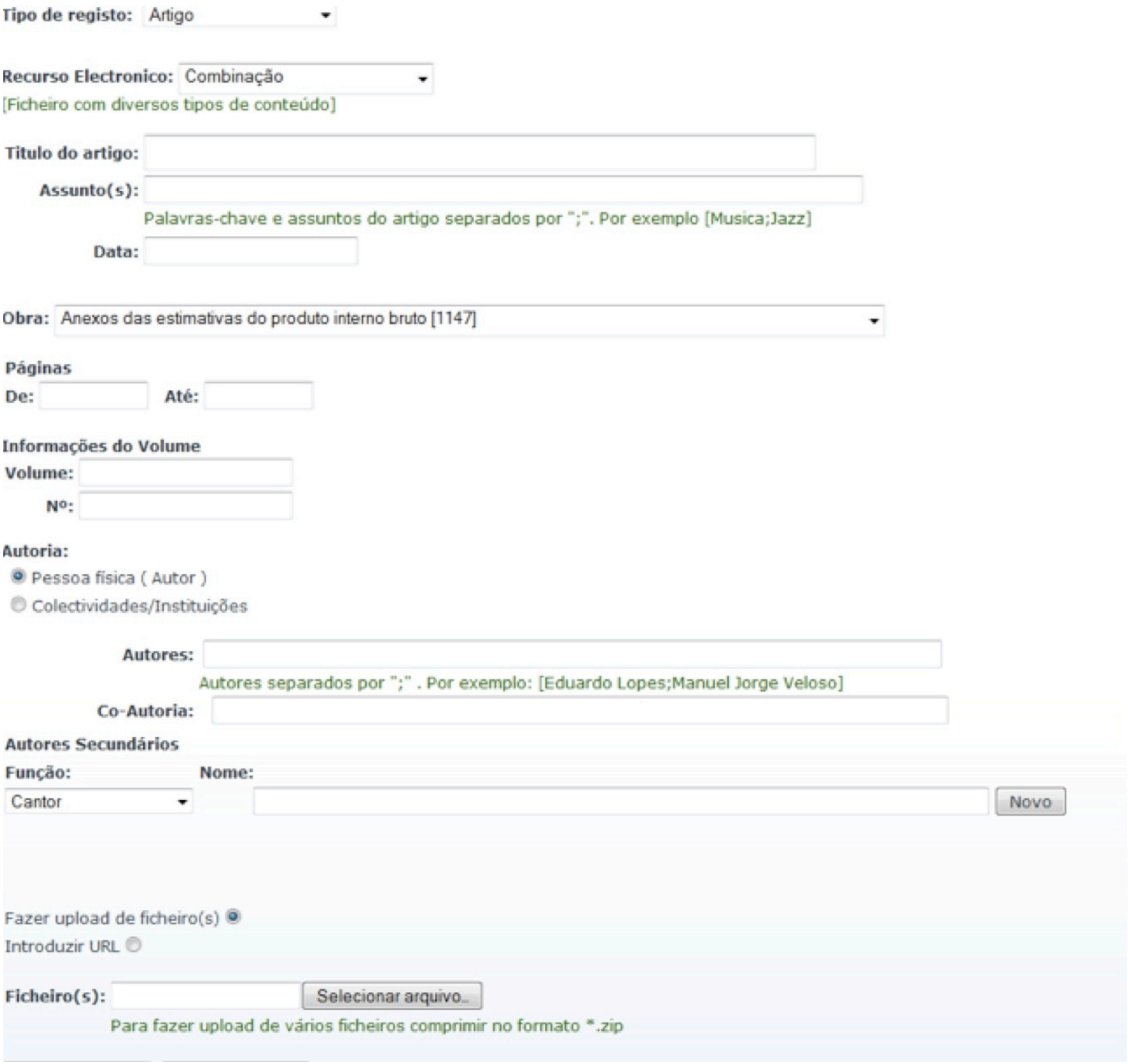
Figure 21 - Periodical publication input form

On the other hand if the administrator wants to insert a new article it must fill in the following information:

- Type of electronic resource;
- Title (*);
- Subjects;
- Date of publication;
- Periodical publication it belongs to;
- Range of pages in the periodical publication;
- Volume information;
- Authors (*);

- Co-authors and secondary authors with their respective roles.

(*) Mandatory fields



Tipo de registo: Artigo

Recurso Electronico: Combinação
[Ficheiro com diversos tipos de conteúdo]

Titulo do artigo:

Assunto(s):
Palavras-chave e assuntos do artigo separados por ";". Por exemplo [Musica;Jazz]

Data:

Obra: Anexos das estimativas do produto interno bruto [1147]

Páginas
De: Até:

Informações do Volume
Volume:
Nº:

Autoria:
● Pessoa física (Autor)
○ Colectividades/Instituições

Autores:
Autores separados por ";" . Por exemplo: [Eduardo Lopes;Manuel Jorge Veloso]

Co-Autoria:

Autores Secundários
Função: Nome:
Cantor

Fazer upload de ficheiro(s) ●
Introduzir URL ○

Ficheiro(s):
Para fazer upload de vários ficheiros comprimir no formato *.zip

Figure 22 - Article input form

The administrator may also upload scanned images of the periodical publication/article or to provide an external URL. Since there is only an upload control the administrator has the possibility to compress several files into a single .zip file and upload it all at once. All the uploaded .zip files will be uncompressed when received and stored in the server.

All the data entered by the administrator is validated, and in case of an error the respective error message is displayed. These errors may happen, for instance, if any mandatory data is missing or if a field is filled in the wrong way or format.

This control is used for editing an existing record as well. For this it is used a query string with the name “recordID” containing the value of the primary key from the database belonging to the bibliographic record. If an invalid “recordID” value is passed an error message will be displayed to the administrator.

IndexMenu.ascx - This user control displays a list of links for the different types of indexes available:

- Index of articles
- Index of authors
- Index of periodical publications
- Index of subjects

The reason why there weren’t created menu items for each of the index pages it is because DNN doesn’t allow setting query string parameters for menu items. To minimize the repetition of code the index pages use the same *.ascx file that will display a different type of index accordingly to the value of the query string “Type”.

IndexRecords.ascx – This user control displays the indexing of available information accordingly to the type chosen. The type is chosen accordingly to the query string “Type”. The index results are displayed grouped by their starting letter. The user may chose any starting letter and the results will be updated.

Type of index	QueryString[“Type”] Value	Description
Article	Artigo	Shows the list of articles available accordingly to the letter chosen.
Author	Autor	Shows the list of authors available accordingly to the letter chosen.
Periodical Publication	Publicação em Série	Shows the list of periodical publications available accordingly to the letter chosen
Subjects	Assunto	Shows the list of subjects available accordingly to the letter chosen

Table 23 - IndexRecords query strings

ListAuthor.ascx - This user control displays the list of available authority members. These authority members must have been created with the CreateAuthor user control.

The administrator may filter out the type of authority member displayed in the table. It can also delete or edit a selected authority member.

ListComments.ascx - This user control allows an administrator to check the list of comments accordingly to the bibliographic record and state chosen. It can also change the state of any comment.

An user comment will only be shown in the portal after being validated.

ListPendingComments.ascx – This user control displays new user comments waiting to be validated. The administrator may accept or refuse these comments. It allows the administrator to easily manage new comments.

ListRecords.ascx - This user control allows an administrator to check the list of available bibliographic records. It can also edit or delete a bibliographic record.

The type of bibliographic record shown can be chosen as well. In order to find a specific bibliographic record faster the administrator may enter its ID.

ListRelation.ascx - This user control allows an administrator to check all existent relations between an authority member and a bibliographic record. It also allows the administrator to remove any of the listed relations.

ListRRRelation.ascx - This user control allows an administrator to check all existent relations between two distinct records. The administrator has also the option to remove any of the relations displayed.

NewRelation.ascx - This user control allows an administrator to create a new relation between an authority member and a bibliographic record. The administrator may also choose the role that the authority member will have. If there aren't any authors or bibliographic records in the system an error message will be displayed.

NewRRRelation.ascx - This user control allows an administrator to create a relation between two distinct bibliographic records. If there isn't more than 1 bibliographic record it won't be possible to create a relation.

PicGallery.ascx - This user control allows an user to visualize the digital content of a bibliographic record that is stored in a scanned image of the original document. For the gallery it was used a jQuery plugin named Galleriffic [15]. Depending on the number of images uploaded for a specific bibliographic record the user may use the arrow keys for changing to the next or previous image respectively.

Relations.ascx - This user control displays links for listing the types of relations:

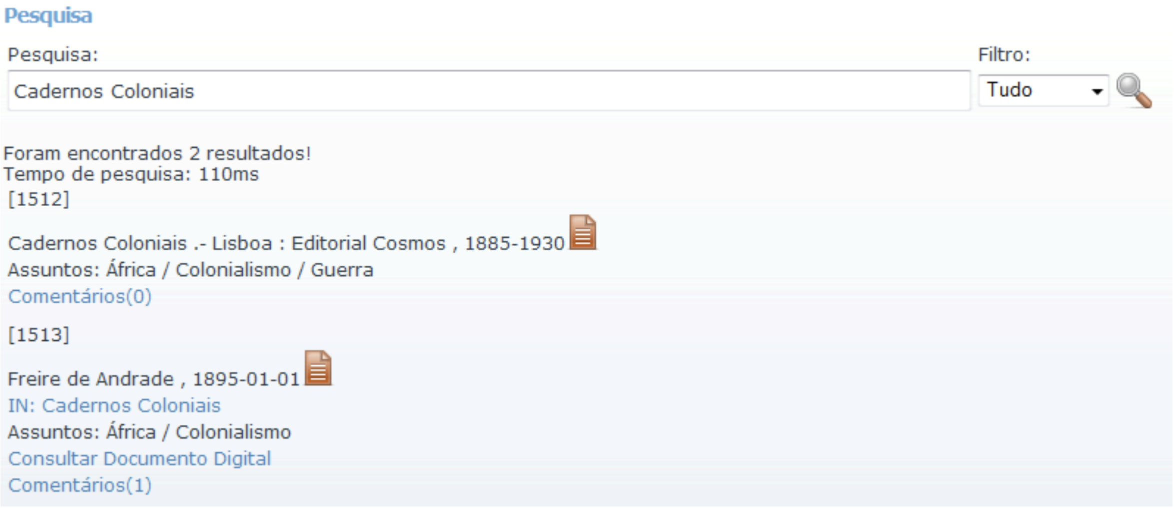
- Authority-to-record relations.
- Record-to-record relations;

Search.ascx - This user control allows a user to search in the Jazz Messengers repository. The user may filter the results by choosing one of the categories available:

- Author;
- General;
- Title;
- Subject.

After a search is made a status label with the number of results found and the time it took in microseconds is presented.

Each bibliographic record found in the search has a small description like title, year of publication, authors and subjects. These fields are used as source for other queries containing such terms. It is also displayed the number of comments made to each bibliographic record. When the user clicks on the number of comments it will visualize the comments. If the user is logged in he may also leave his comment. If the user wants to see more details about a certain bibliographic record he may click on the image next to the title. In Figure 23 is displayed the results of a search performed.



Pesquisa

Pesquisa: Cadernos Coloniais

Filtro: Tudo

Foram encontrados 2 resultados!
Tempo de pesquisa: 110ms

[1512]

Cadernos Coloniais .- Lisboa : Editorial Cosmos , 1885-1930
Assuntos: África / Colonialismo / Guerra
Comentários(0)

[1513]

Freire de Andrade , 1895-01-01
IN: Cadernos Coloniais
Assuntos: África / Colonialismo
Consultar Documento Digital
Comentários(1)

Figure 23 - Search example

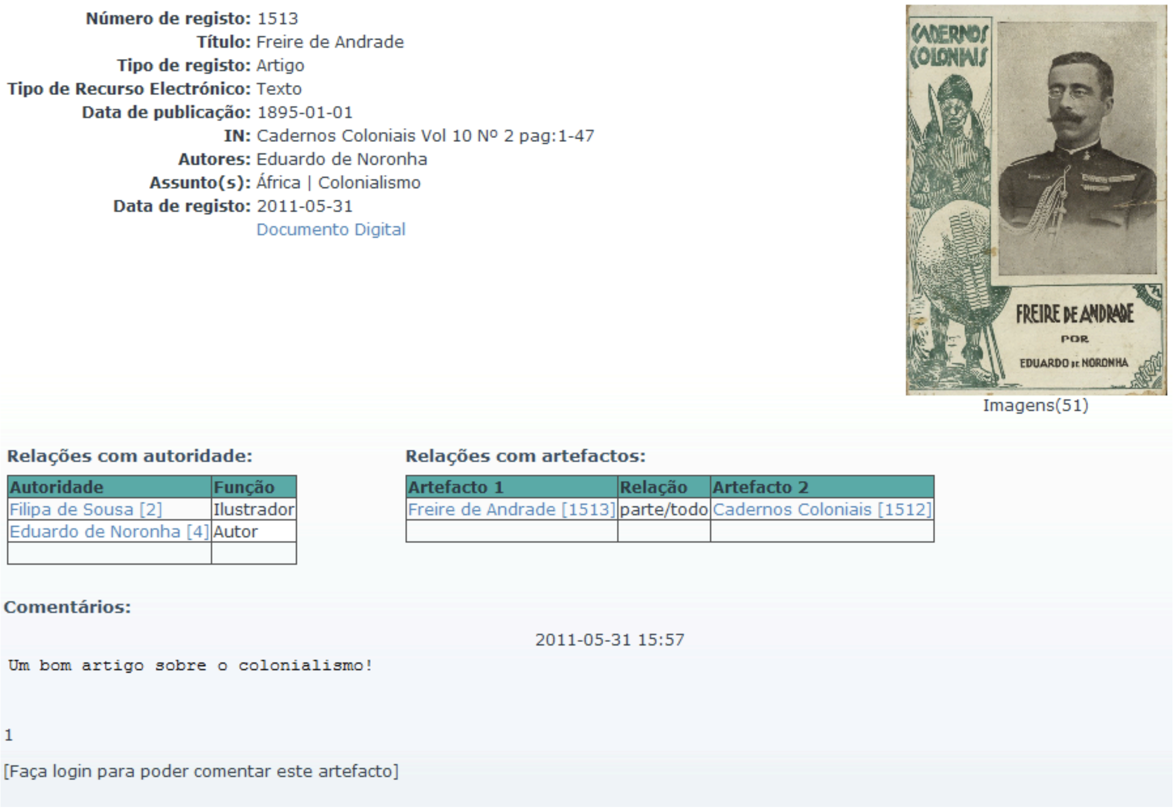
ViewAuthor.ascx - This user control allows an user to see an authority member's full description.

ViewComments.ascx - This user controls allows an user to check the list of all comments made to a specific bibliographic record. If the user is logged in he may also leave its comment.

ViewRecordDetails.ascx - This user control is displayed when the user selects the option to show more details about a certain bibliographic record. Here it will be displayed all the information existent about that bibliographic record.

Depending on if the bibliographic record is an article or a periodical publication there will be displayed all the UNIMARC fields belonging to it. If it has any uploaded image, the first one is displayed as a thumbnail next to the UNIMARC information. If the selected bibliographic record is a periodical publication all the related articles will be displayed as links to them. After that all the existent relations between the selected bibliographic record and authority members or other distinct bibliographic records are listed in tables. These relations are displayed in the form of links so the user is able to check for detailed information about a certain authority member or bibliographic record respectively. In the end of the page are displayed the comments made by users. If the user is logged in he may also leave his comment.

For a better understanding an example is presented in Figure 24.



Número de registo: 1513
Título: Freire de Andrade
Tipo de registo: Artigo
Tipo de Recurso Electrónico: Texto
Data de publicação: 1895-01-01
IN: Cadernos Coloniais Vol 10 Nº 2 pag:1-47
Autores: Eduardo de Noronha
Assunto(s): África | Colonialismo
Data de registo: 2011-05-31
Documento Digital

Imagens(51)

Relações com autoridade:

Autoridade	Função
Filipa de Sousa [2]	Ilustrador
Eduardo de Noronha [4]	Autor

Relações com artefactos:

Artefacto 1	Relação	Artefacto 2
Freire de Andrade [1513]	parte/todo	Cadernos Coloniais [1512]

Comentários:

2011-05-31 15:57

Um bom artigo sobre o colonialismo!

1

[Faça login para poder comentar este artefacto]

Figure 24 - Record details example

3.2.4.2 Module definitions

Module definitions are used to group user controls that together perform the desired operations. This will allow the navigation between user controls. Each module definition can be considered as a logic group of functionality. Each user control added to the module definition must have a specific key defined. This key is used with the DNN method “EditURL(“key”)” allowing the navigation between the different user controls within that module definition. The first user control displayed has its key value blank.

A module definition can be created using the DNN host menu. Host > Module Definitions, then choose edit the module. After that scroll down to “Module Definitions” and click the button “Add Definition”. The following module definitions were created:

Module Definition	User Control(s)	Comments
Indexing	IndexMenu.ascx ViewAuthor.ascx PicGallery.ascx IndexRecords.ascx ViewRecordDetails.ascx Search.ascx	This module definition is for indexing the available authors, subjects, articles and periodical publications. It also allows getting more details about the bibliographic records and its authors. For each subject indexed there is a link that clicked will perform a search for bibliographic records associated
Search	Search.ascx PicGallery.ascx ViewRecordDetails.ascx ViewComments.ascx	This module definition is used when a search is performed. The user may check a bibliographic record details, its images and comments.
New Record	EditMensaJazzForm.ascx	This module definition creates new bibliographic records
Record Listing	ListRecords.ascx PicGallery.ascx EditMensaJazzForm.ascx	This module definition is for listing all the available records and to perform edit/delete operations. It also allows the user to visualize the available images if any.
Create Record-to-Record Relation	NewRRRelation.ascx	This module definition is to create a new Record-to-

		Record relation.
Create Author-to-Record Relation	NewRelation.ascx	This module definition allows the creation of a new Authority-to Record-relation.
Listing Relations	Relations.ascx ListRelation.ascx ListRRRelation.ascx	This module definition allows the listing and management of all relations
Create new authority member	CreateAuthor.ascx	This module definition allows the creation of a new authority member.
Listing Authority Members	ListAuthor.ascx	This module definition is for listing all authority members and performs the edit/delete operations on them.
Listing comments	ListComments.ascx ListPendingComments.ascx	This module allows the management of all the comments performed. All comments states can be changed here.

Table 24 - Module Definitions

Chapter 4 - **The MensaJazz Portal**

In this chapter it will be presented the portal developed in this work named MensaJazz. It was called that way because it will store catalogued information regarding the Jazz Messengers. It will be described the portal's front office and back office. The description is followed by screenshots of the actual portal.

4.1 Front office

The front office is available to be used by all users of the portal. It allows the search of information stored in the digital repository.

