Oliveira, Fábio André Lopes de
Pereira, Vítor Emanuel de Matos Loureiro da Silva

Development of an automated tool to consolidate information about Portuguese civil parishes

http://hdl.handle.net/11067/1375

Metadados

Data de Publicação 2015-01-20

Resumo The remarkable growth of the Internet accounts for a substantial creation of knowledge, an important asset responsible for the creation of value for nations. Centralized databases have thus increased in size and complexity. Maintaining and updating the information stored in the databases cannot be done efficiently by humans alone; automated tools have been used for quite some time with various degrees of success. One of the first software tools to emerge was the "web crawler", which is the basis...

Palavras Chave Software para computadores - Desenvolvimento, Administração local - Inovações tecnológicas

Tipo article

Revisão de Pares Não

Coleções [ULF-FET] IJEIM, n. 5 (2013)

DEVELOPMENT OF AN AUTOMATED TOOL TO CONSOLIDATE INFORMATION ABOUT PORTUGUESE CIVIL PARISHES

Fábio Oliveira
fab1u@hotmail.com

Vítor Pereira
v.pereira@fam.ulusiada.pt

Universidade Lusíada de Vila Nova de Famalicão, Portugal
Largo Tinoco de Sousa, 4760-108 Vila Nova Famalicão

Abstract: The remarkable growth of the Internet accounts for a substantial creation of knowledge, an important asset responsible for the creation of value for nations. Centralized databases have thus increased in size and complexity. Maintaining and updating the information stored in the databases cannot be done efficiently by humans alone; automated tools have been used for quite some time with various degrees of success. One of the first software tools to emerge was the "web crawler", which is the basis of how search engines work. Another important class of tools, called "internet bots" or simply "bots" (from the word "robot"), is used to help humans manage large quantities of data.

This work describes the development of an automated tool to gather information from various sources (both online and offline) about Portuguese civil parishes ("freguesias" in Portuguese) that can be used, for instance, by marketing companies or by Wikipedia editors to update their respective web pages.

Even though Wikipedia has used bots for over 10 years, the web pages of Portuguese civil parishes are frequently outdated or have insufficient information. In addition, the information that can be used to update these web pages is scattered in various sources and in a format that does not allow an easy comparison between two or more parishes. For instance, an organization may need to compare the distribution of population from various parishes according to the number of people per family, age group or marital status.

The development of the application followed the main steps of Software Engineering namely, requirement specification, application design and implementation, and testing.

The program is able to receive an updated file containing all Portuguese civil parishes and allows the user to select those desired. Furthermore, the user can select offline databases in the form of a spreadsheet according to stipulated parameters. After this the user can still customize an output text, inserting variables that are replaced later by the application in the form of information from the databases, according to each civil parish. Following the compilation of databases in text form, the output for each civil parish is saved as a text file. In order to demonstrate the potential capability of automatic editing of pages in Wikipedia, the application has the capability to preview the text in a Wikipedia test page.

The result of this particular work for a particular case demonstrates the construction of an easy-to-use and practical tool that both basic and advanced users can use to extract information about Portuguese civil parishes.

Key-words: Internet Bots, Wikipedia, Portuguese civil parishes, Update, Software Engineering.

1. Introduction

The incredible growth of the internet is responsible for the creating of knowledge, one of the most important assets for companies and nations. However, with this growth more and more information is scattered around distinct websites and, within the same website, in various files. This dispersed data is therefore more difficult to collect and order, for instance, to update a database, and so it is quite challenging for an individual to gather all this information.

With the economic crisis affecting Portugal, sometimes these realities are left in the background. Due to a recent law (AR, 2012) there will be a reorganization of the territory towards an increase of efficiency and the stimulation of development of local administrations. The Portuguese government plans to reduce the number of parishes by more than a thousand(Silva, 2013), which represent around 25% out of the current 4259 existing parishes(IGP, 2011). A parish ("freguesia" in Portuguese) is a local secondary administration level, just below the municipality. With the aggregation of parishes, large amounts of information will become outdated and sometimes even inconsistent, such as the combination of several parishes into one whose name is an existing one.

Even though a census was recently taken(INE, 2012), part of the data will no longer be entirely accurate with the disappearance of several parishes. Therefore, it is quite difficult to make comparisons between parishes, since the data is scattered across several platforms (both online and offline) and may not be up to date.

Maintaining and updating this data cannot be done efficiently by humans and so there are automated tools to help. To examine the research problem in detail, various tools were studied including one of the first to emerge, the "web crawler", which serves mainly to make a collection of links and, in more advanced tools, a collection of web pages so that an index with every piece of information can be created, which is the basis of how search engines work(Olston & Najork, 2010). Another important class of tools, called "internet bots" or simply "bots" (from the word "robot"), help humans manage large quantities of data. For example, these computational robots are widely applied in Wikipedia since this online encyclopedia is a centralized information system and its constant editing has to be supervised(Wikipedia:Bots, 2013).

