

**An application to store and explore in a database the emigration
documents**

Ricardo Giuliani Martini

TECHNICAL REPORT

Doctoral Program in computer science (PDInf)
Supervisor: Prof. PhD. Pedro Rangel Henriques
Co-supervisor: Prof. PhD. Giovanni Rubert Librelotto

Braga - Portugal, september 2014

Abstract

This document presents an web-based application to store in a database ancient emigration records belonging to municipal archives. This system serves to populate and manage database records with data extracted from passport application forms for emigration, aiming to provide relevant information to people who are interested. In addition to presenting the application, it will also be described as these application forms are organized and stored currently, to demonstrate the real need of the application.

This application was built with the purpose of storing physical documents digitally to preserve people's memory and citizens cultural heritage. It can be said that this system helps, on one hand, people that works with the emigration documents in the Archives (making easier the information storage and manipulation) and, on the other hand, also helps those History Researches or even end-users who want to extract information and know more about the emigration documents.

The application was developed using Javascript jQuery API, and PHP programming language with the aid of the Yii Framework. Yii is a Model-View-Controller framework (MVC-based) that assists PHP developers with some tools to facilitate the development process like the creation of database models (models), forms (views) and actions (controllers).

Keywords: Emigration Documents, Archive, Database, Yii Framework.

Contents

List of Abbreviations	v
List of Figures	vi
1 Introduction	1
2 The structure of emigration documents in the Archive	3
2.1 The emigrants and their concepts	3
2.2 Emigration documents and their data	3
2.3 Reasons and benefits to store physical and digital documents	6
3 Database of the emigration documents	8
3.1 What the database comprises	8
3.1.1 Conceptual model of the database	8
3.1.2 Logical Model of the database	9
3.1.3 Physical Model of the database	11
3.2 Security issues about the data of the database	13
4 Developed Application	14
4.1 How SGPE helps in the upload of data and the ease of its search mechanism	14
4.2 Used technologies and tools to develop SGPE	15
4.2.1 The Yii Framework	15
4.3 The web-based SGPE information system	16
4.3.1 Adding a record	21
4.3.2 Viewing a record	22
4.3.3 Updating a record	23
4.3.4 Deleting a record	23
4.3.5 Report menu	24
4.3.6 Problems found	24
5 Recovery of the physical documents of the municipal Archive	27
5.1 Gathering data from Emigration Passport files	27
5.2 Data recovery from the spreadsheets to the database	28
6 Conclusion	31

Bibliography

List of Abbreviations

[<i>CSS</i>]	Cascading Style Sheets
[<i>DBA</i>]	Database Administrator
[<i>DDL</i>]	Data Definition Language
[<i>DTD</i>]	Document Type Definition
[<i>MVC</i>]	Model-View-Controller
[<i>SGPE</i>]	Sistema Gerenciador de Processos de Emigração
[<i>SQL</i>]	Structured Query Language
[<i>Yi</i>]	Yes it is

List of Figures

3.1	Logical model of the database	10
3.2	Physical model of the database	12
4.1	Screen for the register of the emigrants	18
4.2	Timeline of the main page of the SGPE	20
4.3	List of Emigrant’s documents	20
4.4	Generic menu	21
4.5	Form to create an emigrant document	21
4.6	Review of a registered emigrant document	22
4.7	Management options	22
4.8	Form to update an emigrant document	23
4.9	Pop-up of exclusion confirmation	23
4.10	Graphic showing the count of the records in the database	24
4.11	Stack percentage of records	24
4.12	Example of queries	24
4.13	Rendering the correct component (a) and rendering the incorrect component (b)	25
4.14	Correct component (a) and incorrect component (b)	25
5.1	Initial spreadsheet that stores data about emigration documents	27
5.2	Log file	29

Chapter 1

Introduction

This report aims to introduce and describe the web-based application developed to support municipal Archives handling a collection of emigration documents. This work was done to fulfill a partial requirement of PDIInf, the doctoral program at Department of Informatics in University of Minho - Portugal. The initial idea of develop an application with this purpose emerged from the thesis theme "Formal Description and Automatic Generation of Learning Spaces based on Ontologies" approaching as a case study, the automatic generation of Learning Spaces for Museums. To set up this case study, it is necessary to obtain the information of the records of emigrants in digital form.