4.1.1 Home page

The home page (Figure 25) allows any user to perform searches in the digital library. There are several search options:

"Assunto" – searches for bibliographic records according to their subject
"Autor" – searches for bibliographic records by their author
"Título" – searches for bibliographic record using their title
"Tudo" – searches for anything related to the bibliographic record

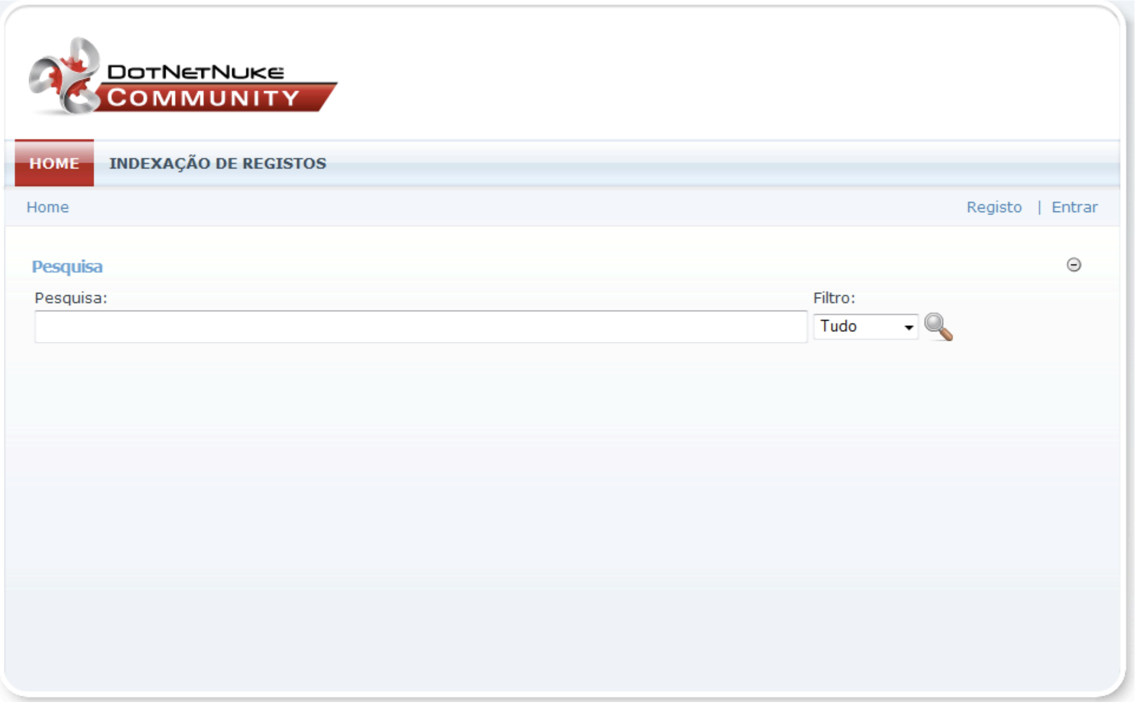
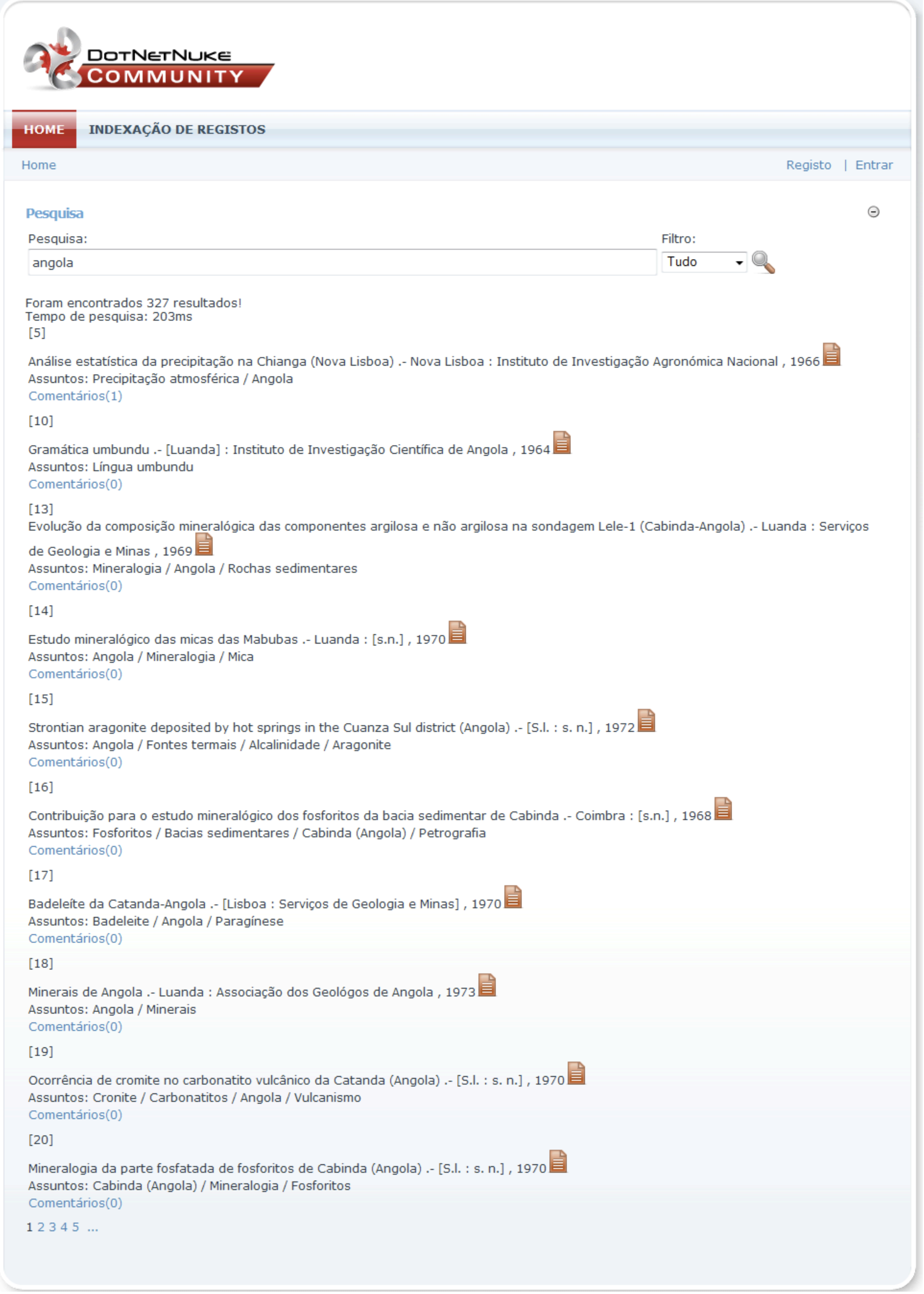


Figure 25 - Home page

4.1.2 Search results

When a user performs a search its results (Figure 26) will be displayed in the same page. Each result found is a small description of a bibliographic record. It is also shown the number of comments that it has. If the user wants to see more details about a certain bibliographic record it may click the button that is next to the title (Figure 28 and Figure 29). If a bibliographic record has its digital contents available the user may click on its link to visualize it (Figure 35). If the user clicks on the link that shows up how many comments were made it will go to a page that displays all comments (Figure 30).

If no results are found it is displayed a warning message to the user (Figure 27).



Pesquisa

Pesquisa: angola

Filtro: Tudo

Foram encontrados 327 resultados!
Tempo de pesquisa: 203ms

[5]

Análise estatística da precipitação na Chianga (Nova Lisboa) .- Nova Lisboa : Instituto de Investigação Agronómica Nacional , 1966
Assuntos: Precipitação atmosférica / Angola
Comentários(1)

[10]

Gramática umbundu .- [Luanda] : Instituto de Investigação Científica de Angola , 1964
Assuntos: Língua umbundu
Comentários(0)

[13]

Evolução da composição mineralógica das componentes argilosa e não argilosa na sondagem Lele-1 (Cabinda-Angola) .- Luanda : Serviços de Geologia e Minas , 1969
Assuntos: Mineralogia / Angola / Rochas sedimentares
Comentários(0)

[14]

Estudo mineralógico das micas das Mabubas .- Luanda : [s.n.] , 1970
Assuntos: Angola / Mineralogia / Mica
Comentários(0)

[15]

Strontian aragonite deposited by hot springs in the Cuanza Sul district (Angola) .- [S.l. : s. n.] , 1972
Assuntos: Angola / Fontes termais / Alcalinidade / Aragonite
Comentários(0)

[16]

Contribuição para o estudo mineralógico dos fosforitos da bacia sedimentar de Cabinda .- Coimbra : [s.n.] , 1968
Assuntos: Fosforitos / Bacias sedimentares / Cabinda (Angola) / Petrografia
Comentários(0)

[17]

Badeleíte da Catanda-Angola .- [Lisboa : Serviços de Geologia e Minas] , 1970
Assuntos: Badeleite / Angola / Paragínese
Comentários(0)

[18]

Minerais de Angola .- Luanda : Associação dos Geológos de Angola , 1973
Assuntos: Angola / Minerais
Comentários(0)

[19]

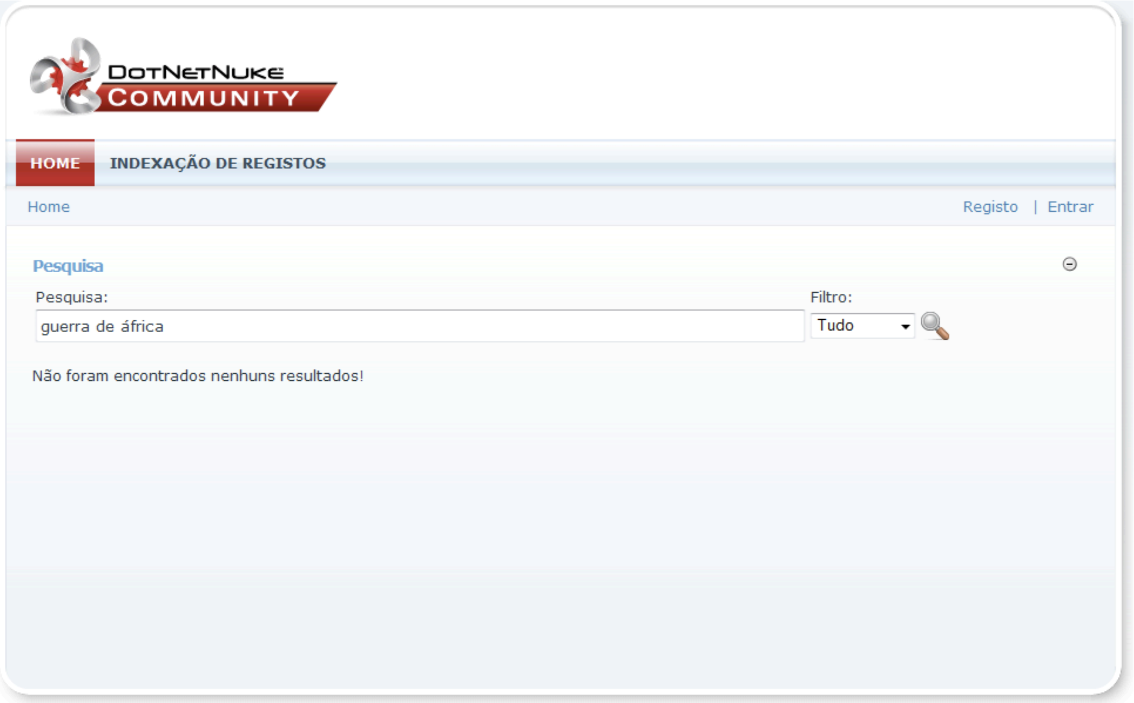
Ocorrência de cromite no carbonatito vulcânico da Catanda (Angola) .- [S.l. : s. n.] , 1970
Assuntos: Cronite / Carbonatitos / Angola / Vulcanismo
Comentários(0)

[20]

Mineralogia da parte fosfatada de fosforitos de Cabinda (Angola) .- [S.l. : s. n.] , 1970
Assuntos: Cabinda (Angola) / Mineralogia / Fosforitos
Comentários(0)

1 2 3 4 5 ...

Figure 26 - Search results



Pesquisa: guerra de áfrica

Filtro: Tudo

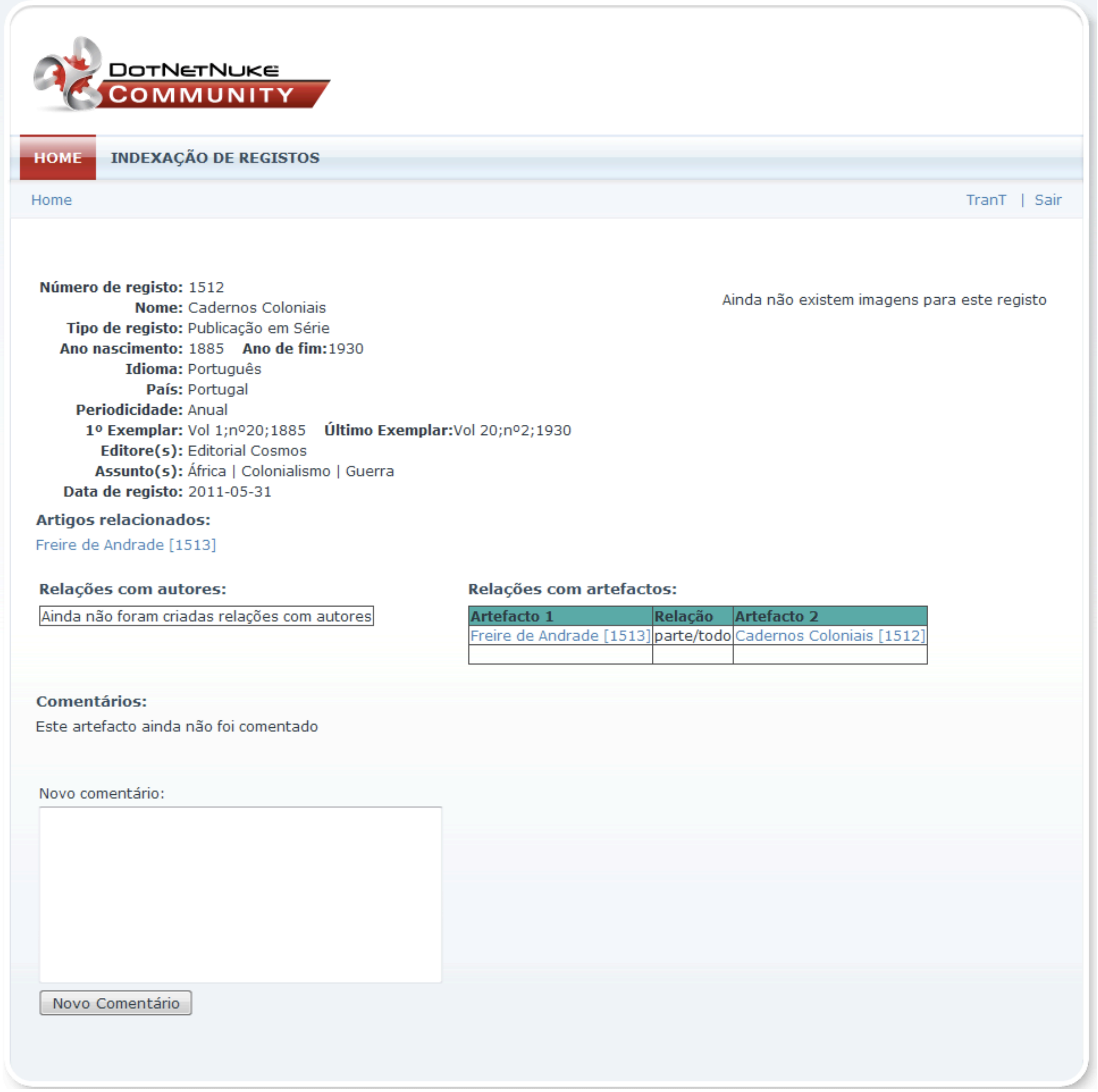
Não foram encontrados nenhuns resultados!

Figure 27 - No results found on search

4.1.3 Periodic Publication details page

When the user clicks on the details button from a search result that is a periodical publication it will be shown all the information regarding it. In Figure 28 it can be seen that the UNIMARC information about that bibliographic record is displayed first.

On the right appears a thumbnail of the first image available from that bibliographic record if any. Bellow that, there are listed the relations between that periodical publication and authority members and the relations between other bibliographic records. In those tables each reference to an authority member or another bibliographic record is a link to their respective details. At the end of the page there are listed the user's comments about that periodical publication.



Número de registo: 1512
Nome: Cadernos Coloniais
Tipo de registo: Publicação em Série
Ano nascimento: 1885 **Ano de fim:**1930
Idioma: Português
País: Portugal
Periodicidade: Anual
1º Exemplar: Vol 1;nº20;1885 **Último Exemplar:**Vol 20;nº2;1930
Editore(s): Editorial Cosmos
Assunto(s): África | Colonialismo | Guerra
Data de registo: 2011-05-31

Ainda não existem imagens para este registo

Artigos relacionados:
Freire de Andrade [1513]

Relações com autores:
Ainda não foram criadas relações com autores

Relações com artefactos:

Artefacto 1	Relação	Artefacto 2
Freire de Andrade [1513]	parte/todo	Cadernos Coloniais [1512]

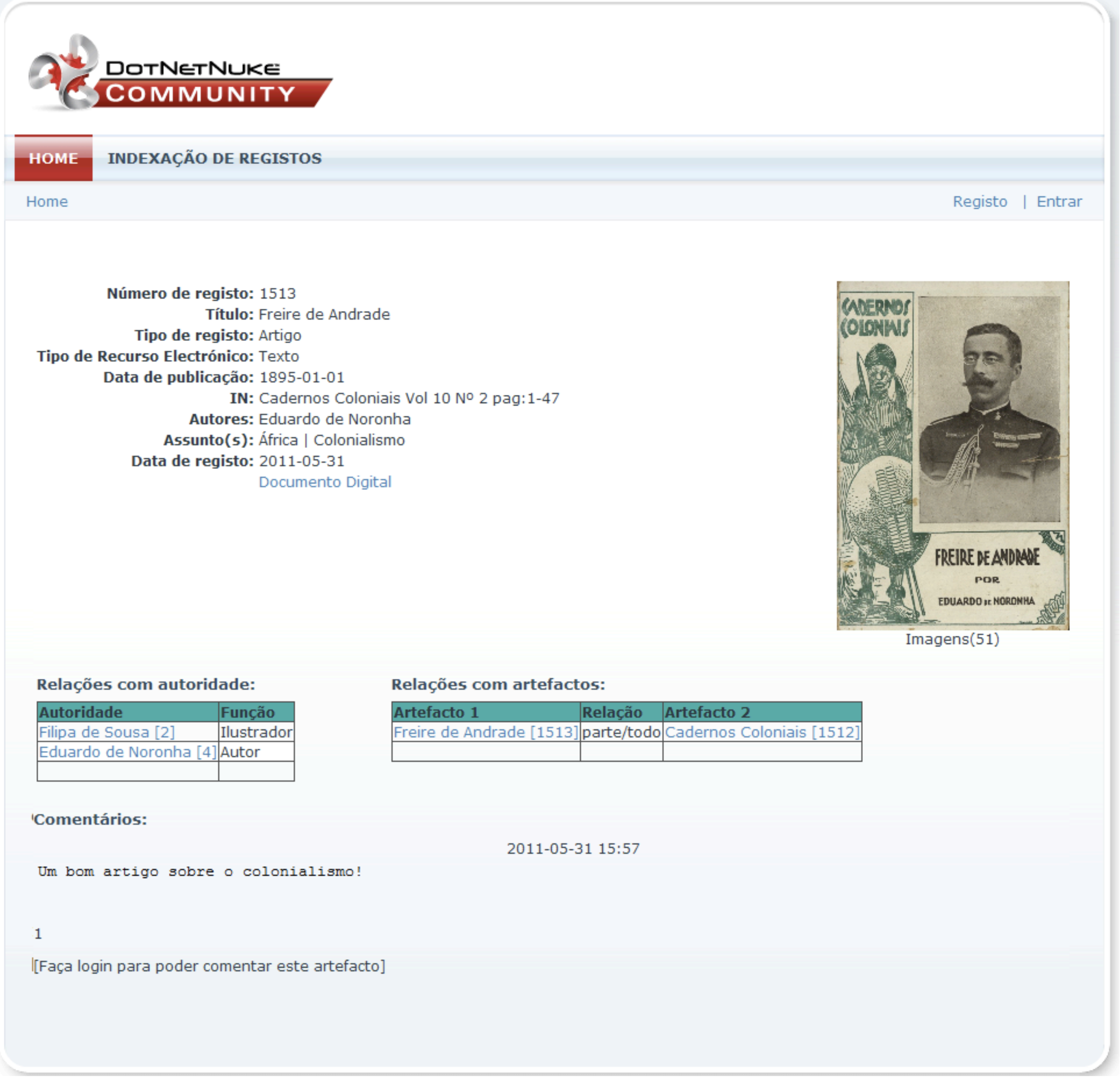
Comentários:
Este artefacto ainda não foi comentado

Novo comentário:

Figure 28 - Periodical Publication details

4.1.4 Article’s details page

This page has the same structure as the one described for the periodical publication but this time describing an article.