The remainder of this study discusses an automated tool to solve the problem with scattered information. An application was developed, based on these internet tools, that is able to collect and organize large amounts of data about Portuguese parishes.

2. Literature Review

Searching for information on the internet is trivial and relatively easy with the help of search engines nowadays, but, in order to make this possible, an intricate world of tools seeking and fetching data exists. In 1993 one of the first web crawlers was implemented, the World Wide Web Wanderer, a search engine with the simple task of progressively scanning the Web and collecting sites(Gray, 1996). The method is highly complicated, but at its core there is a "spider" (a web crawling tool) that crawls the internet like an arachnid crawls on a web. This process starts with the syntactic analysis of a web page (known technically as "HTML parsing") searching links and its division in links that belong to that website (internal links) and links that do not belong to that website (external links). Each internal link is subsequently scanned for more internal links, since these are the ones that contain information relative to the initial website, until the queue of internal links is entirely explored(Heaton, 2002).

From having links to the process of gathering information it's necessary to use a different yet very similar and more general web tool – an Internet Bot – that tries to mimic the human way of searching data but with the speed, availability and response time that only a software tool can achieve. A simple example is making a bot click on a webpage button over and over again, a tedious task that a bot excels at. Wikipedia has used bots for over 10 years, since "Rambot", created one in 2002 with the objective of adding and updating the U.S.A.'s county and city articles using data from its census(User: rambot, 2010). For this information collecting there are two types of bots: the normal bot, very intricate and focused on a specific website interface, and the CatBot that is designed for a large variety of websites(Heaton, 2002).

All internet bots have, at some point, to deal with HTML parsing in order to access the required information. The spiders do that by searching inside the tags that contain the attribute "href", since these have the URLs needed. Internet bots can search anywhere in a HTML document and because of this they have more general use. For instance they can retrieve information from a website table by simply searching for the data contained between the HTML tags of a table.

3. Methodology

In order to create the software tool the plan-driven waterfall model was used. In a plan-driven process, all of the process activities are planned in advance and progress is measured against this plan. The waterfall model is a software process with separate activities where, in principle, an activity has to be finished before moving on to the next activity. It includes four fundamental activities (Sommerville, 2010):

Specification, where the functionality of the software and constraints on its operation are defined;

Design, which involves identifying software components and their

relationship and developing design models;

Implementation, which is the process of converting the design to a computer program;

Testing, to ensure that the software requirements have been fulfilled and, thus, the program does what it is supposed to do.

The next four sections describe in detail these fundamental activities.

3.1 Specification

The software requirement specification is the basis of the project and therefore the choices made are critical to its success. Requirements are divided into functional requirements and non-functional requirements. While functional requirements specify what the system should do, non-functional requirements are constraints on its operation and frequently apply to the whole system(Pressman, 2010). The most important non-functional requirement of the software tool is that it must be an easy-to-use and practical tool for both basic and advanced users. On the other hand, functional requirements include:

- The system shall compile data from online and offline platforms according to each parish;
- For offline usage, data in a spreadsheet format (typically ".xls" files) shall be accepted by the system;
- There must be a customization option for columns in spreadsheet files so that the input can be made even if the files have columns in a different order and number;
- These spreadsheet files shall have a management system, to add or delete and to select specific columns as outputs;
- A selection of Portuguese districts, municipalities and parishes shall exist in a way that allows the users to select various parishes from the same municipality or choose a municipality and, hence, automatically select all parishes belonging to that municipality;
- For online data retrieving, the system shall access official and trustworthy websites;
- Online data should contain contacts for the parishes such as phone numbers, addresses and e-mails when they exist;
- There should be a way to choose between the different databases (spreadsheet files) that will be included in the output;
- There should be a simple text editor to include variables related to databases;
- Most customizations should be automatically saved to improve user efficiency and maintain system integrity, allowing the user to return to the application and pick up where he or she left off;
- Output data shall be displayed on screen;
- Output data shall be exported to text or spreadsheet files;
- Output text should include variables that are replaced by actual data from each parish;
- Output data should be saved in various ways, such as saving all

information to one unique file or to one file for each parish, municipality or district;
- While the application is fetching and organizing data a visual indicator of progress should be used, representing how far the operation has progressed.

Having a good list of requirements does not always mean they remain unaffected until the end of the project since change may happen for various reasons. The following activity, software design, starts to connect the different parts of the system and eventually may identify problems with the requirements. Since the waterfall model is not a linear model, it may involve feedback from one activity to another.