The collection of emigration documents treated in this work is dated from the 1960s and 1970s, however, the application can be used for any time period. This is very important because the organization of the documents of migration is defined by the Portuguese government (Junta da Emigração); They change forms from time to time however the application must be able to deal with it.

This report aims at discussing why to develop an application for this purpose, as well as showing the difficulties found in handling aiding tools (like *Yii framework*¹) and the results obtained with the finalization and deployment of the system. Furthermore, this document presents all the process until achieve the objective (the final system). Below are listed the steps performed to reach the final goal.

1. The analysis of emigration records, the data they contain and how they were organized in physical documents;
2. The modeling of the data identified in the previous item and the creation of the database;
3. The research and the study of a framework with good documentation that meets the needs required by this work including the chosen programming language (PHP);
4. The development of the final application;
5. The recovery of the data from a spreadsheet that contains information collected before the development this system.

Many problems were encountered. Some of them were concerned with the recover of the data from the initial spreadsheet, but most of the troubles were related to extensions and known bugs of the Yii framework. These problems were circumvented and are reported in Chapter 4 and Chapter 5, respectively.

The report is organized as follows. Chapter 1 introduces the idea supporting there work on and the development phases. Chapter 2 presents how Emigration data is organized in their original support (the Archive physical documents). Chapter 3 discusses the design of the database, as well as its final schema. In Chapter 4 is presented the developed system with its features and is explained

¹Available at: <http://www.yiiframework.com>

how the application helps with the records to be stored. In this chapter also some screenshots of the application are shown. Chapter 5 demonstrates what was done to collect data from the emigration documents before this application is finished and how it was possible to recover the data from the spreadsheets to the database. Finally, in Chapter 6 is described what was made and future works.

Chapter 2

The structure of emigration documents in the Archive

To introduce the domain of this project, in this chapter it will be described the most important concepts that there are around the topic Emigration in what concerns the official documentation stored in Archives.

2.1 The emigrants and their concepts

The main concept for this work is the emigrant, who is a person who leaves his country to live in another. Most of them do it to improve their economic conditions, search for jobs, escape from an area of natural disaster, existence of wars, etc.

The concept of emigrant has undergone changes over the years. According to the Portuguese Decree-law 34 330¹ of December 27th, 1944, an emigrant is defined as the citizen who leaves the national territory to work in another country, women who will accompany or join the emigrant husband, relatives by consanguinity in any degree of the direct line or up to the third degree of cross-line of any emigrant wishing to accompany or to join him.

Years later, another Decree-law was established. The Portuguese Decree-law number 44 428² of June 29th, 1962, added a clause about the definition of emigrant: those who transfer their residence to foreign country benefiting from the quality of emigrant or equivalent.

2.2 Emigration documents and their data

From this moment, having the definition of emigrant, the application form to get official permission and obtain a passport for emigration can be detailed with all necessary data.

These documents of emigration, as already mentioned in Chapter 1, contain information such as general data of the documents, general data about the emigrant, data of the applicant, dispatch, attached documents, family members accompanying the emigrant, desired type of transportation, qualifications (literary and professional) and criminal records, family members in charge of the emigrant who remain in the country, previous travel abroad, details of the person calling the emigrant, employment contract, details about women and minors employed, details of the married men who leave the family in the country of origin, aid in the destination country, among other information. All this data, included in the emigrant documentation described, is needed to the emigrant obtain his passport.