Número de registo: 1513
Título: Freire de Andrade
Tipo de registo: Artigo
Tipo de Recurso Electrónico: Texto
Data de publicação: 1895-01-01
IN: Cadernos Coloniais Vol 10 Nº 2 pag:1-47
Autores: Eduardo de Noronha
Assunto(s): África | Colonialismo
Data de registo: 2011-05-31
Documento Digital

Imagens(51)

Relações com autoridade:

Autoridade	Função
Filipa de Sousa [2]	Ilustrador
Eduardo de Noronha [4]	Autor

Relações com artefactos:

Artefacto 1	Relação	Artefacto 2
Freire de Andrade [1513]	parte/todo	Cadernos Coloniais [1512]

Comentários:

2011-05-31 15:57

`Um bom artigo sobre o colonialismo!`

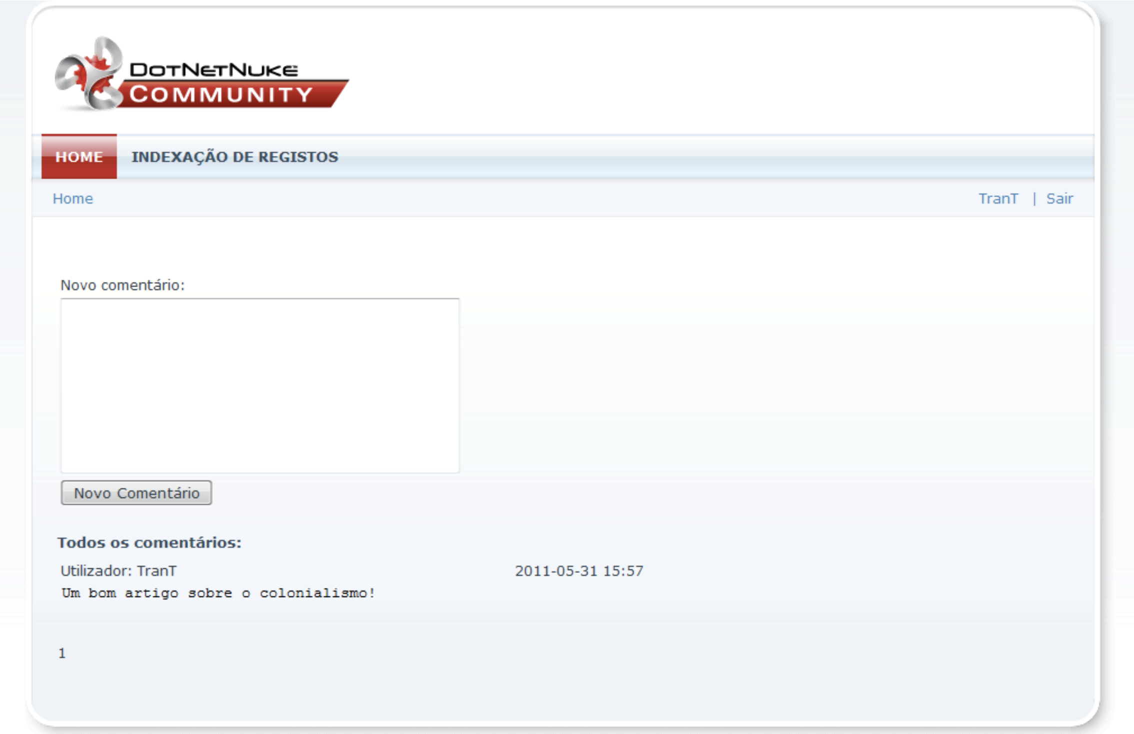
1

[Faça login para poder comentar este artefacto]

Figure 29 - Article's description

4.1.5 Comments

The user may check for the bibliographic records comments by clicking on the link available in the search results or by checking its details. A comment author's username is only shown to authenticated users and only these users may leave comments.



Novo comentário:

Todos os comentários:

Utilizador: TranT 2011-05-31 15:57

Um bom artigo sobre o colonialismo!

1

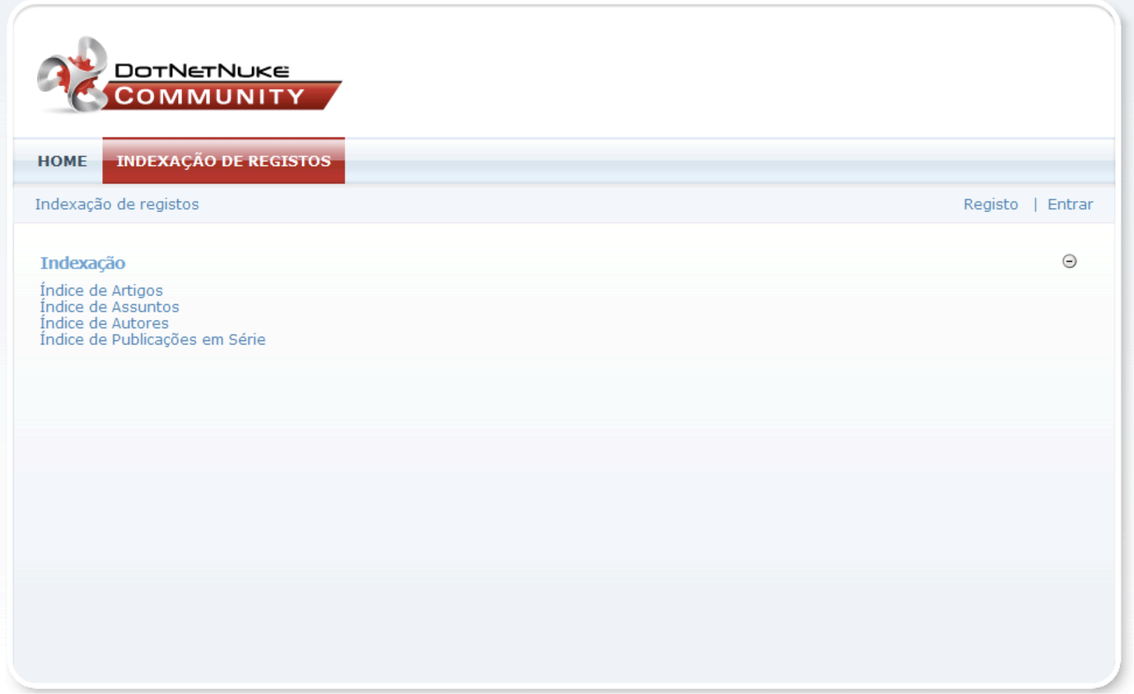
Figure 30 - Bibliographic record comments

4.1.6 Index page

The index page has four links (Figure 31) each one representing a different type of index:

- "Índice de artigos" – Article's index;
- "Índice de assuntos" – Subject's índex;
- "Índice de autores" – Author's índex;
- "Índice de publicações periódicas" – Periodical Publication's índex.

Depending on the selected type of index the page will display a list of all articles/subjects/authors or periodical publications available. This list is grouped accordingly to their first starting letter. If the user clicks on one of the links from that list a search will be performed with it. Some entries may have a details button that can be clicked for displaying more information (Authority member's details in Figure 34).



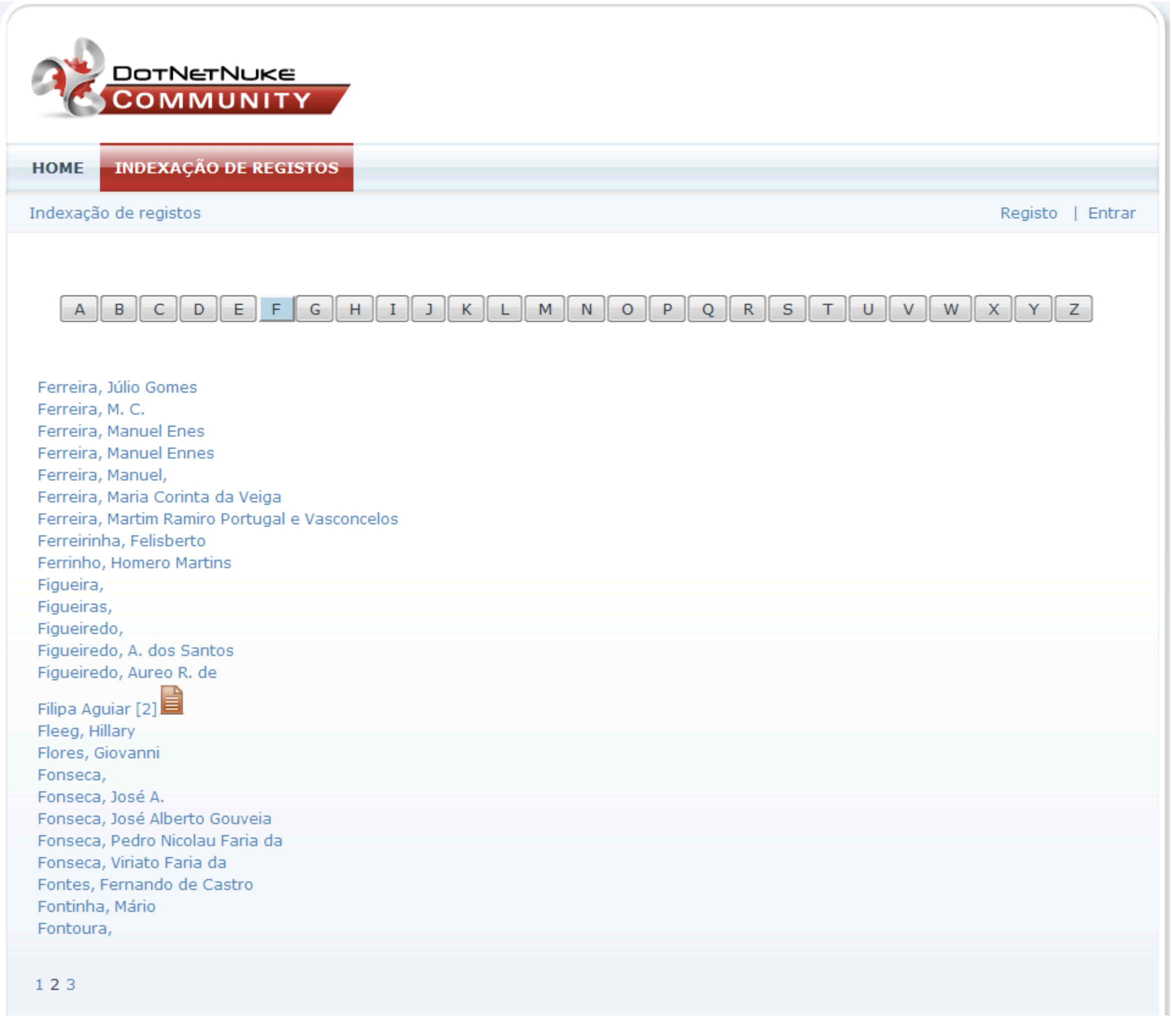
Indexação de registos

Indexação

Índice de Artigos
Índice de Assuntos
Índice de Autores
Índice de Publicações em Série

Figure 31 - Index page

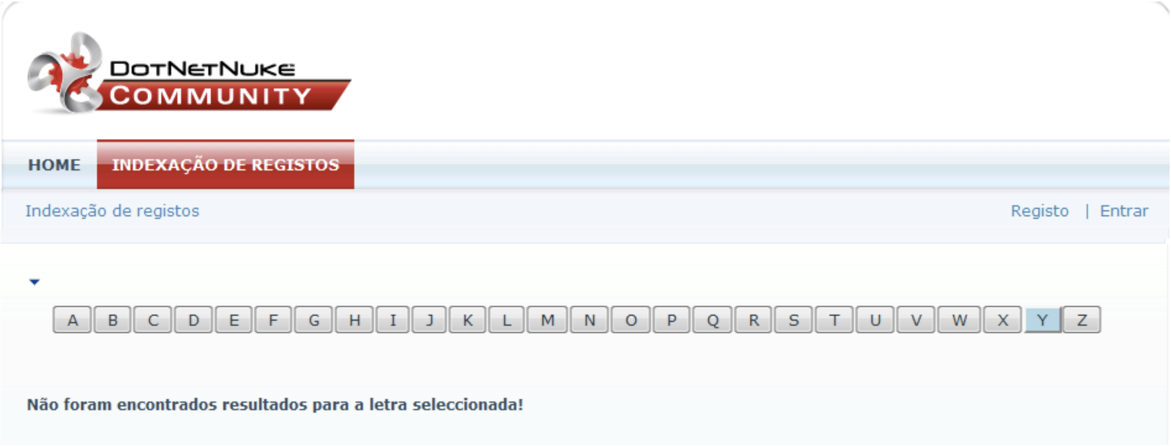
In Figure 32 it is shown the index of all available authors that have their name starting with the selected letter (in this case 'F'). The user can check more information about those authors it by clicking on their details button. In Figure 32 the author "Filipa de Sousa" has that details button. All those authors that have a details button belong to the authority index.



Ferreira, Júlio Gomes
Ferreira, M. C.
Ferreira, Manuel Enes
Ferreira, Manuel Ennes
Ferreira, Manuel,
Ferreira, Maria Corinta da Veiga
Ferreira, Martim Ramiro Portugal e Vasconcelos
Ferreirinha, Felisberto
Ferrinho, Homero Martins
Figueira,
Figueiras,
Figueiredo,
Figueiredo, A. dos Santos
Figueiredo, Aureo R. de
Filipa Aguiar [2]
Fleeg, Hillary
Flores, Giovanni
Fonseca,
Fonseca, José A.
Fonseca, José Alberto Gouveia
Fonseca, Pedro Nicolau Faria da
Fonseca, Viriato Faria da
Fontes, Fernando de Castro
Fontinha, Mário
Fontoura,

Figure 32 - Author's index

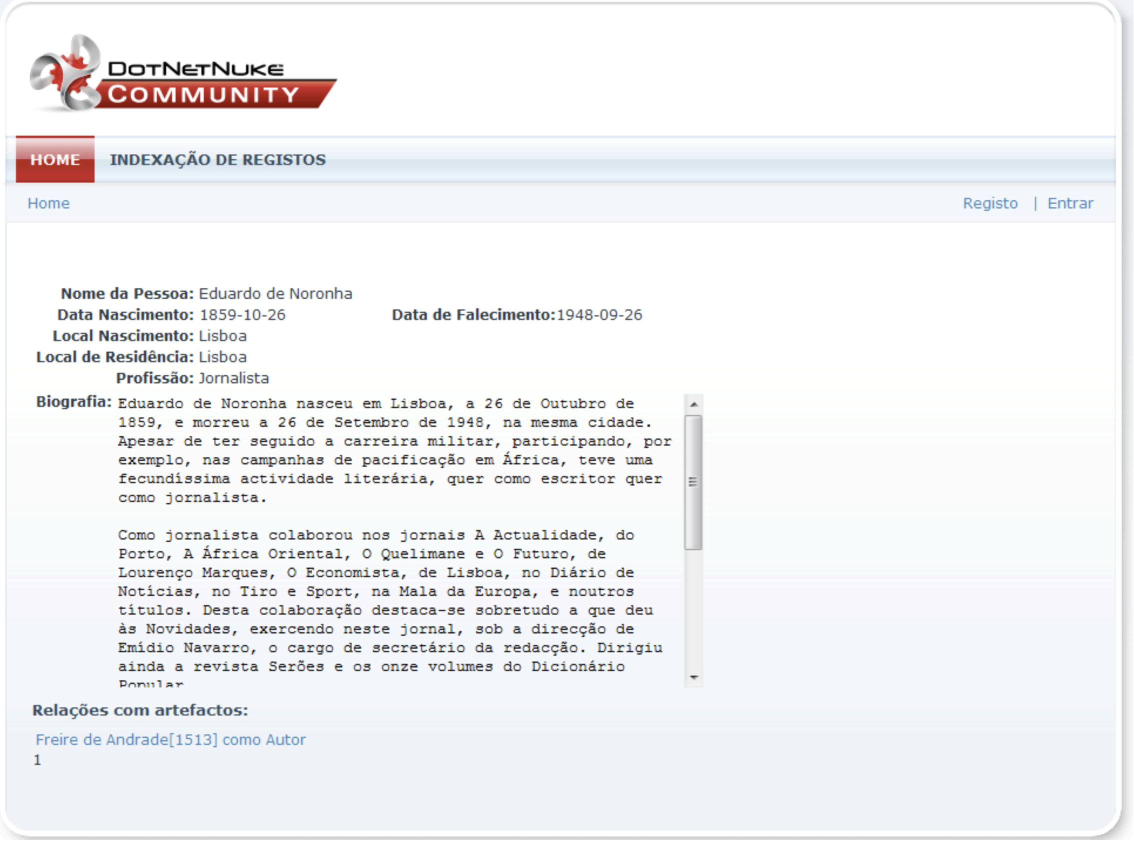
If there are no authors with the selected letter a warning message is displayed to the user (Figure 33).



Não foram encontrados resultados para a letra seleccionada!

Figure 33 - No results found

By clicking in the details button of the author “Eduardo de Noronha” the page from Figure 34 is displayed.



Nome da Pessoa: Eduardo de Noronha
Data Nascimento: 1859-10-26 **Data de Falecimento:**1948-09-26
Local Nascimento: Lisboa
Local de Residência: Lisboa
Profissão: Jornalista
Biografia: Eduardo de Noronha nasceu em Lisboa, a 26 de Outubro de 1859, e morreu a 26 de Setembro de 1948, na mesma cidade. Apesar de ter seguido a carreira militar, participando, por exemplo, nas campanhas de pacificação em África, teve uma fecundíssima actividade literária, quer como escritor quer como jornalista.

Como jornalista colaborou nos jornais A Actualidade, do Porto, A África Oriental, O Quelimane e O Futuro, de Lourenço Marques, O Economista, de Lisboa, no Diário de Notícias, no Tiro e Sport, na Mala da Europa, e noutros títulos. Desta colaboração destaca-se sobretudo a que deu às Novidades, exercendo neste jornal, sob a direcção de Emídio Navarro, o cargo de secretário da redacção. Dirigiu ainda a revista Serões e os onze volumes do Dicionário Popular

Relações com artefactos:
Freire de Andrade[1513] como Autor
1

Figure 34 - Author's details

4.1.7 Gallery page

In this page the user may visualize all the available pictures of a bibliographic record. It can use the arrow keys to go to the next or previous picture.