3.2 Design

In order to organize the requirements and ensure they reach a stage at which an executable software is developed, it is essential to know how to solve these problems(Sommerville, 2010).

The user-interface design for this software, shown in Figure 1, is made up of six steps, each one contributing to the final output produced. In the first step the core database is configured in order to specify a key column that is used to create a relationship between all other offline databases selected and managed in step two. Step three uses step one's database to display and create an area where it is possible to choose localities. The next step enables the specific choosing of offline and online databases that will be included in the output, including also if and how the output files are saved. Editing and managing variables within texts is prepared in step five. Finally, in the last step the software assembles all data, displaying it or saving it for the user.

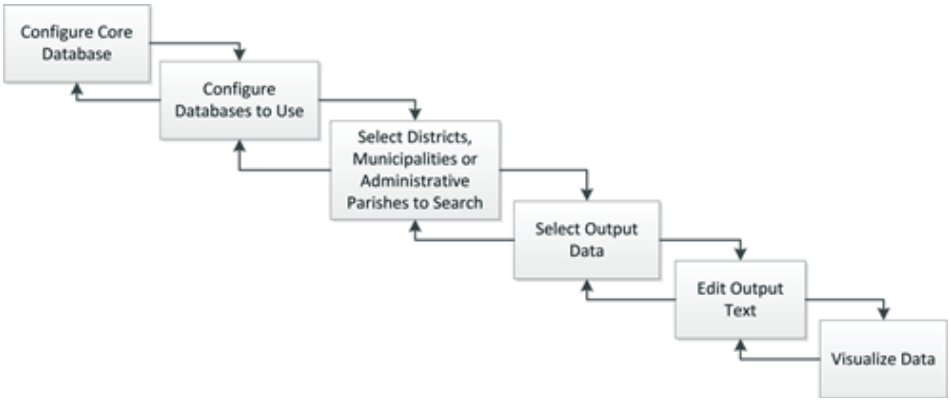


Figure 1 - Waterfall type interface

The customization of the core database is implemented using variables from the spreadsheet itself. The file used for this can be found on the official website of the Portuguese Revenue and Customs Authority (ATA, 2010). As presented in Figure 2, in order for the software to work a core database file (extension ".xls") is loaded. The columns that should be specified are: district, municipality, parish and the parish code. Before going to the next stage these variables are saved in a text file for future access.

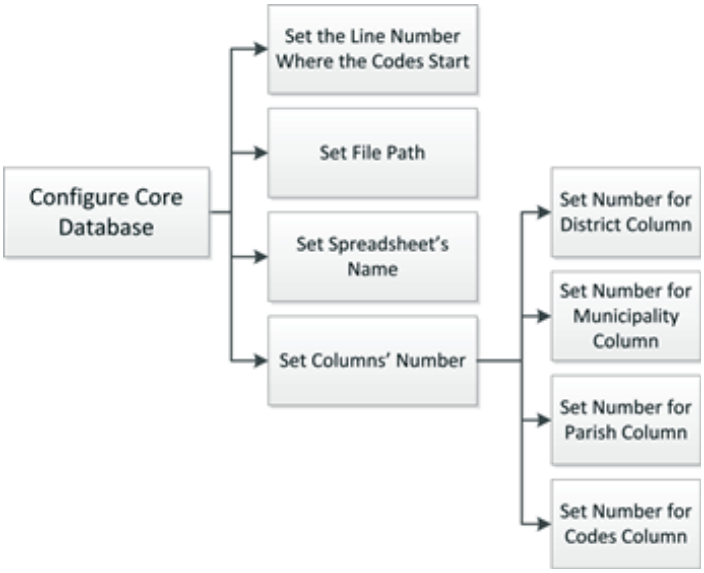


Figure 2 - Core database configuration

Each parish in Portugal has a unique code associated with it. This code is composed of six digits and represents a primary key that the system requires to interconnect databases. The diagram of interaction between databases, shown in Figure 3, represents the requirements for the core database to link to all other offline databases.