Each item mentioned above will be described in Table 2.1 in order to clarify its meaning and the data contained:

¹Available at: <https://dre.pt/application/file/568641>

²Available at: <https://dre.pt/application/file/164552>

Table 2.1: *Explanation about the data of the emigration documents*

Explanation about the data of the emigration documents		
Item	Fields contained	Meaning
General data about the emigrants documents	Emigrants document number in City Council; Emigrants document Number in Board of Emigration; and Year.	Data that identifies the emigration's documents.
General data about the emigrant	Name; Nationality; Date of Birth; Filiation; Civil Status; Name of Spouse; Number of Identity; and Residence.	Data that identifies the emigrant, their parents, the spouse and address (residence).
Data of the applicant	Name; Residence (place and town (parish)); and, Country and location.	Data that identifies the applicant and location that he intends to emigrate.
Dispatching³	Date that the emigration document was sent; Number of the craft; Date that was granted; Number of the license of emigration; Passport number; Ship, class and company; and Boarding date.	Data that identifies the dispatch of the documents of the emigrant.
Attached documents	Several documents such as: medical certificate, birth certificate, marriage certificate, work contract, etc.	Documents provided to get a visa. There is a list of documents to be marked as checked when the applicant deliver them.
Family members accompanying the emigrant	Name; Date of Birth; Age; kinship with the emigrant; and literary qualifications.	Data that identifies the members of family that accompanying the emigrant.
Desired type of transportation	Desired type of transportation; Approximate date of shipment; If the pass on the company is paid; Port of Boarding; and Port of landing.	Data that identifies the desired type of transportation by the emigrant. These data are grouped as "Boarding" in the emigration documents.
Qualifications (literary and professional) and criminal records	Current occupation; Literary qualifications; Entity where worked; If the applicant has already been judged by the courts; If the applicant or his relatives have any case pending.	Data that identifies the literary and professional qualifications of the applicant, as well as data about the emigrant's criminal records.
to be continued		

³Portuguese: expediente

Explanation about the data of the emigration documents (continued)		
Item	Fields contained	Meaning
Family members in charge of the emigrant who remain in the country	Name; Age; kinship with the emigrant; and Residence of each person of the family.	Data that identifies the family members in charge of the emigrant who remain in the country.
Previous travel abroad	Whether or not it is the first time that the applicant is traveling; Date on which returned; Passport number and date; Entity that issued the passport; Whether or not the applicant came repatriated; Reasons for repatriation; and if the applicant has already submitted a request to emigrate.	Data that identifies if is the first time that the applicant is traveling to another country.
Details of the person calling the emigrant	Name; Residence; kinship of the emigrant with the caller; How long is resident in the country that the emigrant goes; and Data of the passport that the caller did leave the country.	Data that identifies the caller who is calling the emigrant to the destination country.
Employment contract	Name and Residence of the contractor; If the applicant knows the contractor and since when; If the applicant does not know the contractor, how it obtained the contract; Name, residence, occupation and filiation of the intermediary; Passport that the intermediary did leave the country; kinship of the applicant with the intermediary and how much was paid or has to pay for the employment contract.	Data that identifies the contractor and the intermediary, if exists.
Details about women and minors employed	Whether or not already worked to the contractor or to the family of him/her; Duration of the work; Occupation.	Data that identifies women and minor employed.
Details of the married men who leave the family in the country of origin	Knowledge of woman about the target location of the husband; and if the wife considers that her maintenance is assured by the husband in the country that she stays.	Data about the married man who leave the family in the country of the origin. This data (statement) should be provided by the woman of the married man.
to be continued		

Explanation about the data of the emigration documents (continued)		
Item	Fields contained	Meaning
Aid in the destination country	Whether or not the applicant has family in the destination country; Name of the people who will give the aids; Degree of kinship; Residence; Occupation; How long he/she lives in that country; What are the possibilities to provide assistance; If the person who will give assistance has already traveled to Portugal. If yes, when and the length of stay; If has maintained correspondence with these relatives; If the relatives know the applicant's claim and vowed to assist it; and how it will help the applicant in the country of destination.	Data about the people who will aid the emigrant in the destination country.