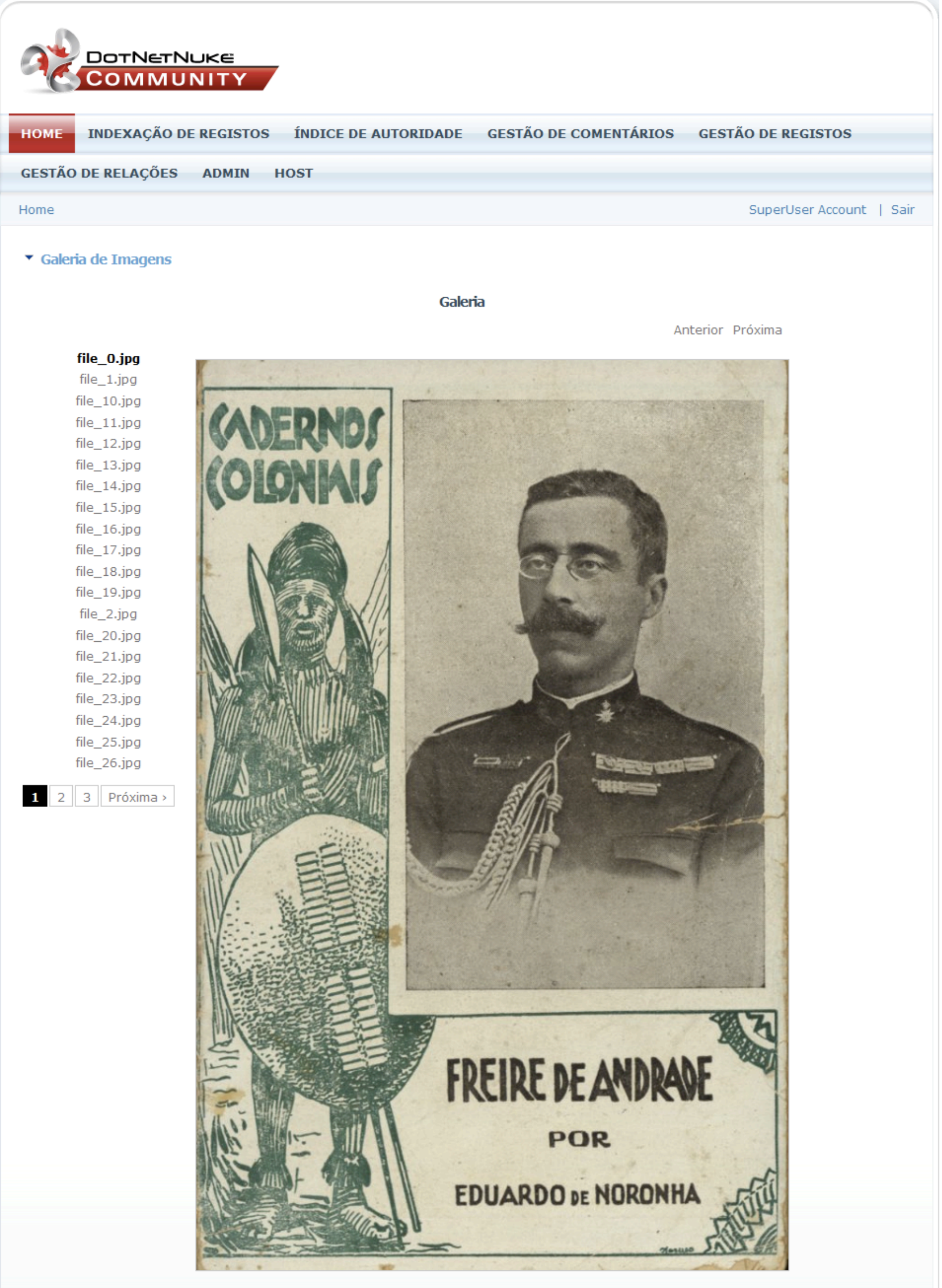
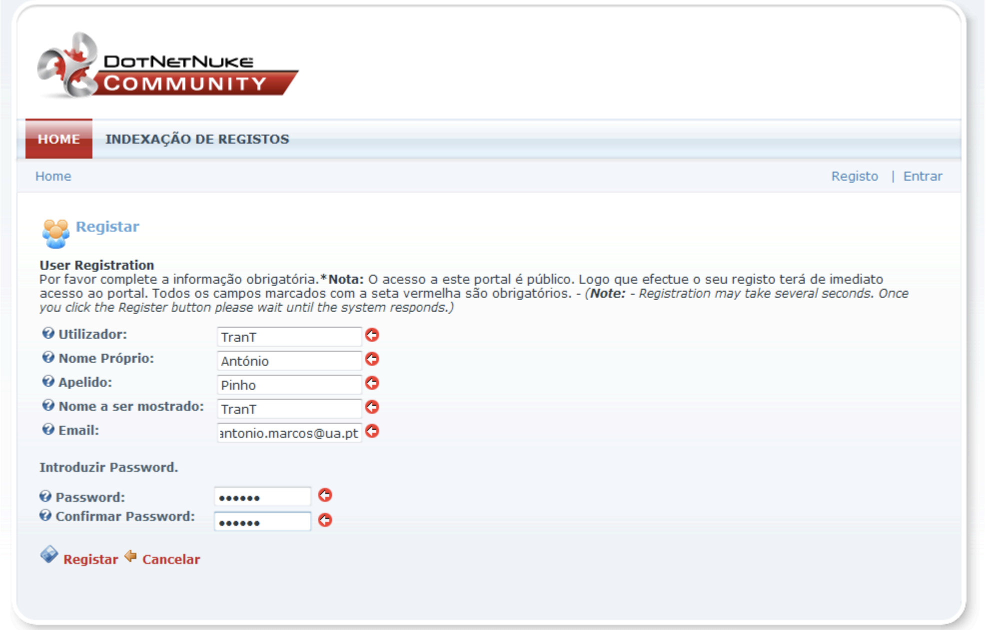


Figure 35 - Scanned documents gallery

4.1.8 User registry

Any user can create an account for using the portal by clicking on the “Registo” link located in the upper right corner of the page. This way it will be able to comment the bibliographical records and check the usernames of all the comments already made. DNN handles all the users’ accounts management.



User Registration
Por favor complete a informação obrigatória.***Nota:** O acesso a este portal é público. Logo que efectue o seu registo terá de imediato acesso ao portal. Todos os campos marcados com a seta vermelha são obrigatórios. - (***Note:** - Registration may take several seconds. Once you click the Register button please wait until the system responds.)*

Utilizador:	TranT
Nome Próprio:	António
Apelido:	Pinho
Nome a ser mostrado:	TranT
Email:	antonio.marcos@ua.pt

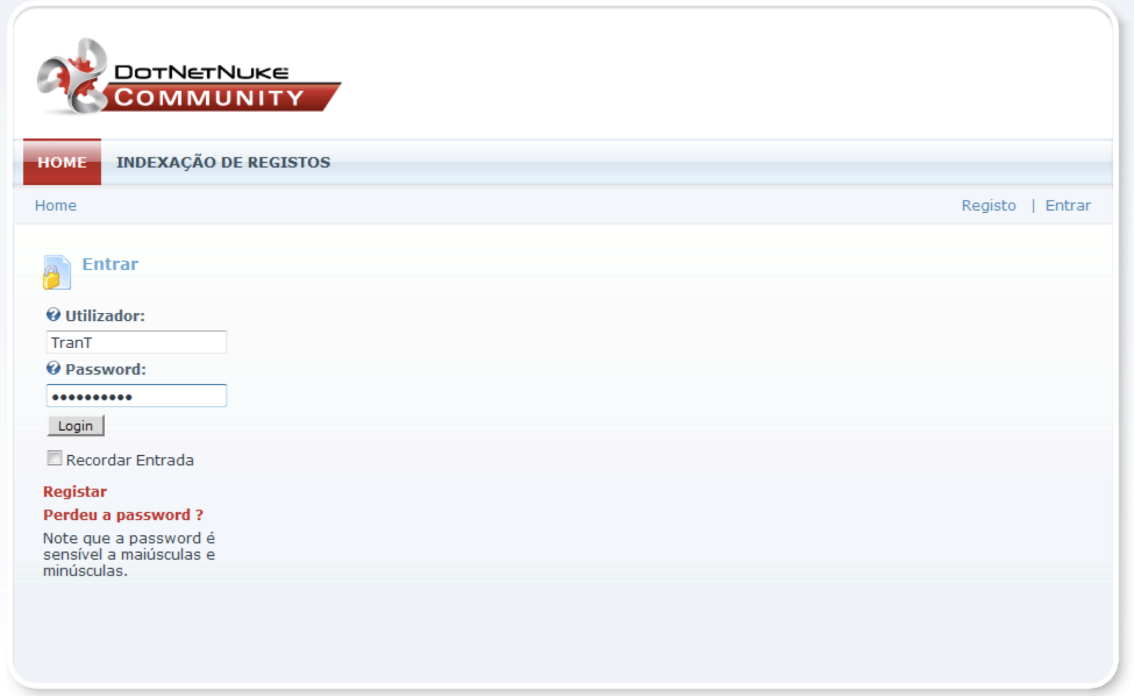
Introduzir Password.

Password:	••••••
Confirmar Password:	••••••

Figure 36 – Create user account

4.1.9 User login

The user may login into its account by clicking on the link “Entrar” located on the upper right corner of the page.



Entrar

Utilizador: TranT

Password: ••••••••••

☐ Recordar Entrada

Note que a password é sensível a maiúsculas e minúsculas.

Figure 37 - User login

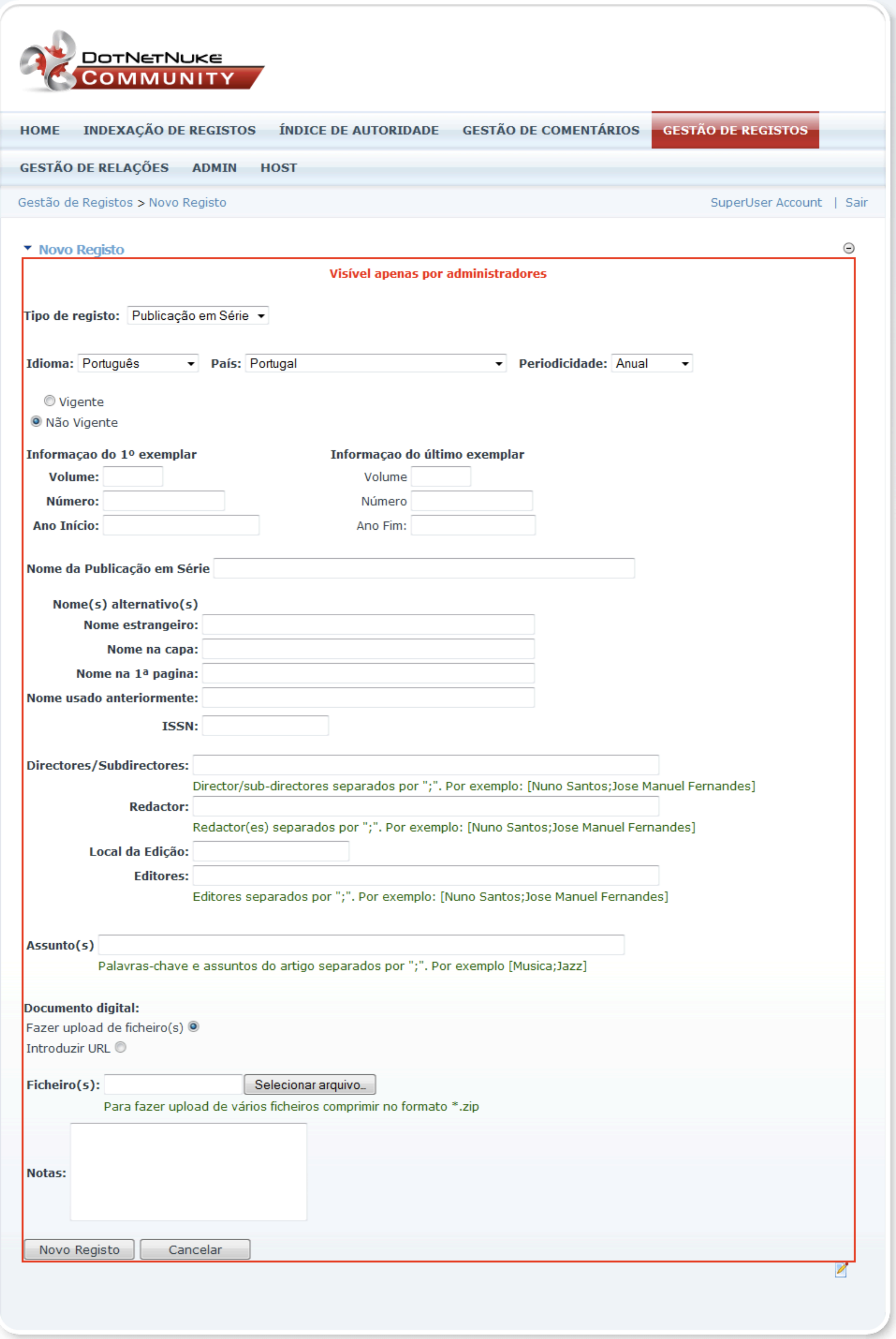
4.2 Back office

The back office developed for this portal will allow authenticated users with administration privileges to perform management operations in the portal. These operations consist on inserting/editing/deleting bibliographic records, authority members, relations and user’s comments.

4.2.1 Bibliographic records’ management page

In this page an administrator may edit/delete a bibliographic record from the portal. Each bibliographic record is listed in a table and each entry has 2 buttons that allow the administrator to edit or delete the corresponding bibliographic record.

This menu item has a sub-menu to a page that will allow the creation of new bibliographic records.



Visível apenas por administradores

Tipo de registo: Publicação em Série

Idioma: Português País: Portugal Periodicidade: Anual

○ Vigente
● Não Vigente

Informaçao do 1º exemplar
Volume:
Número:
Ano Início:

Informaçao do último exemplar
Volume
Número
Ano Fim:

Nome da Publicação em Série

Nome(s) alternativo(s)
Nome estrangeiro:
Nome na capa:
Nome na 1ª pagina:
Nome usado anteriormente:
ISSN:

Directores/Subdirectores:
Director/sub-directores separados por ";". Por exemplo: [Nuno Santos;Jose Manuel Fernandes]
Redactor:
Redactor(es) separados por ";". Por exemplo: [Nuno Santos;Jose Manuel Fernandes]
Local da Edição:
Editores:
Editores separados por ";". Por exemplo: [Nuno Santos;Jose Manuel Fernandes]

Assunto(s)
Palavras-chave e assuntos do artigo separados por ";". Por exemplo [Musica;Jazz]

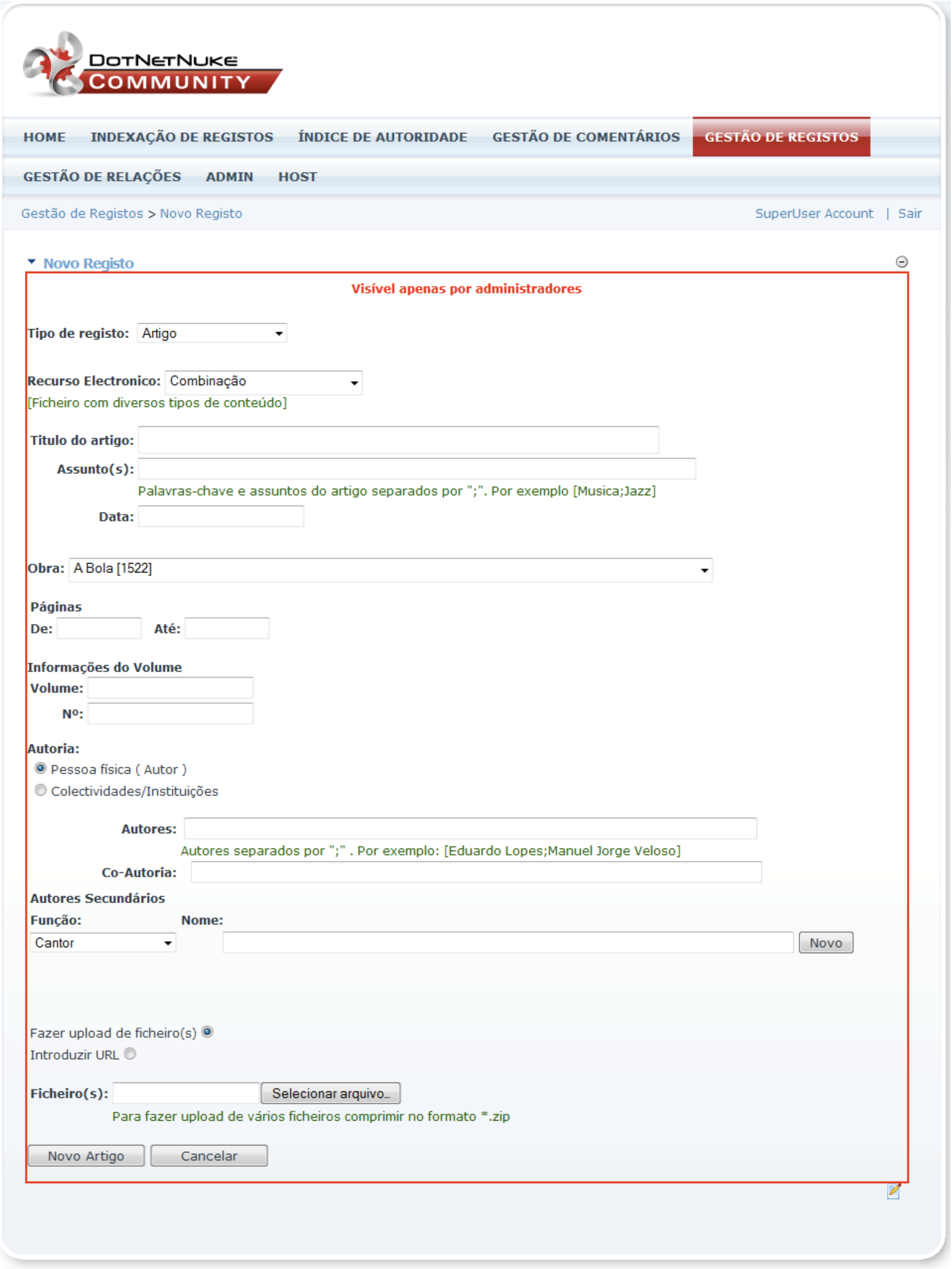
Documento digital:
Fazer upload de ficheiro(s) ●
Introduzir URL ○

Ficheiro(s):
Para fazer upload de vários ficheiros comprimir no formato *.zip

Notas:

Figure 38 - Create new periodical publication

The form presented depends on whether the user choses to create a new periodical publication (Figure 38) or a new article (Figure 39). This can be done by selecting the type of record in the first drop down list.



Novo Registo

Visível apenas por administradores

Tipo de registo: Artigo

Recurso Electronico: Combinação
[Ficheiro com diversos tipos de conteúdo]

Titulo do artigo:

Assunto(s):
Palavras-chave e assuntos do artigo separados por ";". Por exemplo [Musica;Jazz]

Data:

Obra: A Bola [1522]

Páginas
De: Até:

Informações do Volume
Volume:
Nº:

Autoria:
● Pessoa física (Autor)
○ Colectividades/Instituições

Autores:
Autores separados por ";" . Por exemplo: [Eduardo Lopes;Manuel Jorge Veloso]
Co-Autoria:

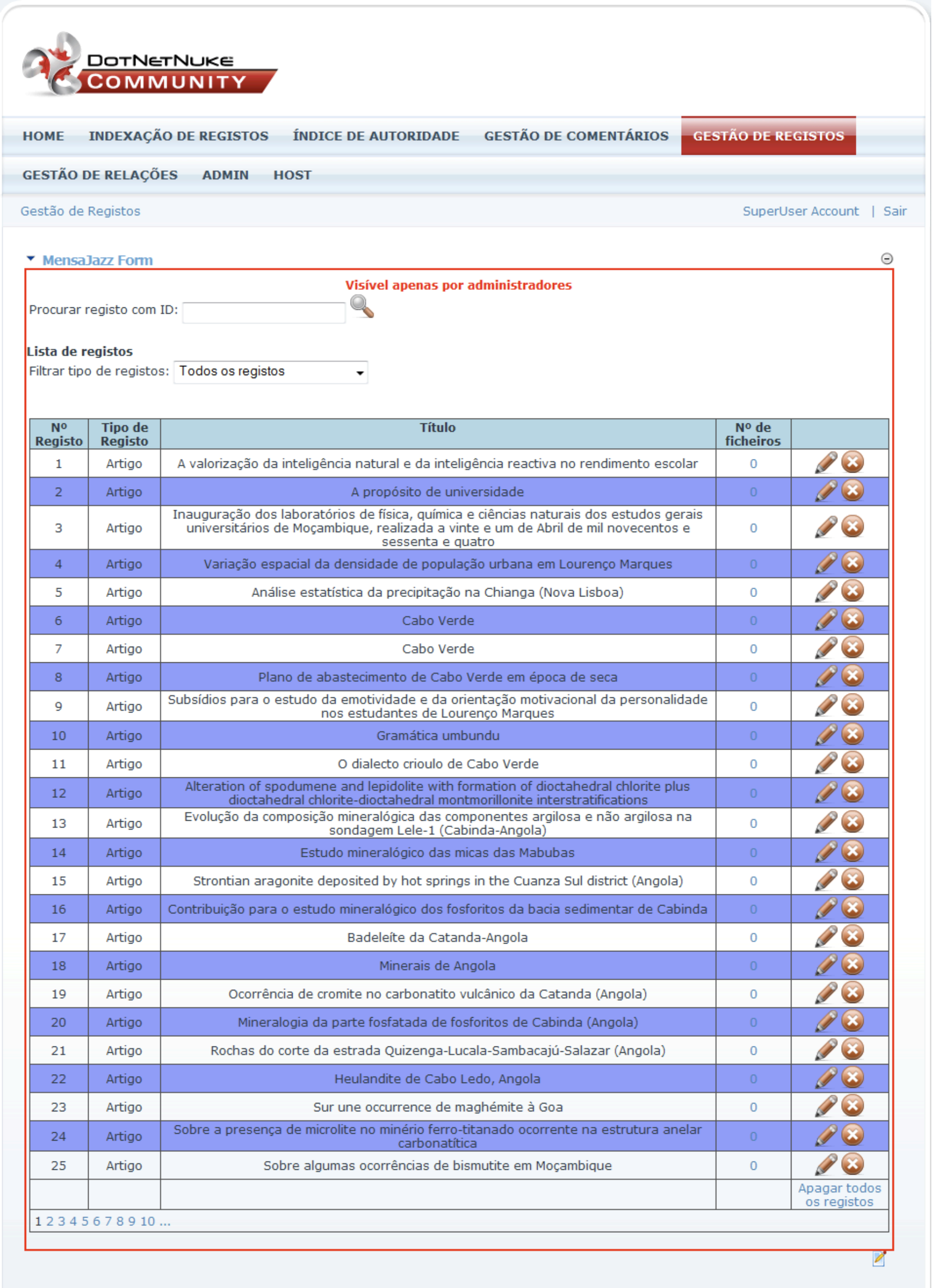
Autores Secundários
Função: Cantor
Nome:

Fazer upload de ficheiro(s) ●
Introduzir URL ○

Ficheiro(s):
Para fazer upload de vários ficheiros comprimir no formato *.zip

Figure 39 - Create new article

The textbox shown in Figure 40 allows the administrator to look for a specific bibliographic record and to either edit or delete it. It can also filter the list of bibliographic records accordingly to their type.