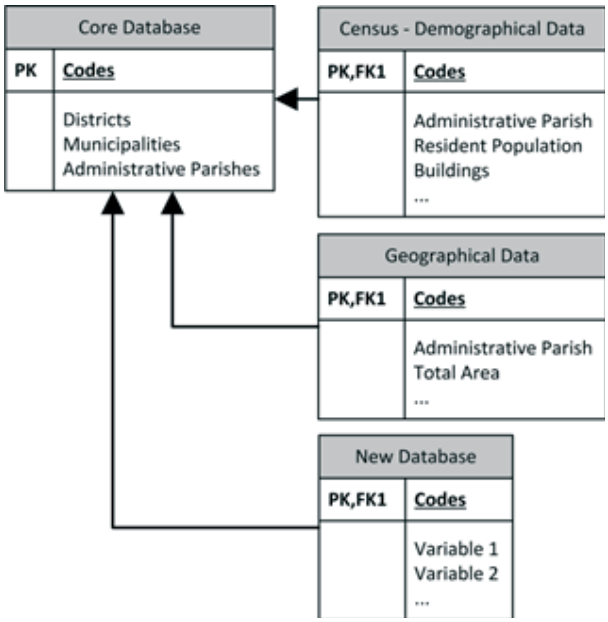


Figure 3 - Diagram of interactions between databases

In order to make sure the application remains supplied with updated files, a database management system was put into practice. This management system was implemented by saving all parameters related to the new files, which are separated by semicolons (similar to a comma-separated values or ".csv" file). These parameters include a database name, file path, sheet name and code number column. To add a new file the variables contained within it are also required. As shown in Figure 4, this occurs by typing in the column names and then selecting the ones that will be part of the output. The files needed for this can be found either on the National Institute of Statistics website (INE, 2012) or the Portuguese Geographical Institute website(IGP, 2012).

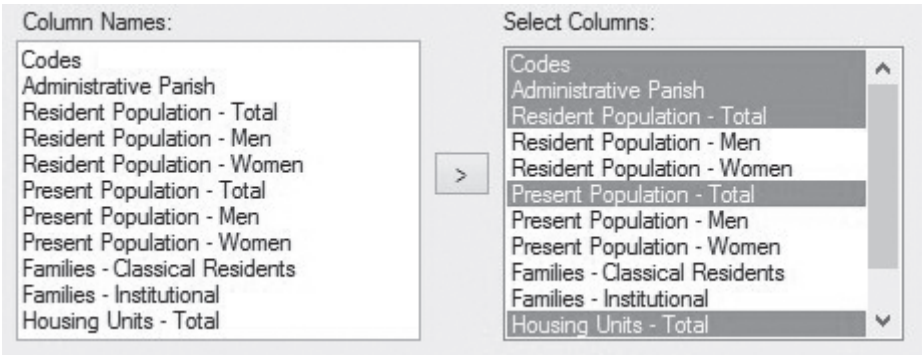


Figure 4 – Inserting and selecting variables by giving names to columns

In order to produce useful results the user needs to select at least one parish. This selection is made by having three lists: districts, municipalities and parishes. The first list (districts) corresponds to the largest geographical areas. By selecting one district the user can then select one or more municipalities belonging to that district and then the same is true for parishes for a given municipality. To accomplish this, the core database is accessed using a technique known as OLEDB: the geographical area selected by the user – district or municipality – is searched in order to filter all municipalities or parishes belonging, respectively, to that district or municipality. The flowchart represented in Figure 5 illustrates the steps that take place when the user selects a district and the software searches all municipalities belonging to that district.