2.3 Reasons and benefits to store physical and digital documents

There are several reasons to store and maintain historical documents in Archives. According to the book (1), "in offices, private and public organizations, for-profit and not-for-profit, people create and destroy records, file and misfile them, and store in boxes, desks, and cabinets. Eventually, the records accumulate and are a nuisance as they become more and more difficult to use and store". The referred book describes how to store, organize and keep documents physically. However the given problem for the nuisance and difficulty of use and storage of these records is the smallest of the problems, since there are other problems like natural disasters such as floods, fires, etc. Furthermore, there are issues of theft and loss of documents. These problems are circumvented with the creation of a computer database, i.e., the storage of the documents in digital format. This is precisely the aim of the application described in this report.

Storing these records in digital format helps people to find information about their family faster than searching in the physical documents, for example. In addition, once transcribed a document, it will no longer be necessary to reinterpret the original document in future researches (except in cases where the end-user wants to see the original document), since the documents were filled out by hand or typewriter.

In this report, the term "end-user" can be understood as professors, researchers, students, authors, genealogists or any type of people that wants to study these documents. This information has benefits for both the Archive, and the Community. Some benefits reported in (1) are summarized below, organized into two parts: benefits to the Archive (organization) (see Table 2.2) and to the Community (see Table 2.3).

Table 2.2: *Benefits to the Archive*

Benefits to the Archive	
Benefit	Description
Daily Operations	Daily operations require the use of the records with little circulation. Easy access to the information of the documents.
Social responsibility	An archive preserves historical information about the organization by documenting its past. This documentation is invaluable to society and is informative and educational to a variety of potential archives' users.
Security	Adequate security protects records from theft by authorized and unauthorized users from within and outside the organization. Security is an issue for all archives, whether or not their records have any monetary value.
Preservation	The provision of adequate facilities for the protection, care, and maintenance of archives.

Table 2.3: *Benefits to the Community*

Benefits to the Community	
Benefit	Description
Facility on access	The search for documents and information is faster than in the original document
Facility in the execution of queries	Easy way to run queries about the emigration records without needing to handle paper documents
Readability	The legibility of the documents is important to the end-user, i.e., once transcribed a document, will no longer be necessary to try to understand it in future research

As described in tables 2.2 and 2.3, there are some benefits to the Archive and to the Community when these documents are stored in an organized manner. To achieve the benefits mentioned, the computerization is an important factor.

With this factor in mind, it was decided to create a database to store the information of the emigrants documents. Thus, with the data organized in a database, the queries become easier, especially with the help of the application described in this report. Chapter 3 explains how the database was constructed and the information that it stores based on the study of this chapter.

Chapter 3

Database of the emigration documents

In this chapter is described the relational database (how it is organized: conceptual, logical and physical model), and how the security of the data of the database is made.

3.1 What the database comprises

3.1.1 Conceptual model of the database

The data described in Section 2.2 was modified and rearranged (grouped or separated) and is described below in a conceptual model:

- Data about the emigrants documents:
 - Registration number at City Hall, registration number at the Board of Emigration and year;
 - Age and civil status of the emigrant at the document;
 - Destination (country and location);
 - Dispatching (date, craft and passport);
 - Boarding data (type of the transport, company, date, location, designation of the transport, whether or not the pass is paid, landing port, and with who the applicant travels);
 - Qualifications (literary qualifications, occupation and workplace);
 - Economic conditions (remuneration, number of days that work, reason for the emigration and expenses of travel);
 - Criminal record (whether or not is judged and if the applicant or his relatives have any case pending);
 - Aid in the destination country (name of the people who will give the aids, kinship, residence, occupation and aid provided);
 - Kinship of the emigrant with the caller;
 - Employment contract (how the emigrant get the employment contract, occupation and salary);
 - Women and minors employed (whether or not already worked to the contractor or to the family of him/her, duration of the work and occupation);
 - Declaration of the woman (knowledge of woman about the target location of the husband and if the wife considers that her maintenance is assured by the husband in the country that she stays).
- Residence (Location) of the emigrant:
 - County;