Visível apenas por administradores

Procurar registo com ID:

Lista de registos

Filtrar tipo de registos: Todos os registos

Nº Registo	Tipo de Registo	Título	Nº de ficheiros	
1	Artigo	A valorização da inteligência natural e da inteligência reactiva no rendimento escolar	0	✎ ✖
2	Artigo	A propósito de universidade	0	✎ ✖
3	Artigo	Inauguração dos laboratórios de física, química e ciências naturais dos estudos gerais universitários de Moçambique, realizada a vinte e um de Abril de mil novecentos e sessenta e quatro	0	✎ ✖
4	Artigo	Variação espacial da densidade de população urbana em Lourenço Marques	0	✎ ✖
5	Artigo	Análise estatística da precipitação na Chianga (Nova Lisboa)	0	✎ ✖
6	Artigo	Cabo Verde	0	✎ ✖
7	Artigo	Cabo Verde	0	✎ ✖
8	Artigo	Plano de abastecimento de Cabo Verde em época de seca	0	✎ ✖
9	Artigo	Subsídios para o estudo da emotividade e da orientação motivacional da personalidade nos estudantes de Lourenço Marques	0	✎ ✖
10	Artigo	Gramática umbundu	0	✎ ✖
11	Artigo	O dialecto crioulo de Cabo Verde	0	✎ ✖
12	Artigo	Alteration of spodumene and lepidolite with formation of dioctahedral chlorite plus dioctahedral chlorite-dioctahedral montmorillonite interstratifications	0	✎ ✖
13	Artigo	Evolução da composição mineralógica das componentes argilosa e não argilosa na sondagem Lele-1 (Cabinda-Angola)	0	✎ ✖
14	Artigo	Estudo mineralógico das micas das Mabubas	0	✎ ✖
15	Artigo	Strontian aragonite deposited by hot springs in the Cuanza Sul district (Angola)	0	✎ ✖
16	Artigo	Contribuição para o estudo mineralógico dos fosforitos da bacia sedimentar de Cabinda	0	✎ ✖
17	Artigo	Badeleíte da Catanda-Angola	0	✎ ✖
18	Artigo	Minerais de Angola	0	✎ ✖
19	Artigo	Ocorrência de cromite no carbonatito vulcânico da Catanda (Angola)	0	✎ ✖
20	Artigo	Mineralogia da parte fosfatada de fosforitos de Cabinda (Angola)	0	✎ ✖
21	Artigo	Rochas do corte da estrada Quizenga-Lucala-Sambacajú-Salazar (Angola)	0	✎ ✖
22	Artigo	Heulandite de Cabo Ledo, Angola	0	✎ ✖
23	Artigo	Sur une occurrence de maghémite à Goa	0	✎ ✖
24	Artigo	Sobre a presença de microlite no minério ferro-titanado ocorrente na estrutura anelar carbonatítica	0	✎ ✖
25	Artigo	Sobre algumas ocorrências de bismutite em Moçambique	0	✎ ✖
				Apagar todos os registos

1 2 3 4 5 6 7 8 9 10 ...

Figure 40 – List of all available bibliographic records

4.2.2 Authority members management page

In this page an administrator may edit/delete an authority member from the portal. Each authority member is listed in a table and each entry has 2 buttons that allow the administrator to edit or delete the corresponding authority member.

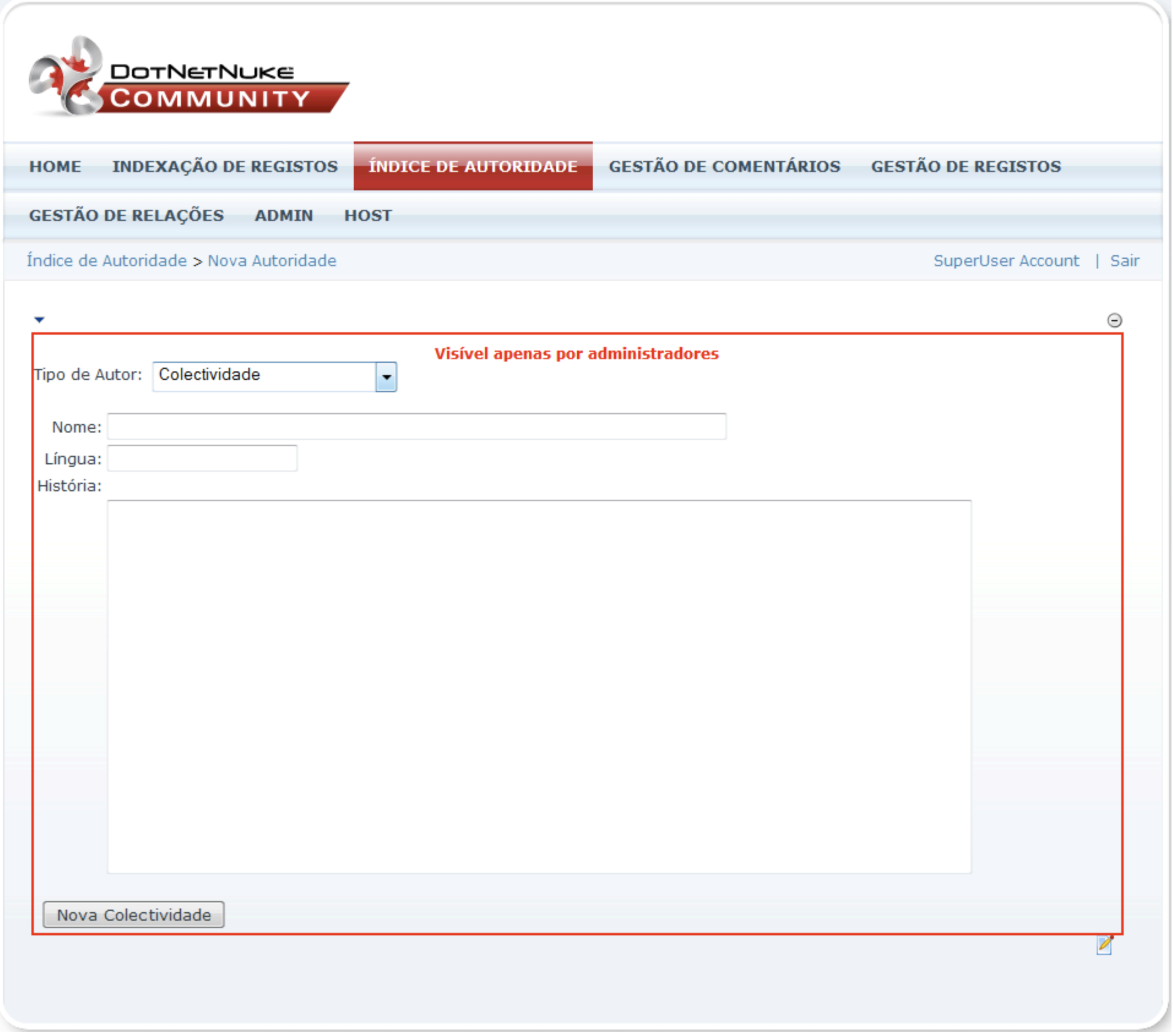
This menu item has a sub-menu to a page that will allow the creation of new authority members.

The authority member created can be of one of three types:

- Colectivity;
- Family;
- Author.

By selecting the type of authority in the dropdown list the administrator is presented with a different form.

When the administrator wants to create a new authority member of the type colectivity the form in Figure 41 is shown.



Tipo de Autor: Colectividade

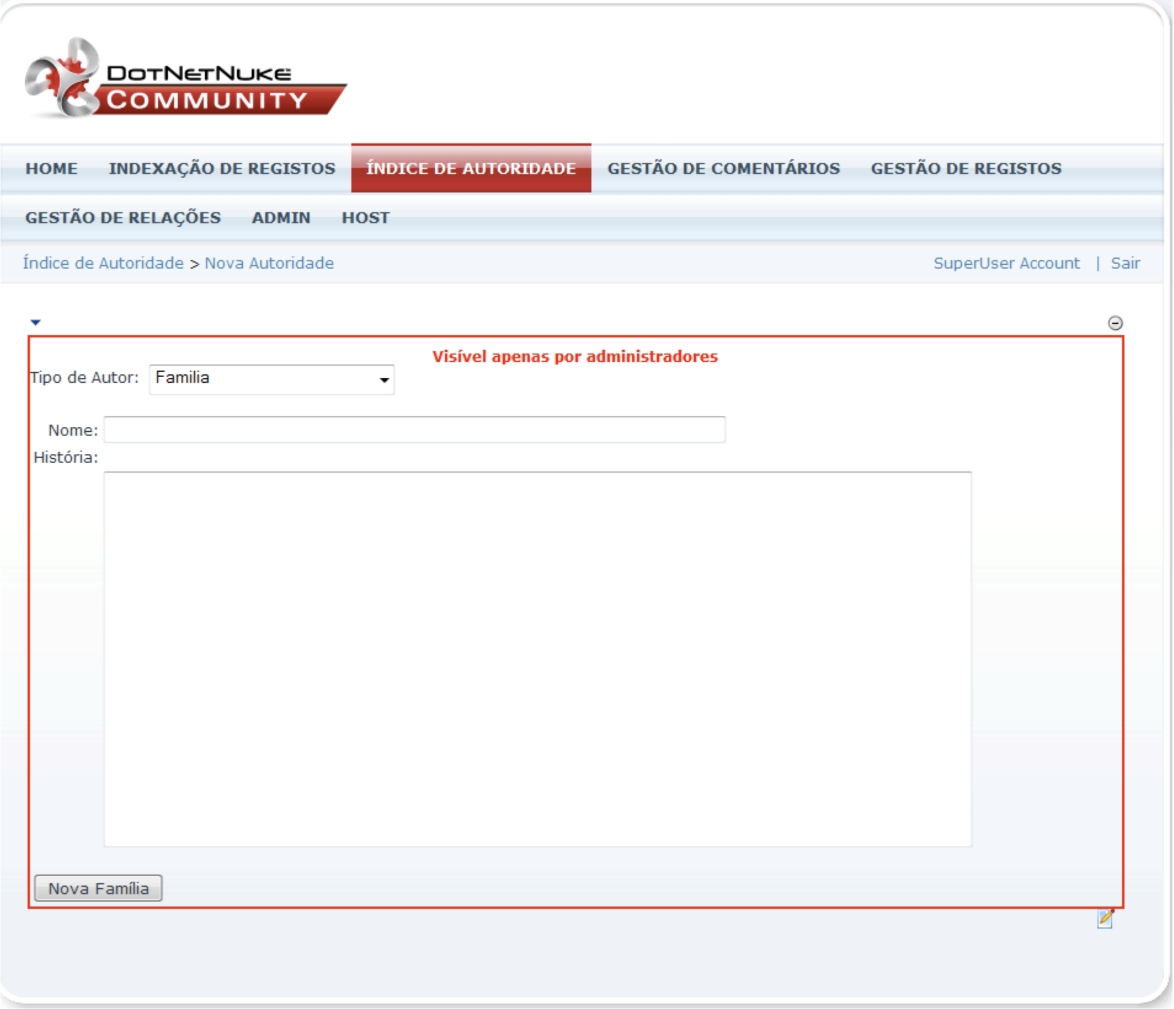
Nome:

Língua:

História:

Figure 41 - Create new authority member of type colectivity

If the administrator choses to create a new authority member of the type Family he is presented with the form from Figure 42.



Visível apenas por administradores

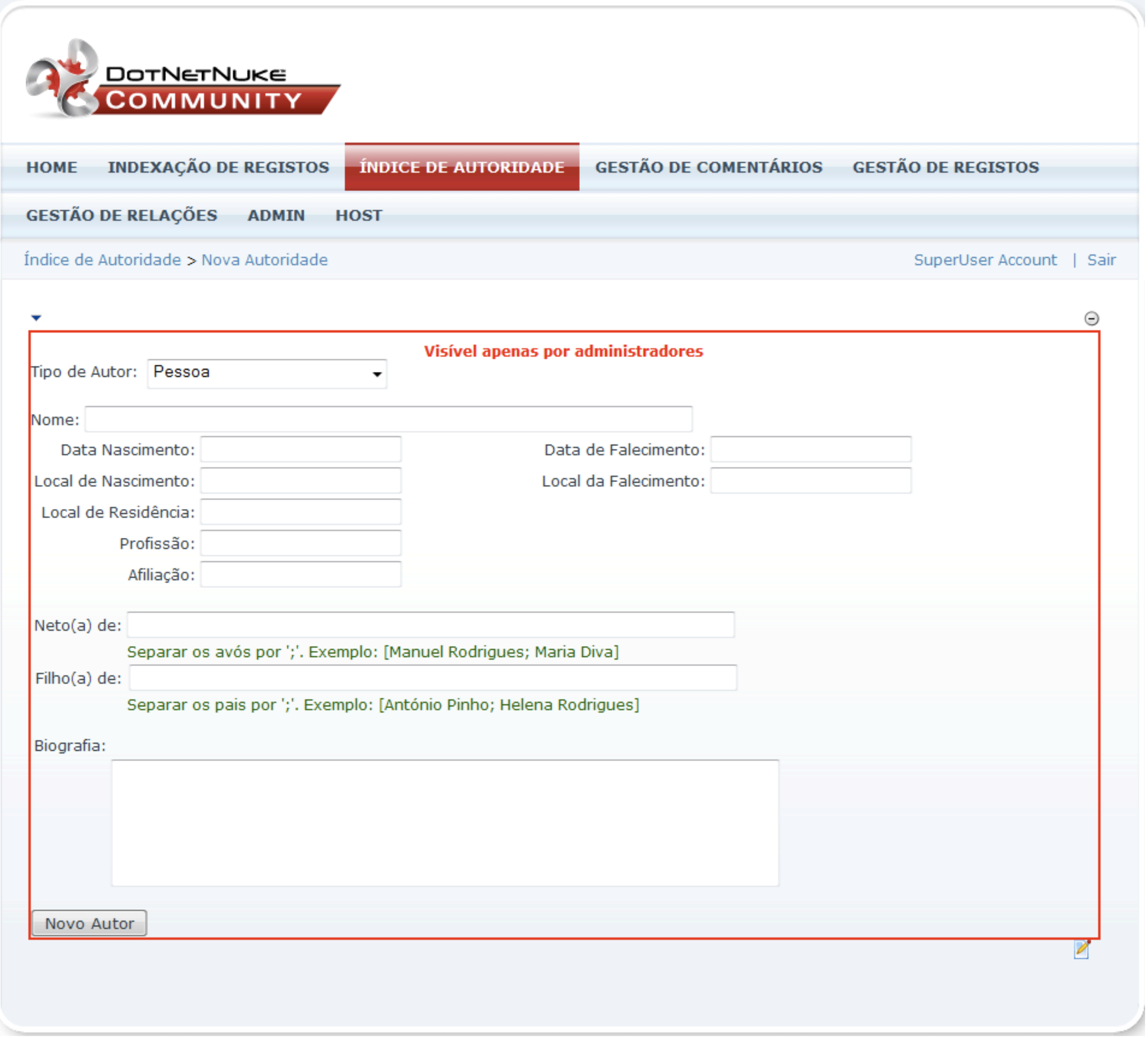
Tipo de Autor: Familia

Nome:

História:

Figure 42 - Create new authority member of type Family

Finally if the administrator wants to create an authority member of type Person he has to fill in the form shown in Figure 43.



Visível apenas por administradores

Tipo de Autor: Pessoa

Nome:

Data Nascimento:

Data de Falecimento:

Local de Nascimento:

Local da Falecimento:

Local de Residência:

Profissão:

Afiliação:

Neto(a) de:

Separar os avós por ';'. Exemplo: [Manuel Rodrigues; Maria Diva]

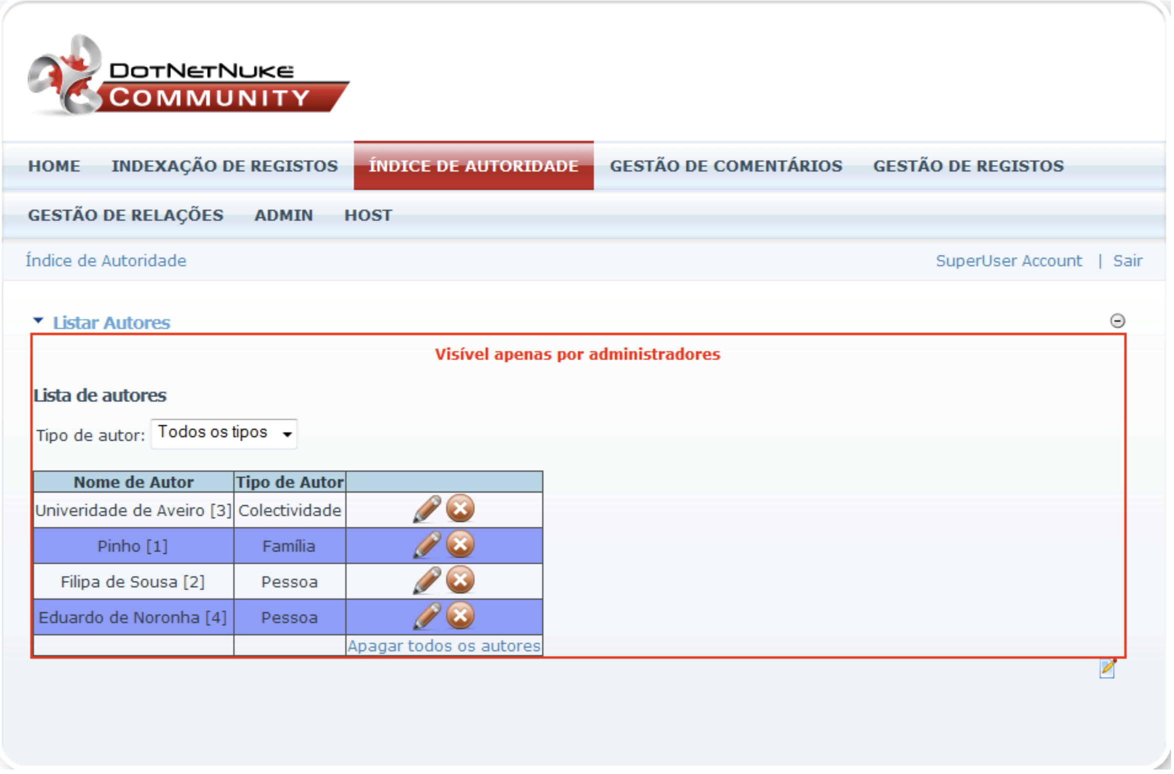
Filho(a) de:

Separar os pais por ';'. Exemplo: [António Pinho; Helena Rodrigues]

Biografia:

Figure 43 - Create new authority member of type Person

In the page shown in Figure 44 the user may edit/delete the selected authority member from the list. The authority members are listed accordingly to the type chosen in the dropdown list.