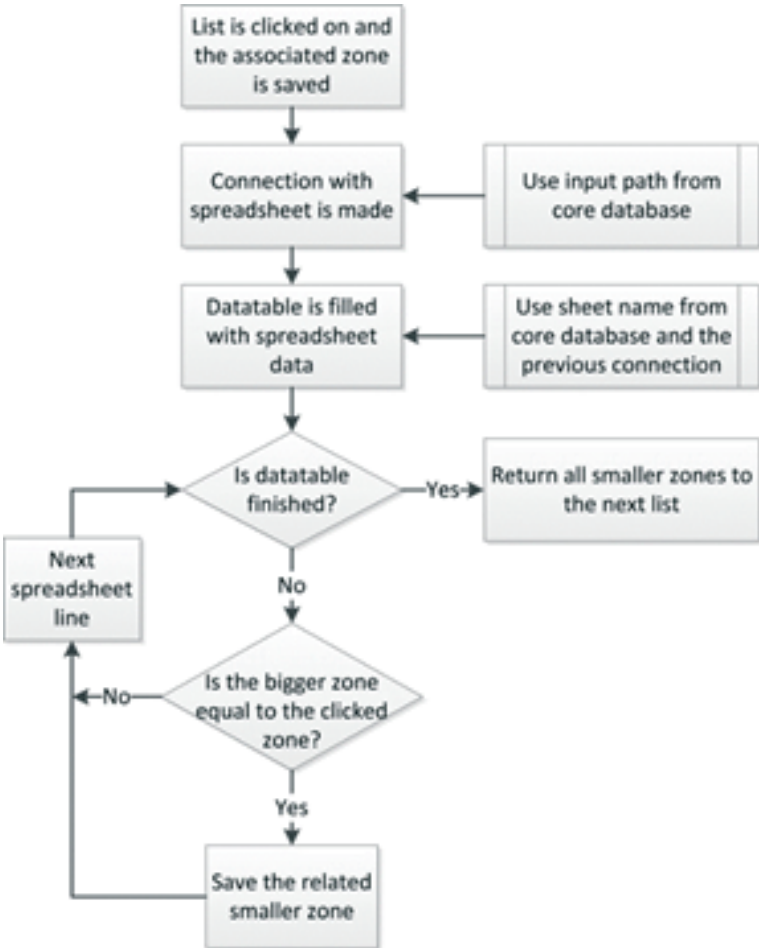


Figure 5 - Logic flowchart for selecting zones from lists

For efficiency purposes, the application uses a technique that involves an adjustment to the links to bypass the use of a web crawler. As shown in Figure 6, a webpage of a parish can be accessed by using the related district, municipality and parish or, alternatively, the parish code, thus simplifying the data search. However, due to the fact that the Portuguese language uses accented characters, for one website (www.anafre.pt), the district, municipality and parish names must go through a process of parsing, since links do not contain spaces or accented characters; more specifically, a conversion from spaces to dashes is needed to access the correct link.

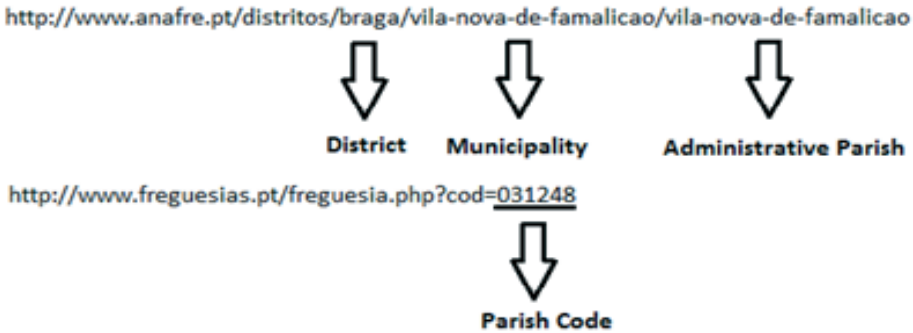


Figure 6 - Techniques of searching for parish data on websites through links

The "Select Output Data" step allows the user to choose which data will be processed, presented and saved; for example, which online data is collected or how final generated data is saved. This is implemented by assembling the individual texts from each parish according to user preference.

The user can create a text template by clicking the variables chosen previously. By doing this, a field or tag with the name of the variable will appear in the text. The application can then change these variables to their respective values from the parishes and a text based on the template is created for each parish.

Lastly, a spreadsheet will appear containing the generated data. This spreadsheet is controlled like a data table where lines correspond to an individual parish and the columns are the attributes selected by the user. As shown in

Figure 7, the final spreadsheet construction starts by retrieving all parish codes that were selected in step three, by receiving the saved database information relative to the selected columns in which the output is made and by adding the column names to the spreadsheet, which are relevant for text editing. The optional header names are the optional names from online databases that by being optional undergo a distinct treatment. After this, the offline databases will be searched for data referred to by the previously collected parish codes, hence adding that data to the final table. The process that follows this is the online search and consists of the previously mentioned data from parishes on websites that can be accessed by changing certain parameters within its links. Before the

information is presented to the user it is organized in a spreadsheet according to the order of the selected output databases.