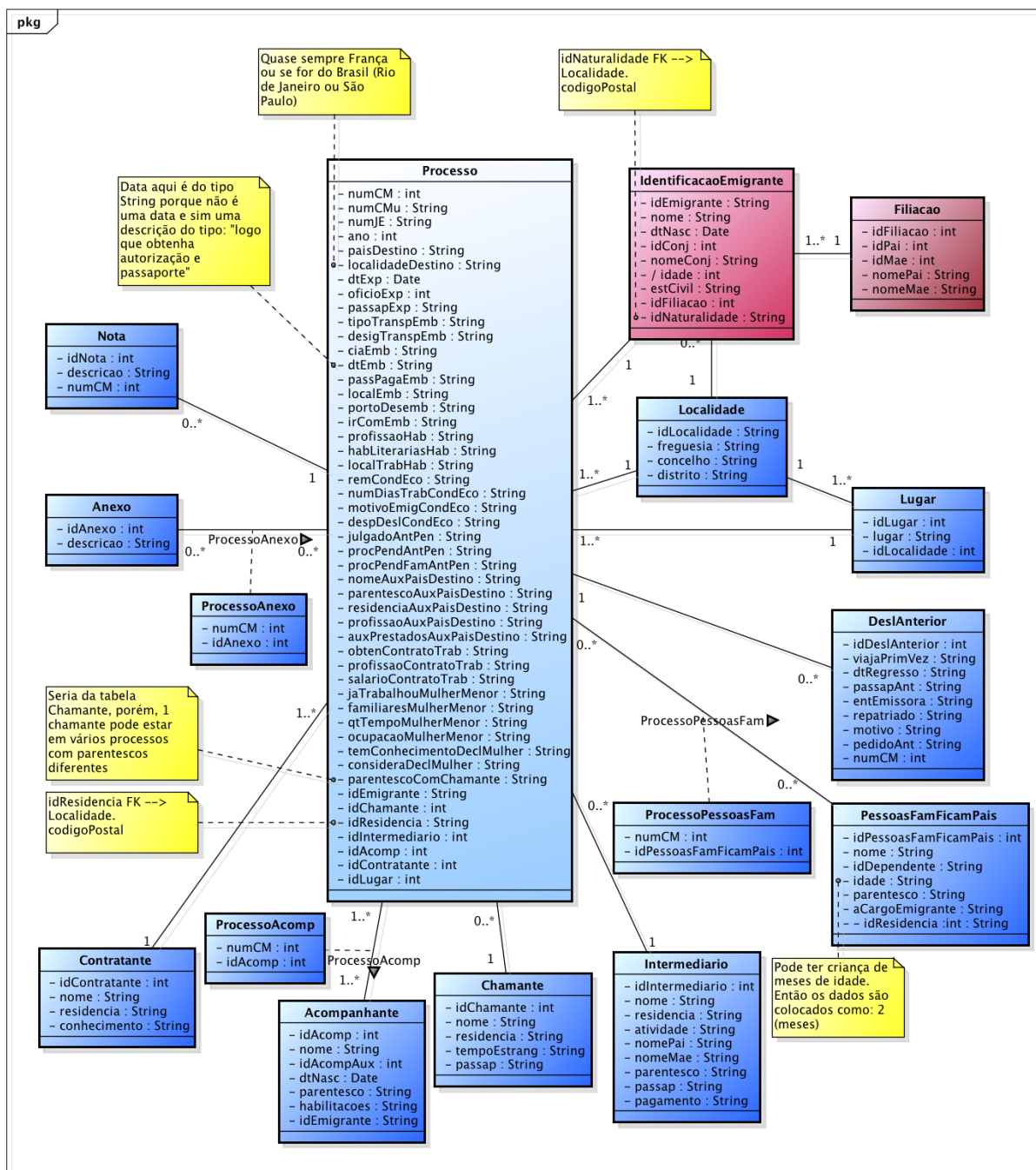
- District;
 - Parish; and
 - Street.
- Data about the emigrant: emigrant's identification, name, place and date of birth, filiation and spouse;
 - Filiation of the emigrant: name of the parents of the emigrant;
 - Data about the caller: name, residence, time that is in the abroad and passport;
 - Data about the contractor: name, residence and if the caller knows the emigrant;
 - Data about the intermediary: name, residence, occupation, parents' name, kinship with the emigrant, passport and how much was paid to the intermediary;
 - Description of the attached documents;
 - Data about the accompanying: name, age, date of birth, kinship with the emigrant and qualifications;
 - Data about the persons (family) who stays in the country: name, age of the dependent, kinship with the emigrant, if the person is in charge of the emigrant and residence;
 - Description of other information (notes) about the emigration documents; and
 - Previous travel abroad: whether or not it is the first time that the applicant is traveling, date on which returned, passport number and date, and entity that issued the passport.