Visível apenas por administradores

Lista de autores

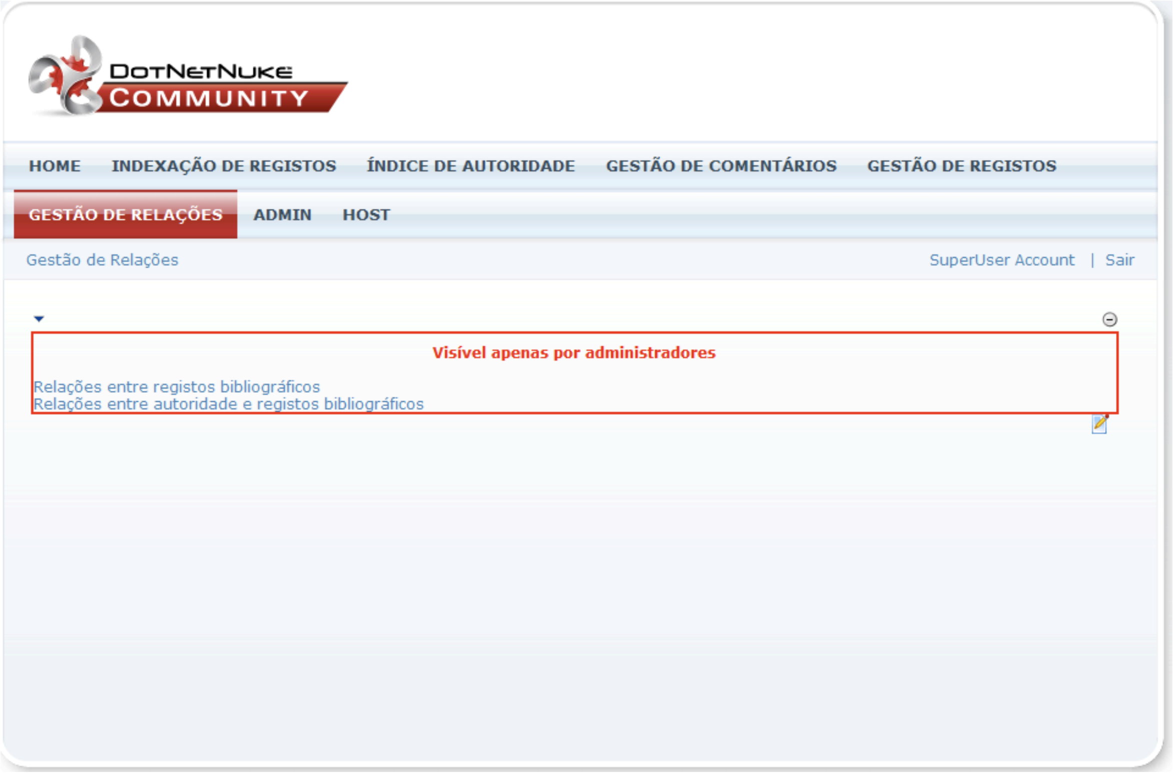
Tipo de autor: Todos os tipos

Nome de Autor	Tipo de Autor	
Univeridade de Aveiro [3]	Colectividade	✎ ✖
Pinho [1]	Família	✎ ✖
Filipa de Sousa [2]	Pessoa	✎ ✖
Eduardo de Noronha [4]	Pessoa	✎ ✖
		Apagar todos os autores

Figure 44 - List of all available authority members

4.2.3 Relation's management page

In this page the administrator has to choose which type of relations it wants to list since there are two different types of relations (Figure 45). After choosing the type of relations to list the administrator has several dropdown lists that will filter out the results so it is easier to find a particular relation.

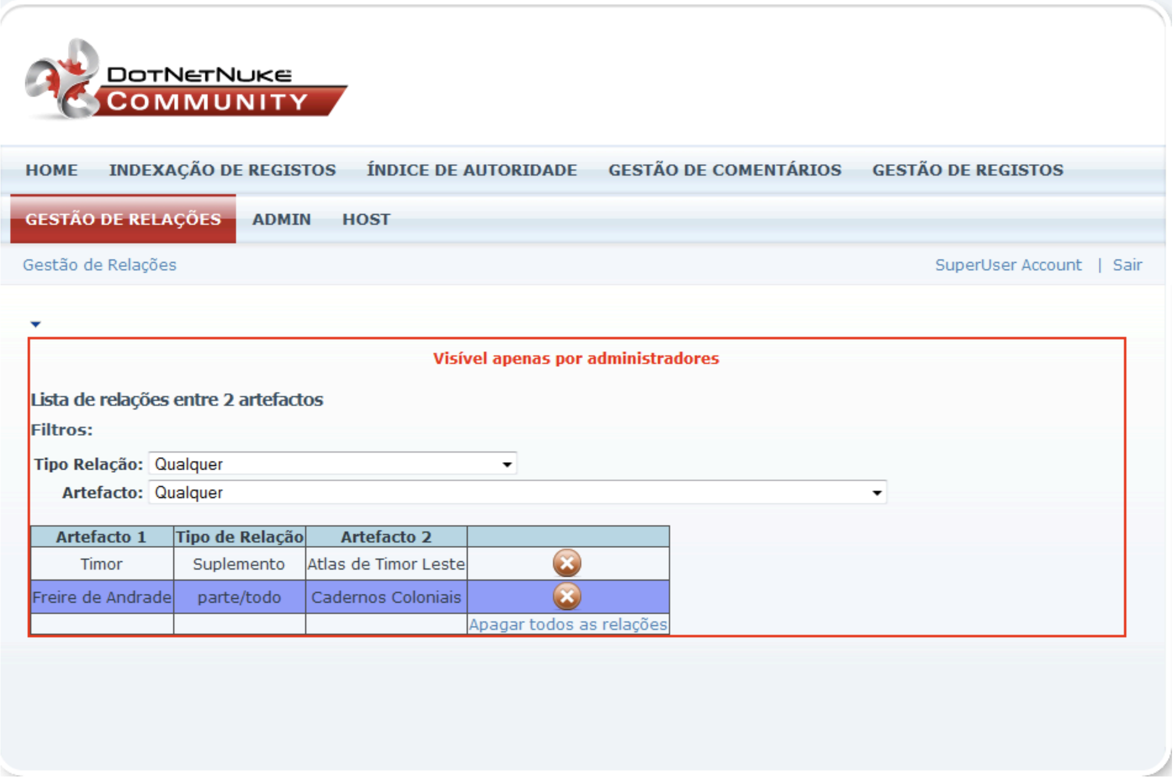


Visível apenas por administradores

Relações entre registos bibliográficos
Relações entre autoridade e registos bibliográficos

Figure 45 - Relations management

When the administrator choses to list the relations between two bibliographic records the page in Figure 46 is displayed.



Visível apenas por administradores

Lista de relações entre 2 artefactos

Filtros:

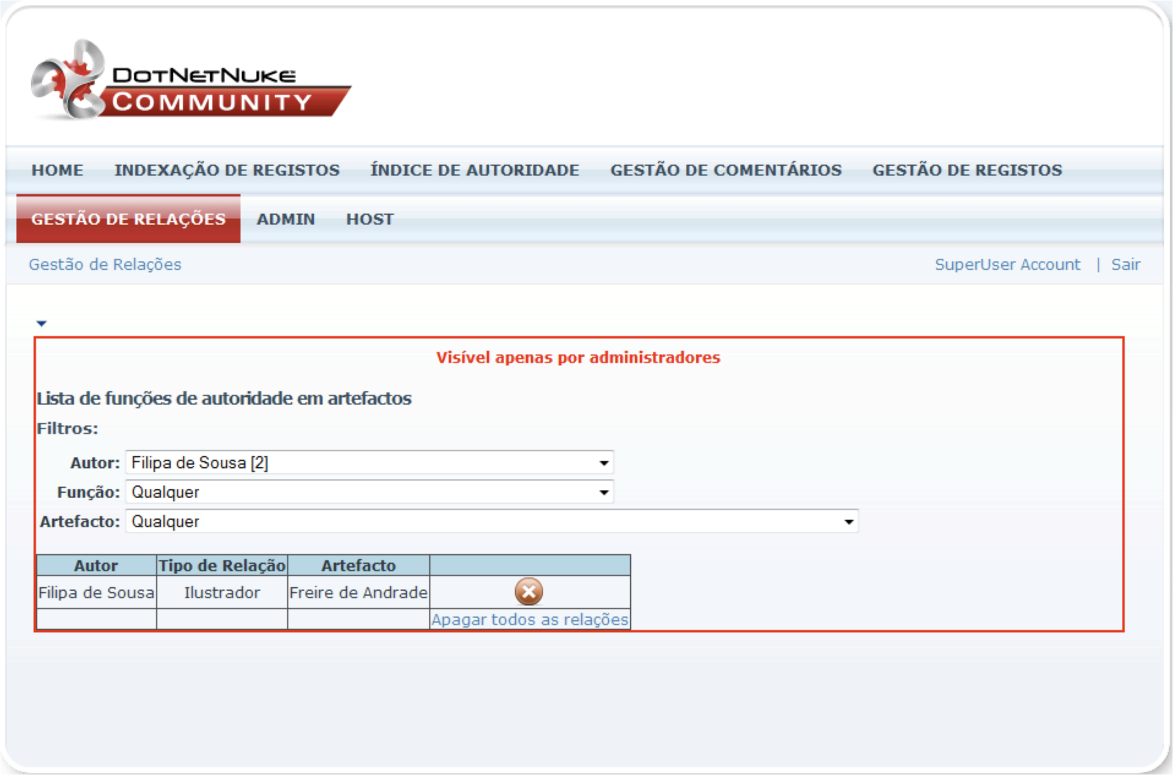
Tipo Relação: Qualquer

Artefacto: Qualquer

Artefacto 1	Tipo de Relação	Artefacto 2	
Timor	Suplemento	Atlas de Timor Leste	☒
Freire de Andrade	parte/todo	Cadernos Coloniais	☒
			Apagar todos as relações

Figure 46 - List of relations between two bibliographic records

On the other hand if the user choses to list the relations between authority members and bibliographic record the page in Figure 47 will be displayed.



Visível apenas por administradores

Lista de funções de autoridade em artefactos

Filtros:

Autor: Filipa de Sousa [2]

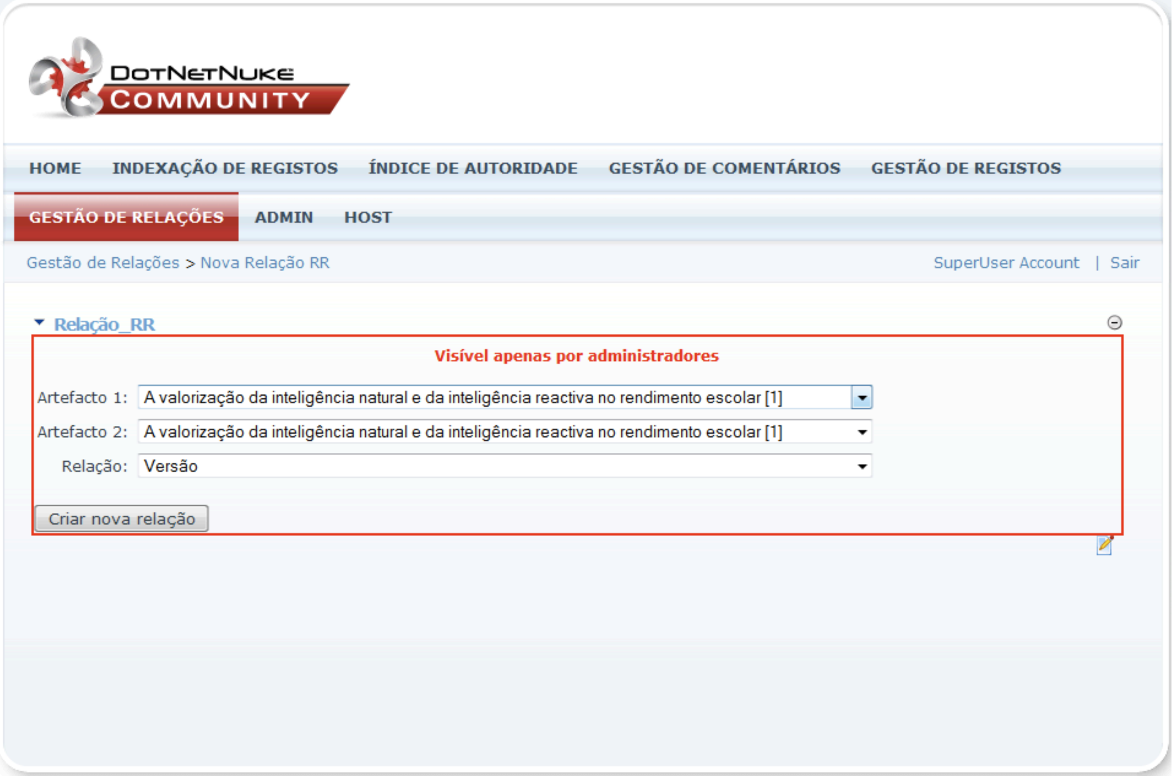
Função: Qualquer

Artefacto: Qualquer

Autor	Tipo de Relação	Artefacto	
Filipa de Sousa	Ilustrador	Freire de Andrade	⊗
			Apagar todos as relações

Figure 47 - List of relations between authority members and bibliographic records

There is a sub-menu for the page that allows the creation of each type of relation. In Figure 48 is shown the page that allows the creation of a relation between two distinct bibliographic records.



Visível apenas por administradores

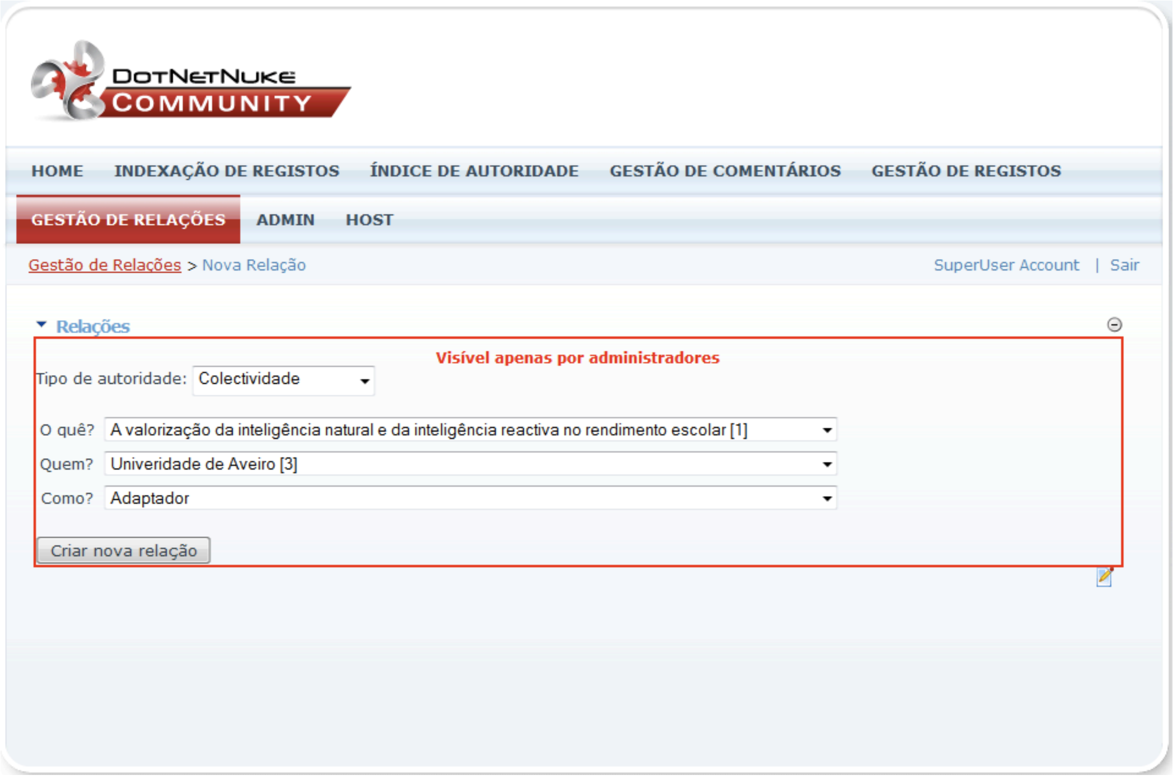
Artefacto 1: A valorização da inteligência natural e da inteligência reactiva no rendimento escolar [1]

Artefacto 2: A valorização da inteligência natural e da inteligência reactiva no rendimento escolar [1]

Relação: Versão

Figure 48 - New relation between bibliographic records

In order to create a relation between an authority member and a bibliographic record the administrator has the page shown in Figure 49.



Visível apenas por administradores

Tipo de autoridade: Colectividade

O quê? A valorização da inteligência natural e da inteligência reactiva no rendimento escolar [1]

Quem? Univeridade de Aveiro [3]

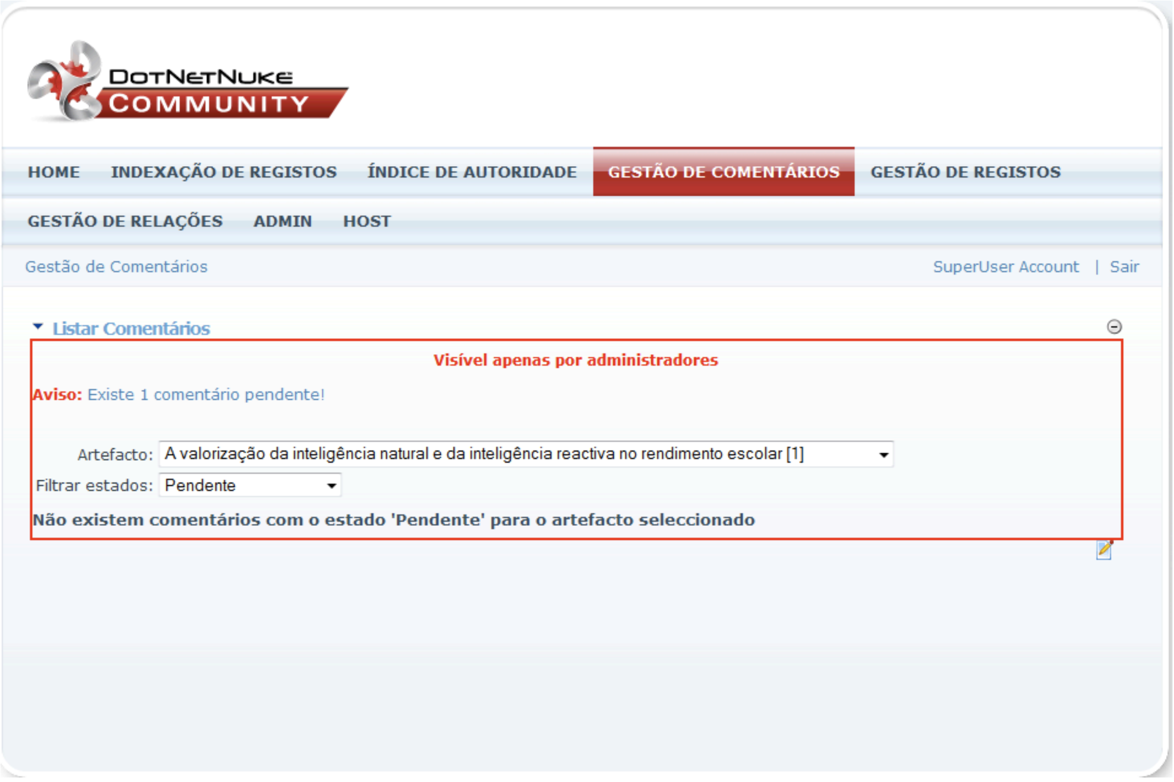
Como? Adaptador

Figure 49 - New relation between an authority member and a bibliographic record

4.2.4 Comments management page

This page will allow the administrator to control all shown comments made by users. Before a comment is shown to all users the administrator must validate it.

Whenever there are new comments to be validated it is shown a warning showing how many new comments exist. If the user clicks on that warning it will be redirected to a page where all these new comments are listed and can be validated (Figure 51).