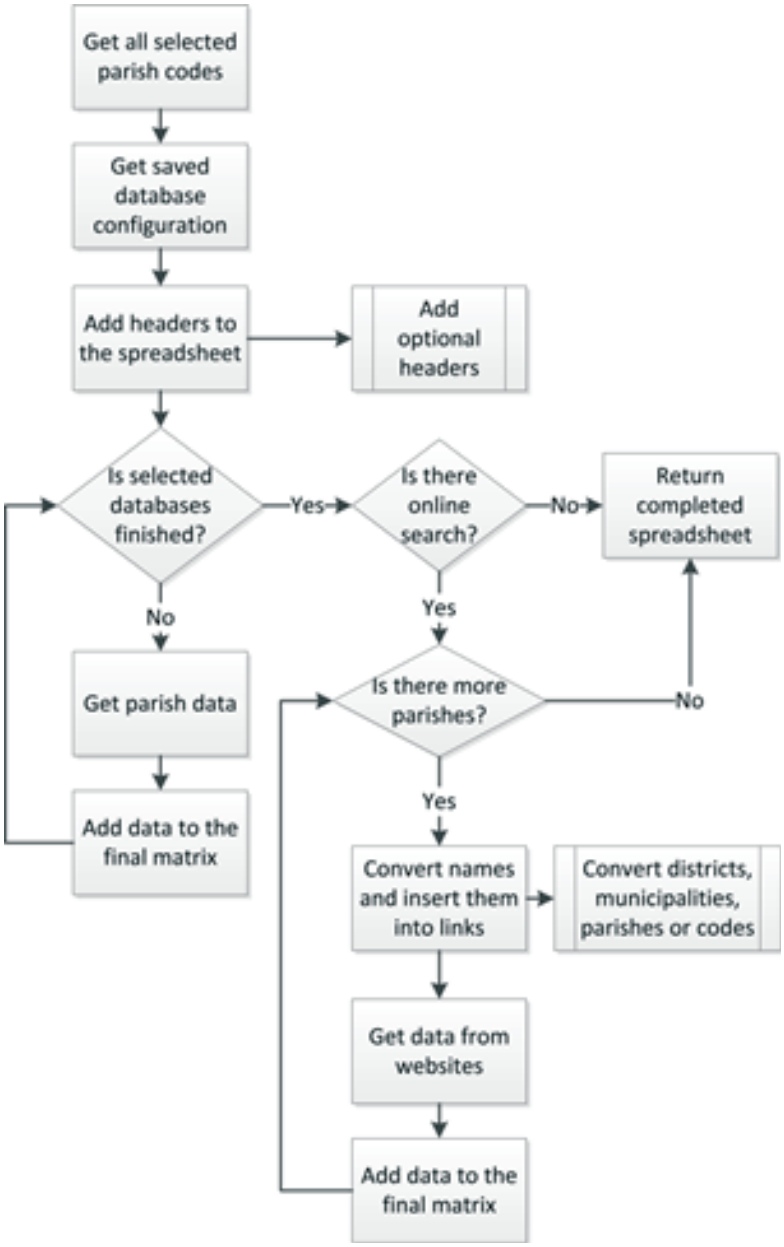


Figure 7 - Logic flowchart of data collecting and organizing

3.3 Implementation

In the implementation phase the design is translated into source code and visually appealing interfaces. The programming language and the integrated development environment used were, respectively, C# and Visual Studio 2010 Express Edition. The interface developed for this software is similar to an installation wizard, where there are a series of steps to achieve the final output. Each step is a customization for the system to work and, to keep it simple, it has variables that can be easily read from the spreadsheet databases; for instance the name of the sheet, the number of a special column or a special line.

One of the most important steps of the program is the selection of parishes. The interface for this step is minimalistic, contributing to the intuitive design, as shown in Figure 8.

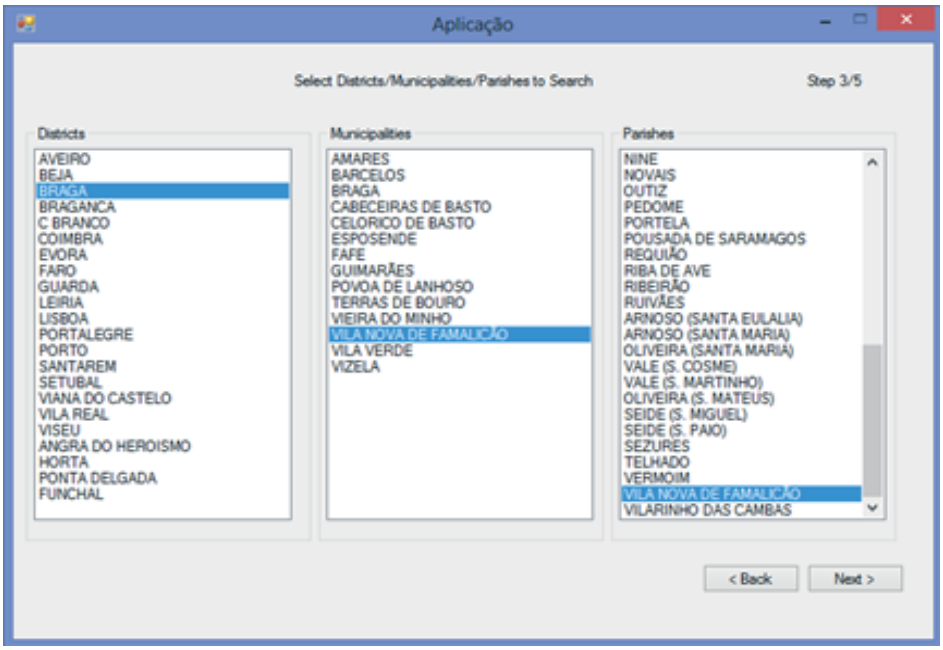
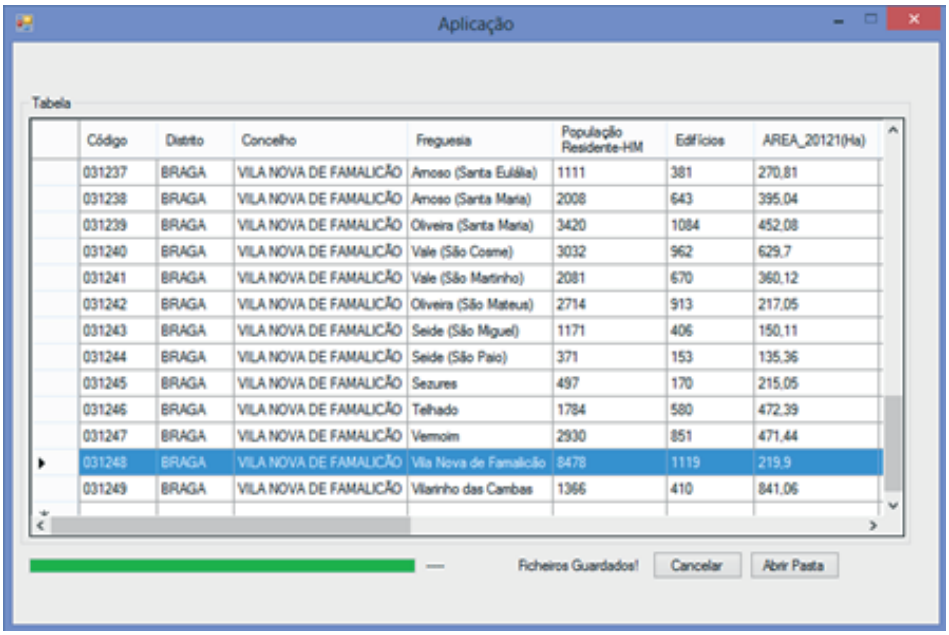