The database was built taking into account the data listed above (conceptual model); the next subsections 3.1.2 and 3.1.3 describe the technical part of the database.

3.1.2 Logical Model of the database

The logical model shows the relations between the database tables as well as primary and foreign keys, i.e., the technical part of the database, but not yet the physical model, where is defined how the database was built.

Figure 3.1 illustrates the logical model with the primary and foreign keys, relations between the tables, and the cardinality of these relations.



powered by Astah

Figure 3.1: Logical model of the database

As already mentioned, this step contains the primary and foreign keys of the database tables. Each class in Figure 3.1 have a primary key and some classes have foreign keys assigned to it. In Table 3.1 is described each primary key and references of the foreign keys. To deal with relations with cardinality 0..* to 0..* (many to many) were created association classes that specify the common data between tables in the relation. The tables that hold these characteristics are *ProcessoAcomp*, *ProcessoAnexo* and *ProcessoPessoasFam*.

Table 3.1: *Primary and Foreign Keys of the Database*

Primary and Foreign Keys of the Database		
Table	Primary Key	Foreign Key
Acompanhante	idAcomp	
Anexo	idAnexo	
Chamante	idChamante	
Contratante	idContratante	
DeslAnterior	idDeslAnterior	numCM references Processo
Filiacao	idFiliacao	
IdentificacaoEmigrante	idEmigrante	idFiliacao references Filiacao; idNaturalidade references Localidade
Intermediario	idIntermediario	
Localidade	idLocalidade	
Lugar	idLugar	idLocalidade references Localidade
Nota	idNota	numCM references Processo
PessoasFamFicamPais	idPessoasFamFicamPais	idResidencia references Localidade
Processo	numCM	idEmigrante references IdentificacaoEmigrante; idChamante references Chamante; idResidencia references Localidade; idLugar references Lugar; idIntermediario references Intermediario; idContratante references Contratante
ProcessoAcomp	idProcessoAcomp	numCM references Processo; idAcomp references Acompanhante
ProcessoAnexo	idProcessoAnexo	numCM references Processo; idAnexo references Anexo
ProcessoPessoasFam	idProcessoPessoasFam	numCM references Processo; idPessoasFamFicamPais references PessoasFamFicamPais

After describing the logical model of the database its physical model will be discusses in the next subsection 3.1.3.

3.1.3 Physical Model of the database

The physical model describes in detail the implementation of the final database. Taking into account that model, it is possible to execute the creation of the database and their tables (with the size and the type of the fields) in a computer (server) with the SQL (Structured Query Language) statements. The database chosen for this project was the Mysql. It is important to mention in this point, because the physical model depends of the database chosen. Figure 3.2 shows the physical model of the database with all the characteristics mentioned above.