Visível apenas por administradores

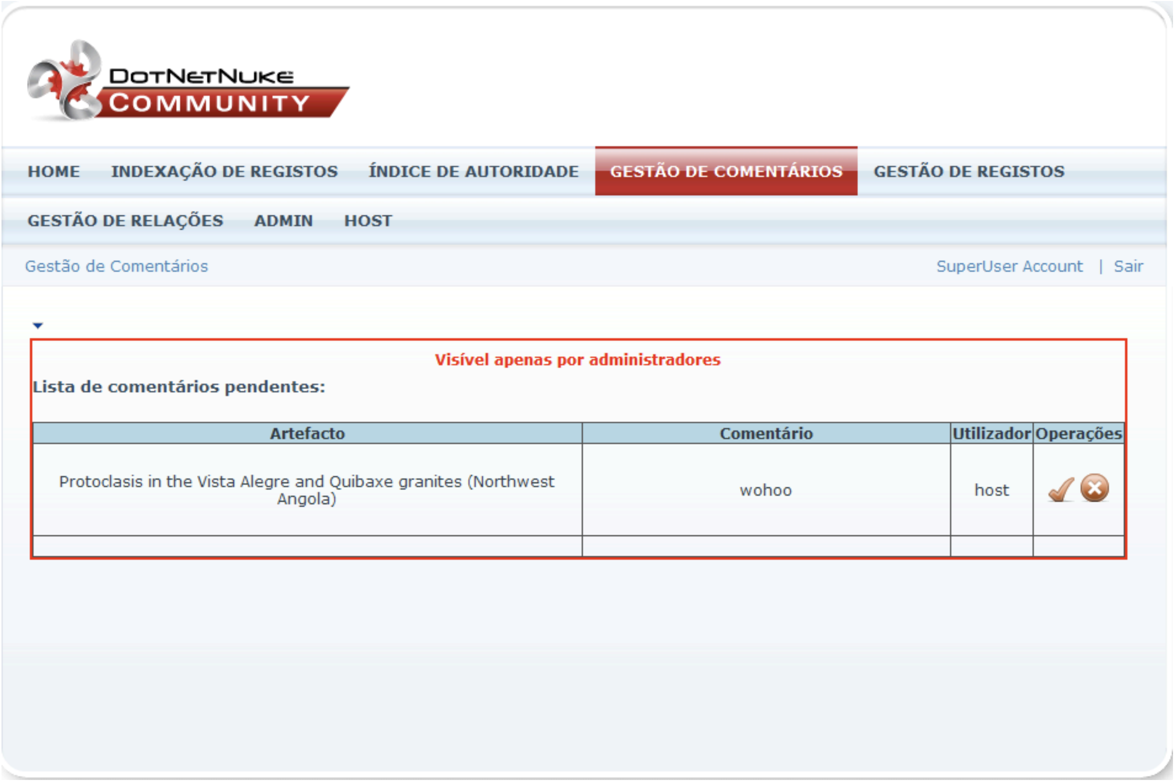
Aviso: Existe 1 comentário pendente!

Artefacto: A valorização da inteligência natural e da inteligência reactiva no rendimento escolar [1]

Filtrar estados: Pendente

Não existem comentários com o estado 'Pendente' para o artefacto seleccionado

Figure 50 - Comments management



Visível apenas por administradores

Lista de comentários pendentes:

Artefacto	Comentário	Utilizador	Operações
Protoclasis in the Vista Alegre and Quibaxe granites (Northwest Angola)	wohoo	host	✓ ☒

Figure 51 - List of new comments

Chapter 5 - **Conclusions**

The “Mensageiros do Jazz” project it’s a multidisciplinary research project which involves several types of media. Due to the project’s coverage its development must be done in a phased way.

The presented Jazz Messengers web portal fulfills the proposed objectives of having an online platform to store Jazz related articles and periodical publications. The development of this work started with some research about bibliographic and cataloguing standards. The first challenge was to find a scheme that allowed the cataloguing, indexing, storage and retrieval of scanned documents related to the Jazz Messengers. The scheme used to catalogue and index the bibliographic records was based on the UNIMARC standard.

UNIMARC was the cataloguing standard chosen to catalogue the bibliographic records because it facilitates the international exchange of data in machine-readable form between national bibliographic agencies and is a format developed by International Federation of Library Associations and Institutions (IFLA) with worldwide recognition. Since it is a very complex and complete standard used for almost any type of resource existent, there was the need to limit the types of resources used. This way it was possible to limit the scope and size of the work developed and having a more concrete specification of what had to be achieved.

The resources chosen in the beginning of the project were articles and periodical publications regarding the history and contribution of the Jazz Messengers. This way it was possible to develop a web platform that allows its users to store these types of documents.

UNIMARC is a very detailed cataloguing standard and if not handled by cataloguing experts may become too much difficult and complex to understand. One of the objectives of this work was to hide this complexity from normal users that don’t have any cataloguing experience. Moreover the system should provide the necessary information to be filled in a common and familiar way so that any user that will be using the system can easily catalogue a scanned document. The GUI has some guidelines that help the user to know how to fill each one of the form’s fields.

With the help of Cristina it was possible to filter out the UNIMARC fields and only include those that are really required and necessary in order to represent the desired resources. The chosen UNIMARC fields were those that are used more often in the type of resources being catalogued. This was done to minimize size of the input form to be filled by users, when they need to catalogue a new resource. This will also avoid having fields that will never be filled by users.

The framework used to develop this work was DNN. It was a real challenge to develop for it since it has a whole different way of building websites. It already has a pre-built set of functionalities like user authentication and some access control allowing the developer to focus on the development of modules that give the website new functionalities. Therefore this will allow and even faster and easy development of modules.

However for some operations DNN can become too much complex not allowing certain things to be performed. One example is not allowing the developer to pass query string parameters to the menus' URLs.

So with this it can be said that DNN is good for developing simple modules but it lacks the right documentation and set of examples for performing more complex and specific tasks. Maybe the paid version of DNN has a broader set of examples and documentation but the open-source version still has much to be improved.

5.1 Future work

The work developed in this thesis is just a starting point for what the Jazz Messengers project is aiming. There is still much to be done in order to make the web portal even more appealing and complete. In a near future the most important things to be done are:

- Creating an access control mechanism that controls, which information from a resource can be shown and what resources a user has access to. This will allow the creation of a set of resources that are only available to authorized users.
- Improving the graphic user interface so it becomes more appealing and easy to use by its users.
- Extending the types of resources supported by the platform in order to allow multimedia files to be stored in the portal.
- Allowing users to use their Facebook login so that they can share their comments in the social network with their friends.

References

[1] "Holocaust Memorial Day: Google's Holocaust Collection Latest in Growing Trend towards Online Archives - Telegraph." Telegraph.co.uk - Telegraph Online, Daily Telegraph and Sunday Telegraph - Telegraph. Web. 13 Mar. 2011. http://www.telegraph.co.uk/news/newstopics/world-war-2/8284070/Holocaust-Memorial-Day-Googles-Holocaust-collection-latest-in-growing-trend-towards-online-archives.html

[2] "IFLA Universal Bibliographic Control and International MARC Core Programme (UBCIM)." IFLA - The Official Website of the International Federation of Library Associations and Institutions. Web. 13 Mar. 2011. <http://archive.ifla.org//VI/3/p1996-1/unimarc.htm>.

[3] Harris, Alan, and Jeff Sanders. “Pro ASP.NET 4 CMS: Advanced Techniques for C# Developers Using the .NET 4 Framework”. New York: Apress, 2010. Print.

[4] R., Walker, “Professional DotNetNuke 5: Open Source Web Application Framework for ASP.NET”. City: John Wiley \& Sons, 2009.

[5] L. Candela et al.: “The DELOS Digital Library Reference Model - Foundations for Digital Libraries”. Version 0.98, February 2008

[6] Fernandes, Marco Paulo dos Santos (2010), “Arquitectura P2P e SOA para bibliotecas digitais”, Tese de Doutoramento, Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

[7] "Knowledge vs. Information." Innovation Zen. Web. 03 Apr. 2011. <http://innovationzen.com/blog/2006/10/02/knowledge-vs-information/>.

[8] "Memórias De África E Do Oriente | Memória De África Introduction Objectives." Memórias De África E Do Oriente | Memória De África Apresentaçoo. Web. 03 Apr. 2011. <http://memoria-africa.ua.pt/introduction/objectives/tabid/152/language/en-US/Default.aspx>.

[9] "UNIMARC Manual : Bibliographic Format 1994 - IFLA Universal Bibliographic Control and International MARC Core Programme (UBCIM)." IFLA - The Official Website of the International Federation of Library Associations and Institutions. Web. 25 Apr. 2011. <http://archive.ifla.org/VI/3/p1996-1/sec-uni.htm>.

[10] "Mensageiros Do Jazz | INET-MD | Instituto De Etnomusicologia - Música E Dança." Na Faculdade... — Universidade Nova De Lisboa. Web. 21 May 2011. <http://www.fcsh.unl.pt/inet/projectos/mensageirosdojazz/pagina.html>.

[11] "Unimarc, Bibliográfico, 1, português, Apêndice C, Código de Função." Escolher Formato. N.p., n.d. Web. 21 May 2011. <http://www.unimarc.info/bibliographic/1/pt/appendixC>.

[12] Santos, Johnny ; Teixeira, Cláudio ; Pinto, Joaquim Sousa - “eABC : um repositório institucional virtual”. In XATA 2005 : XML: Aplicações e Tecnologias Associadas . Braga, PT, Fev. 2005 . p. 40 – 51

[13] "Response.Redirect into a New Window (with Extension Methods) - Infinities Loop." Blogs - ASP.NET Weblogs. Web. 22 May 2011. <http://weblogs.asp.net/infinitiesloop/archive/2007/09/25/response-redirect-into-a-new-window-with-extension-methods.aspx>.

[14] ".NET Zip Library #ziplib (SharpZipLib)." ic#code. N.p., n.d. Web. 23 May 2011. <http://sharpdevelop.net/OpenSource/SharpZipLib/>.

[15] "Galleriffic | A JQuery Plugin for Rendering Fast-performing Photo Galleries." Galleriffic. Web. 25 May 2011. <http://www.twospy.com/galleriffic/>.

[16] “World Wide Web Consortium (W3C)”. Web. 25 May 2011. <http://www.w3.org/>.

[17] "XSL Languages." W3Schools Online Web Tutorials. Web. 25 May 2011. <http://www.w3schools.com/xsl/xsl_languages.asp>.

[18] "XSLT Introduction." W3Schools Online Web Tutorials. Web. 25 May 2011. <http://www.w3schools.com/xsl/xsl_intro.asp>.

[19] XPath." Wikipedia, the Free Encyclopedia. Web. 25 May 2011. <http://en.wikipedia.org/wiki/XPath>.

[20] "Transact-SQL." Wikipedia, the Free Encyclopedia. Web. 25 May 2011. <http://en.wikipedia.org/wiki/Transact-SQL>.

[21] "DNN Architecture Design Diagram." Ultimate DNN Blog Module | Premier CMS Suite for DotNetNuke. Web. 25 May 2011. <http://www.sunblognuke.com/Blog/entryid/106/DNN-Architecture-Design-Diagram.aspx>.

[22] "Unimarc, Bibliográfico, 1, Português, Sumário." Escolher Formato. Web. 26 May 2011. <http://www.unimarc.info/bibliographic/1/pt/summary>.

[23] "Unimarc, Bibliographic, 2.3, English, Summary." Escolher Formato. Web. 26 May 2011. <http://www.unimarc.info/bibliographic/2.3/en/summary>.

[24] "Integrity Constraints." Wikipedia, the Free Encyclopedia. Web. 26 May 2011. <http://en.wikipedia.org/wiki/Integrity_constraints>.

[25] Washington, Michael. “Building websites with DotNetNuke 5”. Birmingham, U.K: Packt Pub, 2010.

[26] Walker, Shaun; Scarbeau, Brian; Schultes Stain and Morgan, Ryan in “Professional DotNetNuke 5 : open source Web application framework for ASP.NET”. Indianapolis, IN: Wiley, 2009.

[27] "What Is the Dublin Core?" Seo Tips,Techniques, Optimization, Analytics & Video Optimize. Web. 28 May 2011. <http://seobabes.blog.com/seo/what-is-the-dublin-core/>.

[28] "DCMI Glossary." DCMI Home: Dublin Core® Metadata Initiative (DCMI). Web. 28 May 2011. <http://dublincore.org/documents/usageguide/glossary.shtml>.

[29] "Dublin Core." Wikipedia, the Free Encyclopedia. Web. 28 May 2011. <http://en.wikipedia.org/wiki/Dublin_Core>.

[30] "DotNetNuke Forums, Find a DNN Forum for DotNetNuke Help." DotNetNuke an ASP.NET Web Content Management Platform, .NET CMS. Web. 29 Apr. 2011. <http://www.dotnetnuke.com/tabid/795/default.aspx>.

[31] "Metadata Principles and Practicalities." D-Lib Magazine. Web. 30 May 2011. <http://www.dlib.org/dlib/april02/weibel/04weibel.html>.

[32] "Dublin Core Metadata Element Set, Version 1.1." DCMI Home: Dublin Core® Metadata Initiative (DCMI). Web. 30 May 2011. <http://dublincore.org/documents/dces/>.

[33] "Multitier Architecture." Wikipedia, the Free Encyclopedia. Web. 30 May 2011. <http://en.wikipedia.org/wiki/Multitier_architecture>.

[34] Mosher, Barb. "Scalability, Analytics, Ease of Use the Focus of DotNetNuke Professional 5.1." Content Management System CMS Reviews, CMS News, CMS Analysis. Web. 31 May 2011. <http://www.cmswire.com/cms/web-cms/scalability-analytics-ease-of-use-the-focus-of-dotnetnuke-professional-51-004832.php>.

[35] Penry, Andrew. "A Comparison of Two Major Dynamic Web Platforms (LAMP vs. WISA)." Shawn Olson Creative Arts - Art, Poetry, Photography. Web. 01 June 2011. <http://www.shawnolson.net/a/302/a-comparison-of-two-major-dynamic-web-platforms-lamp-vs-wisa.html>.

[36] Bromberg, Peter. "Kentico CMS for ASP.NET Sites." EggHeadCafe - The .NET Developer's Portal of Choice. Web. 2 June 2011. <http://www.eggheadcafe.com/tutorials/aspnet/ee551a85-2206-446e-bc7d-c978f60ec671/kentico-cms-for-aspnet-sites.aspx>.

[37] "Orchard CMS: An Overview of the First Development Stages." WebmasterFormat. Web. 2 June 2011. <http://webmasterformat.com/tools/cms/orchard>.

[38] "Review of Kentico CMS – An ASP.NET Platform." WebmasterFormat. Web. 4 June 2011. <http://webmasterformat.com/tools/cms/kentico>.

[39] Robbins, Thom. “Kentico CMS 5 Website Development: Beginner's Guide : Building Professional and Feature-rich Websites with Kentico CMS 5”. Birmingham, U.K.: Packt Pub., 2010. Print.

[40] Mosher, Barb. "Quick Take Review: Umbraco Web Content Management System." Content Management System CMS Reviews, CMS News, CMS Analysis. Web. 5 June 2011. <http://www.cmswire.com/cms/web-cms/quick-take-review-umbraco-web-content-management-system-004296.php>.

[41] Wahlberg, Nik, and Paul Sterling. “Umbraco User's Guide”. Hoboken: John Wiley, 2010. Print.

[42] "Orchard Project." Wikipedia, the Free Encyclopedia. Web. 5 June 2011. <http://en.wikipedia.org/wiki/Orchard_Project>.

[43] "Top 7 ASP.NET CMS Options." WebmasterFormat. Web. 6 June 2011. <http://webmasterformat.com/blog/top-asp-net-cmss>.

[44] "IBuySpy Portal 1.0 Free Scripts Download - The IBuySpy Portal Solution Kit Demonstrates How You Can Use." RocketDownload - Freeware and Shareware Downloads. Web. 1 June 2011. <http://www.rocketdownload.com/scripts/program/ibuyspy-portal-5447.html>.

[45] “Portugal-África, Fundação. Fundação Portugal-África Página Inicial”. Web. 6 June 2011. <http://www.fportugalafrica.pt>.

[46] "Project Directory." CodePlex - Open Source Project Hosting. Web. 7 June 2011. <http://www.codeplex.com/site/search?SortBy=DownloadCount>.

[47] Pedro Almeida, Marco Fernandes, Miguel Alho, Joaquim Arnaldo Martins, and Joaquim Sousa Pinto. 2006. “SInBAD – A Digital Library to Aggregate Multimedia Documents”. In Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services (AICT-ICIW '06). IEEE Computer Society, Washington, DC, USA, 173-.

[50] Eden, Bradford Lee. “Content Management Systems in Libraries: Case Studies”. Lanham, MD: Scarecrow, 2008. Print.

[51] Witten, I. H., David Bainbridge, and David M. Nichols. “How to Build a Digital Library”. Amsterdam: Morgan Kaufmann, 2010. Print.

[52] Evjen, Bill, Scott Hanselman, and Devin Rader. “Professional ASP.NET 3.5 in C♯ and VB. Indianapolis”, IN: Wiley Pub., 2008. Print.

[53] Nagel, Christian. “Professional C# 2008”. Indianapolis, IN: Wiley Pub., 2008. Print.

[54] Gerner, Jason. “Professional LAMP: Linux, Apache, MySQL, and PHP Web Development”. Indianapolis, IN: Wiley Pub., 2006. Print.

[55] "Interview with DotNetNuke Corp. CEO Navin Nagiah | CMS Critic." CMS Critic | CMS News, Reviews, Interviews and Articles. Web. 03 May 2011. <http://www.cmscritic.com/interview-dotnetnuke-corp-ceo-navin-nagiah/>.

[56] "Web Content Management System." Wikipedia, the Free Encyclopedia. Web. 03 May 2011. <http://en.wikipedia.org/wiki/Web_content_management_system>.

Appendix

1. UNIMARC fields used

All the fields that were used to develop this work will be described. The indicators with the character “#” are considered blank which means that it doesn’t affect the field.

All this content is based on the UNIMARC manual [9].

Field 011 - International Standard Serial Number (ISSN)

This field contains the ISSN issued by any ISSN centre.

Indicator 1 - #
Indicator 2 - #

Subfield ISSN $a. - This subfield keeps the ISSN of a periodical publication record. This subfield is optional and repeatable.

Field 100 - General Processing Data

This field has a fixed length of 36 characters with processing information of the current record. This field is mandatory and not repeatable.

Indicator 1 - #
Indicator 2 - #

Subfield $a - General Processing Data

This subfield holds a string with 36 characters where each position has a meaning:

[0-7] – Contains the date of submission of the record into the system. It is in the format YYYYMMDD.

[8] – It holds the type of date according to the following categories:
a – means that the periodical publication is still being published.
b – means that the periodical publication is no longer being published
d – means that the record is an article that was published along with its periodical publication in a fixed date.

[9-12] – These 4 characters will be filled with the year of birth of the periodical publication or article

[13-16] – These 4 characters depend on the type of date from position (8)
a – “9999”
b – year of death of the periodical publication
d – blank “----“

[17-19] – Target audience filled with “k--“ meaning it is adult/serious content.

[20] – Government publication is filled with “y” that stands for non-official publication.

[21] – Modified Record Code is filled with “0” as the record wasn’t modified to transcribe the data as found in the item.

[22-24] – Language of Cataloguing is a three character code that indicates the language used in cataloguing. A list of these codes are supplied by the “Appendix A” of UNIMARC.

[25] - Transliteration Code indicates whether or not an ISO transliteration scheme is used in the record. It is filled with “y” as no transliteration scheme is used on the records.

[26-27] – Character Sets designates the principal graphic character sets used in communication of the record. It is filled with “01”, which means that it is used the basic Latin set of characters.

[28-33] – Additional Character Sets it is the same as the previous ones but its first two characters will be filled with “03” meaning that it is also used the extended Latin set of characters. The other 4 characters won’t be needed and are filled with blank “----“.

[34-35] – Script of Title that indicates which alphabet is used in the title. It is filled with “ba” which stands for Latin.

Field 101 - Language of the item

This field contains the language of the item. It is a mandatory and not repeatable field.

Indicator 1 – “0”: meaning that the item is in its original language and it wasn’t translated.
Indicator 2 - #

Subfield $a – Language of text

This subfield stores the language of the item. It is optional and repeatable.

Field 102 - Country of Publication or Production

This field contains codes for one or more countries of publications. It is mandatory and not repeatable.

Indicator 1 - #
Indicator 2 - #

Subfield $a – Country of Publication

Contains a code representing the country in which the item was published or produced. These codes are listed in the “Appendix B” from the UNIMARC manual [9].

5.1.1.1 Field 135 - Electronic resources: Computer Files

This Field contains information about computer files

Indicator 1 - #
Indicator 2 - #

Subfield $a – Type of electronic resources

This subfield can have the following values:

- ‘a’ - Numeric : data file with numbers
- ‘b’ – Computer Program : a file that can be executed as a computer program
- ‘c’ – Representational: data file with graphic information
- ‘d’ – Text : data file with text
- ‘u’ – Unknown : unknown file content
- ‘v’ – Combination : various types of content
- ‘z’ – Other : another type of content with no code defined

Field 200 - Title and Statement of Responsibility

This field contains the title along with any other title information and statements of responsibility regarding the title. It is mandatory and not repeatable.

Indicator 1 – 1: this title is significant and will be used as an access point to the item.
Indicator 2 – #

Subfield $a – Title Proper

Contains the main title of the item. It is mandatory and not repeatable.