Figure 8 - Selection of districts, municipalities and parishes

For purposes of demonstrating the automation of the system, the functionality of opening a test webpage from Wikipedia with the text that the user inserted was added. This functionality can be used to preview Wikipedia text with its tags before executing and organizing the data. As shown in Figure 9, the system introduces the text with the Wikipedia code for bold or titles, also with the parish data variables ([var:variable_name]) before value replacing and then displays the content of the page to the user.

Figure 9 – Previewing Wikipedia text

The final step displays a table or spreadsheet, as shown in Figure 10, with a progress bar that indicates how long it will take for the system to execute all the tasks. It also has a shortcut that opens the folder containing the generated files, if the user previously wanted to save them. These files contain the text created by the user with the variables already replaced by the data from the spreadsheet for each parish.



Código	Distrito	Concelho	Freguesia	População Residente-HM	Edifícios	AREA_20121(Ha)
031237	BRAGA	VILA NOVA DE FAMALICÃO	Arnoso (Santa Eulália)	1111	381	270,81
031238	BRAGA	VILA NOVA DE FAMALICÃO	Arnoso (Santa Maria)	2008	643	395,04
031239	BRAGA	VILA NOVA DE FAMALICÃO	Oliveira (Santa Maria)	3420	1084	452,08
031240	BRAGA	VILA NOVA DE FAMALICÃO	Vale (São Cosme)	3032	962	629,7
031241	BRAGA	VILA NOVA DE FAMALICÃO	Vale (São Martinho)	2081	670	360,12
031242	BRAGA	VILA NOVA DE FAMALICÃO	Oliveira (São Mateus)	2714	913	217,05
031243	BRAGA	VILA NOVA DE FAMALICÃO	Seide (São Miguel)	1171	406	150,11
031244	BRAGA	VILA NOVA DE FAMALICÃO	Seide (São Paio)	371	153	135,36
031245	BRAGA	VILA NOVA DE FAMALICÃO	Sezures	497	170	215,05
031246	BRAGA	VILA NOVA DE FAMALICÃO	Telhado	1784	580	472,39
031247	BRAGA	VILA NOVA DE FAMALICÃO	Vermoim	2930	851	471,44
031248	BRAGA	VILA NOVA DE FAMALICÃO	Vila Nova de Famalicão	8478	1119	219,9
031249	BRAGA	VILA NOVA DE FAMALICÃO	Vilarinho das Cambas	1366	410	841,06

Figure 10 - Part of the output spreadsheet for the municipality of Vila Nova de Famalicão

3.4 Testing

For the testing phase, two types of testing were applied: unit testing and system testing. Unit testing validates simple entities such as individual functions. System testing is responsible for testing if the application as a whole behaves as planned by the specified requirements(Huizinga & Kolawa, 2007).