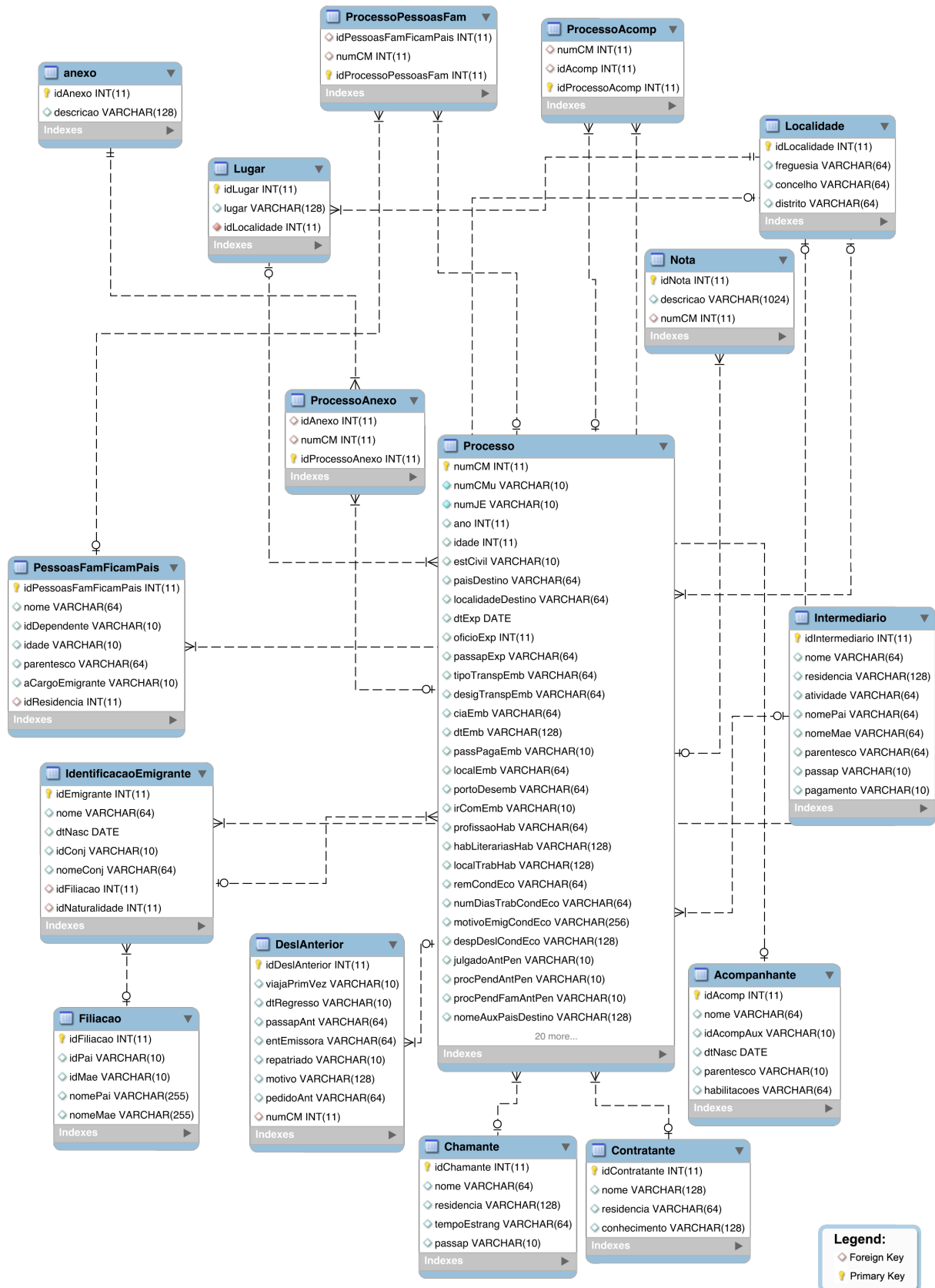


Figure 3.2: Physical model of the database

As said above, the diagram shown in Figure 3.2 can be translated into DDL (Data Definition Language) statements of SQL to create the schema. The listing below is an example of a DDL statement, in this case to create the table `anexo`:

```
1 CREATE TABLE 'anexo' (  
2   'idAnexo' int(11) NOT NULL AUTO_INCREMENT,  
3   'descricao' varchar(128) DEFAULT NULL,  
4   PRIMARY KEY ('idAnexo')  
5 );
```

The physical construction of the database will be completed after the execution of with DDL program written according to the description of all the tables in Figure 3.2. With this done, the next step is to populate the database and ensure that the data is secure. Section 3.2 will describe how security of the data was dealt in the work here reported.

3.2 Security issues about the data of the database

Given the importance of the Archival documents handled in this work, the topic of data security is very relevant. So it will be discussed in this section.