Field 207 - Material Specific Area: Serials Numbering

This field contains numbers and dates of coverage of the first and the last issues of a periodical publication when known. It is optional and repeatable.

Indicator 1 – 0: the numeration is normalized
Indicator 2 – #

Subfield $a - Numbering: Dates and Volume Designations

Contains the numbering and dates.

Field 210 – Publication, Distribution, Etc

This Field contains information on the publication, distribution of the item including dates. It is an optional and not repeatable field.

Indicator 1 – #
Indicator 2 – #

Subfield $a – Publication, Distribution, Etc

This subfield contains the place where the item is distributed and published. It is optional and repeatable.

Subfield $c – Name of the publisher, distributor

This subfield contains the name of the publisher of the item. It is optional and repeatable.

Subfield $d - Date of publication, distribution, etc

This subfield contains the date of birth of the periodical publication and the date of end if it applies. The date of end only appears if the character (8) of the field 100 is “b”. It is optional and repeatable.

5.1.1.2 Field 303 - General Notes Pertaining do Descriptive Information

This field contains any additional information that can be useful to the record. It is optional and repeatable.

Indicator 1 – #
Indicator 2 – #

Subfield $a – Text of note

Contains notes about the record. It is optional and not repeatable.

Field 326 - Frequency Statement Note (Serials)

This field contains a note indicating the frequency that the periodical publication is issued. It is optional and repeatable.

Indicator 1 – #
Indicator 2 – #

Subfield $a – Frequency

Contains a statement indicating the frequency with which a periodical publication is issued

Field 464 - Piece-Analytic Level

This field is used to identify hierarchical links to items at the piece-analytic level. It is made a link to the set level. In this work the set level can be for example a newspaper and the piece-analytic level can be one of its articles. It is optional.

Indicator 1 – #
Indicator 2 – 0: a note for this record is not required

Subfield $t – Title

This subfield contains the relevant title of the item to which the link is being made. It is mandatory and not repeatable

Subfield $p – Physical description

This subfield contains the relevant physical description like the page numbers in which this record is, in the actual periodical publication.

Subfield $v – Volume Number

This subfield contains the volume or number of the periodical publication. It is optional and not repeatable.

Field 510 - Parallel Title Proper

This field contains an alternative title for the record to be used as a note or as an access point. It is optional and repeatable

Indicator 1 – 1: it should be made an access point with this title
Indicator 2 - #

Subfield $a – Parallel Title

This subfield contains the parallel title. It is optional and repeatable.

Field 512 – Cover title

This field contains the title of the cover when it is different from the title in field 200.

Indicator 1 – 1: it should be made an access point with this title

Indicator 2 - #

Subfield $a – Cover title

This subfield contains the cover title. It is optional and repeatable.

Field 514 – Caption title

This field contains the title given at the beginning of the first page of the text when it is different from the title in field 200. It is optional and repeatable.

Indicator 1 – 1: it should be made an access point with this title
Indicator 2 - #

Subfield $a – Caption title

This subfield contains the title as given at the beginning of the first page. It is optional and repeatable.

Field 520 – Former title

This field contains an earlier title of a periodical publication catalogued under a later title. It is treated as an alternative to the title in field 200. It is optional and repeatable.

Indicator 1 – 1: it should be made an access point with this title
Indicator 2 - #

Subfield $a – Former title proper

This subfield contains the title proper of the former title of the periodical publication. It is optional and repeatable.

Field 606 - Topical Name Used as Subject

This field contains a common noun or noun phrase used as a subject heading. It is optional and repeatable.

Indicator 1 – 0: no level is specified all the subjects are on the same level
Indicator 2– #

Subfield $a – Entry Element

Contains the term used as subject. It is optional and repeatable.

Field 700 - Personal Name - Primary Intellectual Responsibility

This field contains the name of the person responsible for the work. It is not repeatable.

Indicator 1 – #
Indicator 2 – 0: name entered in direct order

Subfield $a – Entry Element

Contains the name of the person. It is optional and repeatable

Field 701 - Personal Name - Alternative Intellectual Responsibility

This field contains the name of a person that is considered to have the alternative intellectual responsibility for a work. It is repeatable and repeatable.

Indicator 1 – #
Indicator 2 – 0: name entered in direct order

Subfield $a – Entry Element

Contains the name of the person. It is optional and repeatable.

Field 702 - Personal Name - Secondary Intellectual Responsibility

This field contains the name of a person that is considered to have the secondary intellectual responsibility for a work. It is repeatable

Indicator 1 – #
Indicator 2 – 0: name entered in direct order

Subfield $a – Entry Element

Contains the name of the person. It is optional and repeatable.

Field 710 - Corporate Body Name - Primary Intellectual Responsibility

This field contains the name of the corporate body responsible for the work. It is not repeatable.

Indicator 1 – 0: corporate name
Indicator 2 – 2: name entered in direct order

Subfield $a – Entry Element

Contains the corporate body name. It is optional and repeatable

Field 711 - Corporate Body Name - Alternative Intellectual Responsibility

This field contains the name of the corporate body that is considered to have the alternative intellectual responsibility for a work. It is repeatable.

Indicator 1 – 0: corporate name
Indicator 2 – 2: name entered in direct order

Subfield $a – Entry Element

Contains the corporate body name. It is optional and repeatable.

Field 712 - Corporate Body Name - Secondary Intellectual Responsibility

This field contains the name of the corporate body that is considered to have the secondary intellectual responsibility for a work. It is repeatable

Indicator 1 – 0: corporate name
Indicator 2 – 2: name entered in direct order

Subfield $a – Entry Element

Contains the of the corporate body name. It is optional and repeatable

Field 856 - Electronic Locator and Access

This field contains the required information for locating an electronic item.

Indicator 1 – #
Indicator 2 – #

Subfield $u – Uniform Resource Locator (URL)

Contains the URL to access electronic data.

2. Database Integrity

There are several techniques to ensure data integrity in a database. These techniques are:

- Entity Integrity that ensures that there are no duplicate primary keys;
- Referential Integrity that is specified between two relations and is used to maintain the consistency among tuples in the two relations [24];
- Domain integrity in which every element from a relation should respect the type and restrictions of its corresponding attribute [24];
- Triggers allow the creation of even more complex integrity restrictions at high cost of performance.

The entity integrity is present in every table through the use of primary keys. The referential integrity is guaranteed using foreign keys as the connection between two tables can be validated.

In every database table the domain integrity is assured because every column has a specific data type.

3. Transactions

Along the development of this work transactions were used to guarantee the atomicity of operations. This will ensure that if by any reason there is a failure when executing a query, all the changed made to the database after the beginning of the transaction will be reverted automatically.

Almost every transaction was implemented in the database side where store procedures were specified.

4. Store procedures

It was developed a group of store procedures, which objectives are to create/update/delete data from the database. They are also used to retrieve and search for a specific group of records.

The implementation of these store procedures allows the application to become more independent of the database. With this, changes made to the store procedures won’t be reflected in the application and no changes have to be done to it.

5. Indexes

For testing the performance of the database there were inserted a small amount of bibliographic records (~1500). Initially it was verified that the search for those records was too much slow and optimizations had to be done.

In order to optimize the operations made to the database several indexes were created for some of the database tables. The objective of indexes is to decrease the time needed to access the data belonging to that table.

An index stores the value of the indexed field and a pointer to that record, so when looking for a value instead of having to go though the whole table it will be only necessary to consult its index.

There are two types of indexes, clustered and non-clustered. The clustered indexes reorder the records physically in the hard disk and because of this there can only exist a clustered per table.

Clustered indexes have to be chosen wisely since they reorder data physically. To avoid a constant reordering of data (which is very costly) a new record must be inserted in the table in a sequential way. Because of this the table's primary keys were chosen to have a clustered index as they are generated automatically in an ascending way.

All other indexes were created based on which data fields are accessed more often by either WHERE or JOINs conditions. All the created indexes are going to be described next:

SubFields Table

- CREATE INDEX Idx_SubFields_Code ON SubFields(Code)
- CREATE INDEX Idx_SubFields_RefIDFields ON SubFields(RefIDFields)

Reason: The RefIDField is a foreign key and the Code field is very used and to reduce the execution time an index was created for those fields.

SubField table

- CREATE INDEX Idx_SubField_IDSubField ON SubField(IDSubField) INCLUDE(Value)
- CREATE INDEX Idx_SubField_RefIDSubFields ON SubField(RefIDSubFields)
- CREATE INDEX Idx_SubField_RefIDField ON SubField(RefIDField)

Reason: The Value field of this table is the most used field in the database since it is used to perform searches for all the bibliographic records and to display information regarding a specific bibliographic record. The other fields also have an index because they are very used in JOINs as they are foreign keys.

Fields Table

- CREATE INDEX Idx_Fields_Tag ON Fields(Tag)

Reason: In order to specify a Field its tag must be retrieved. It is a very used field when a search is performed or when the details of a specific bibliographic record are requested.

Field Table

- CREATE INDEX Idx_Field_RefIDFields ON Field(RefIDFields)
- CREATE INDEX Idx_Field_RefIDRecord ON Field(RefIDRecord)

Reason: These fields are both foreign keys that are very used in JOIN queries.

Record Table

- CREATE INDEX Idx_Record_RefIDRecordType ON Record(RefIDRecordType)

Reason: This field is a foreign key and is very used when the bibliographic records are listed accordingly to their record type.

Author_Record_Relation Table

- CREATE INDEX Idx_AuthorRecordRelation_RefIDRecordNumber ON Author_Record_Relation(RefIDRecordNumber)
- CREATE INDEX Idx_AuthorRecordRelation_RefIDPerson ON Author_Record_Relation(RefIDPerson)
- CREATE INDEX Idx_AuthorRecordRelation_RefIDAuthor_Role ON Author_Record_Relation(RefIDAuthor_Role)

Reason: These fields are all foreign keys used when the list of relations between an authority member and a bibliographic record is requested.

Person Table

- CREATE INDEX Idx_Author_Colectivity ON Author_Colectivity(RefIDPerson)
- CREATE INDEX Idx_Author_Family ON Author_Family(RefIDPerson)
- CREATE INDEX Idx_Author_Person ON Author_Person(RefIDPerson)

Reason: All these fields are foreign keys used in JOIN queries. These queries are very used when an user wants to view the all the available information regarding a certain authority member.

Comment Table

- CREATE INDEX Idx_Comment_RefIDRecordNumber ON Comment(RefIDRecordNumber)
- CREATE INDEX Idx_Comment_RefUserID ON Comment(RefUserID)
- CREATE INDEX Idx_Comment_RefIDType_CommentState ON Comment(RefIDType_CommentState)

Reason: when any bibliographic record is requested its comments have to be displayed and because of this all these foreign key fields have indexes associated.

Record_Record_Relation Table

- CREATE INDEX Idx_RecordRecordRelation_RefIDRecordNumber_1 ON Record_Record_Relation(RefIDRecordNumber_1)
- CREATE INDEX Idx_RecordRecordRelation_RefIDRecordNumber_2 ON Record_Record_Relation(RefIDRecordNumber_2)
- CREATE INDEX Idx_RecordRecordRelation_RefIDType_Record_Record ON Record_Record_Relation(RefIDType_Record_Record)

Reason: For gathering information to display the relation between two bibliographic records there were created indexes for each of the foreign keys of this table.

6. Use Cases

All the use cases are presented in this chapter. They were divided in two groups: the use cases from the administrator and the use cases from users (Figure 52 and Figure 53).

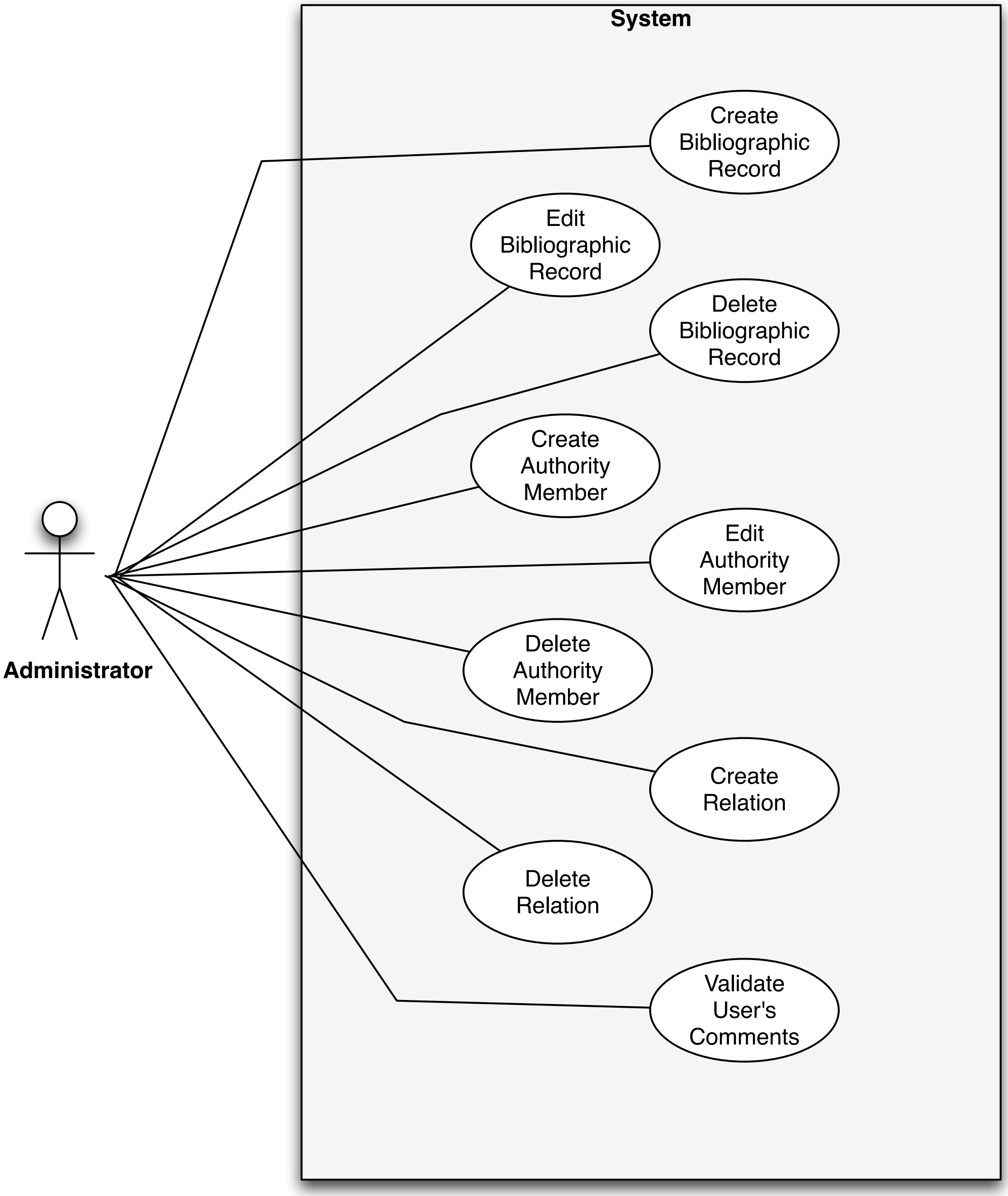


Figure 52 - Administrator use cases

Create Bibliographic Record

This use case describes how an administrator can create a new bibliographic record. There are two types of bibliographic records: articles and periodical publications.

Edit Bibliographic Record

This use case describes how an administrator can edit an existent bibliographic record

Delete Bibliographic

This use case describes how an administrator deletes an existent bibliographic record. When a bibliographic record is deleted all its related content is deleted as well.

Create Authority Member

This use case describes how an administrator creates a new authority member. There are 3 types of authority members: a family, a colectivity and a person.

Edit Authority Member

This use case describes how an administrator edits an existent authority member

Delete Authority Member

This use case describes how an administrator deletes an existent authority member.

Create Relation

This use case describes how an administrator creates a new relation. There are 2 types of relations: a relation between 2 bibliographic records and a relation between an authority member and a bibliographic record.

Delete Relation

This use case describes how an administrator deletes an existent relation.

Validate User’s Comments

This use case describes how an administrator validates the user’s comments. It can validate or refuse a not validated comment.

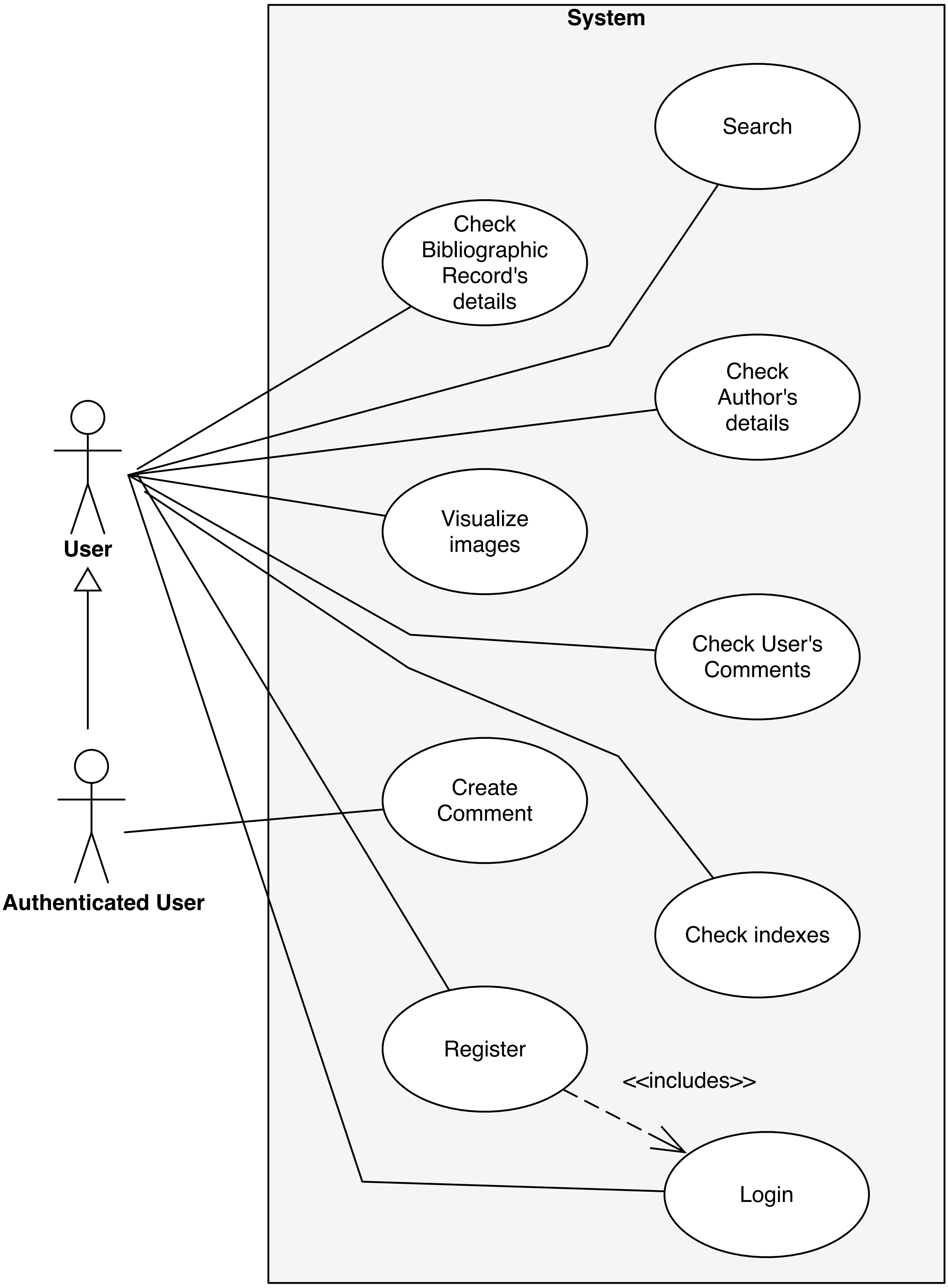


Figure 53 - Users use cases

Search

This use case describes how an user performs a search for bibliographic records.

Check Bibliographic Record’s Details

This use case describes how an user checks a certain bibliographic record details

Visualize Images

This use case describes how an usermeetimeeeee visualizes the images of a bibliographic record.

Check User’s Comments

This use case describes how an user checks the comments made by other users.

Create Comment

This use case describes how an authenticated user creates a comment

Register

This use case describes how an non-authenticated user creates an user account on the portal.

Login

This use case describes how an non-authenticated user logs in the portal.