The unit testing phase involved both an automated testing framework and manual testing. The automated environment used was NUnit version 2.6.2. The goal of unit testing is, on the one hand, to show that individual functions work as expected and, on the other hand, to make sure there are no defects or problems. Tests started with the analysis of the functions or methods that access the databases. After having this data it is possible to organize a simple test to examine a basic output, for instance, choosing a parish and checking if the parish code is the one stored in the database. After this, the generated data spreadsheet is prepared to receive these outputs and similar tests are implemented in the same way. In order to test the choosing of parishes, the first test was conducted for only one parish, then with two or more. The same technique was true for municipalities. As a municipality is a composed of one or more parishes, part of the code was already tested. A similar method was applied when selecting a district. Particular attention was given to functions accessing online information, such as HTML parsing of web sites.

System testing, as the name implies, involved the creation of a system version

in order to test that integrated system. System testing provided coverage of all the features of the application. More specifically, all options for all the steps of the application were used, thus simulating all possible states and events. Besides using abnormal inputs to check that the most common operations do not crash the application, another type of testing, known as release testing, was applied. Release testing, which aims at making sure the system meets its requirements, focused on an approach called scenario testing. Various typical scenarios were devised and associated test cases developed. For example, selecting one (or a few) and all the parishes from a municipality and saving the resulting data to various files and a single file were scenarios or ways in which the system was used.

4. Conclusion

An automated software tool to gather information from various sources (both online and offline) about Portuguese civil parishes was developed. The development of the application, programmed in C#, followed the four fundamental activities of Software Engineering: requirement specification, design, implementation, and testing.

The program uses a core database with a list of all the districts, municipalities, parishes and parish codes, the latter allowing the tool to extract information from various sources. The user can select a combination of one district, one or more municipalities from one district, or one or more (eventually all) parishes from a municipality. In addition to selecting geographical areas, the user can also control the characteristics or information to summarise, including population related, age group, marital status, building, and household composition. Two of the databases used include key statistics from the 2011 census taken by Instituto Nacional de Estatística (Statistics Portugal) and geographical data by the Portuguese Geographical Institute.

This application can be used, for instance, by marketing companies that need to compare parish-specific information or by Wikipedia editors to update the parishes' web pages. Since the online component is optional, other countries with the same kind of administrative model could use it too, even though this was not confirmed.

The result of this particular work for a particular case demonstrates the construction of an easy-to-use and practical tool that both basic and advanced users can use to extract information about Portuguese civil parishes.

References

AR. (30 de Maio de 2012). Lei n.º 22/2012 de 30 de Maio - Aprova o regime jurídico da reorganização administrativa. *In Diário da República* (pp. 2826-2836).

ATA. (2010). *Código das Freguesias*. Obtido de Portal das Finanças: http://info.

portaldasfinancas.gov.pt/NR/rdonlyres/B365CA49-D07B-4FCD-B9BF-8FADCA0EF528/0/coddistconcfreg.xls

Gray, M. (1996). *Credits and Background*. Obtido de Internet growth and statistics: http://www.mit.edu/people/mkgray/net/background.html

Heaton, J. (2002). *Programming Spiders, Bots, and Agregators in Java*. Sybex.

Huizinga, D., & Kolawa, A. (2007). *Automated Defect Prevention: Best Practices in Software Management*. Wiley-IEEE Computer Society Pr.

IGP. (2011). *Carta Administrativa Oficial de Portugal*. Obtido de Instituto Geográfico Português: http://www.igeo.pt/produtos/cadastro/caop/download/Dados_CAOP2011.pdf

IGP. (2012). *Área das Freguesias, Municípios e Distritos da CAOP2012*. Obtido de Instituto Geográfico Português:

http://www.igeo.pt/produtos/cadastro/caop/download/Areas_Freg_Mun_Dist_CAOP20121.zip

INE. (2012). *Censos 2011*. Obtido de Instituto Nacional de Estatística: http://censos.ine.pt/

Olston, C., & Najork, M. (2010). *Web Crawling: Foundations and Trends in Information Retrieval*. Now Publishers Inc.

Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach, Seventh Edition*. McGraw-Hill.

Silva, A. C. (16 de Janeiro de 2013). *Mensagem do Presidente da República à Assembleia da República sobre a Reorganização Administrativa do Território das Freguesias*. Obtido de Página Oficial da Presidência da República Portuguesa: http://www.presidencia.pt/?idc=10&idi=71113

Sommerville, I. (2010). *Software Engineering Ninth Edition*. Addison-Wesley.

User: rambot. (2010). Obtido de http://en.wikipedia.org/wiki/User:Rambot

Wikipedia: Bots. (18 de April de 2013). Obtido de Wikipedia: http://en.wikipedia.org/wiki/Wikipedia:Bots