Any database with important data should have the security of the data made in different ways. Many situations can occur (as mentioned in the chapter 2), like natural disaster, fire, floods, thefts, data corruption by power outage, etc. To ensure that the data is secure if of those happens, some precautions must be taken by the database manager (DBA). This precautions should be done independent of the situation (2).

For this case, the server that contain the database should be protected from invaders, physical damage and inappropriate use of the software that controls (insert, read, update and delete) the records of the database.

The security measures for this database shall take into account all the situations mentioned above. To ensure this, besides the replication of the data in different geographical locations, also is guaranteed the access to the database via software (see the chapter 4) with users control. The backup of the database is made daily.

Chapter 4

Developed Application

In this chapter is presented the final application named SGPE¹ (Sistema Gerenciador de Processos de Emigração) that controls the database described in Chapter 3. It is also shown in this chapter: how the application helps in the management of the data recovered from the emigrant's documents; the easy of searching documents of emigrants in the database; the technologies and tools that were used to develop the application; and the features that the application offers.

SGPE is a web-based application with the goal of store and manage emigrant's documents to later use in the creation of virtual learning spaces, as mentioned in the chapter 1.

The application was developed using tools like the Yii framework, PHP programming language and Javascript (jQuery), as will be described in the section 4.2. The database chosen was Mysql as mentioned in the subsection 3.1.3.

4.1 How SGPE helps in the upload of data and the ease of its search mechanism

SGPE application makes easier some daily operations of the Archives. The operation concerned with this project is the upload of data from the physical documents to a computer database. In the section 2.3, where are described the reasons and benefits outcoming for the operation were discussed.

Some specific benefits that SGPE provides to the users are shown in Table 4.1. This table summarizes the advantages of having a database managed by a web-based application (3).

Table 4.1: *Key Benefits of a web-based application to manage the data of the database (Adapted from (3))*

Key Benefits of a web-based application to manage the data of the database	
Benefit	Description
Power & Flexibility	Sorts and accesses to the web-based data in the ways that SGPE provide using powerful search abilities.
Data Access Control	Provides different views to the end-users. Controls who can edit, add or delete data based on internal rules.
Media independent access of data	Provides many types of media (text, graphics, audio, video, etc.), whats helps the users to use, with more efficiency, the application and the database.
Data update	Provides management of the data from many locations.
Friendly interface	The interface of SGPE offers better ways to search the documents.

Table 4.1 shows the benefits of having a web-based application to manage the database. Maybe the most important feature offered by SGPE is its search mechanism and analyze data to find more

¹English: Management System of the Emigration Documents

efficiently than in the physical documents (older documents), mainly in the documents that were written by hand or typewriter (many times illegible). Next sections present the technologies and tools used to develop the application as well as the features and the search methods that were implemented.

4.2 Used technologies and tools to develop SGPE

The start of the development brought questions about the programming language to choose and the framework to adopt in order to help in the writing of CRUD² (Create, Read/Retrieve, Update and Delete). To take these decisions it was needed to study alternative technologies and tools.

The chosen programming language was PHP. On one hand because of the author's experience, and on the other hand, because PHP offers all the necessary features to develop this kind of application. After selecting the programming language, a framework that works with that language was chosen. The Yii framework, as already referred in Chapter 1, has all the necessary requirements to aid in the development of the application.

4.2.1 The Yii Framework

Yii is a high-performance PHP framework best for developing Web 2.0 applications. The framework comes with rich features like MVC (Model-View-Controller), DAO (Data Access Object) / ActiveRecord, I18N/L10N, among others, what can reduce significantly the development time.

The MVC³ is an architecture to organize the code for a rich client system that splits the total code into the Model (that manages the behavior and data of the application domain); the View (that manages the display of information); and the Controller (that interprets the inputs (mouse, keyboard, etc.) from the user, informing the model and/or view to change as appropriate) (4).

Other important features are DAO⁴ and ActiveRecord, which help with the persistence of the objects, simplifying the persistence mechanism of the application.

Another interesting feature that comes with the Yii is the so called "i18n and l10n"⁵ that means internationalization and localization, respectively. This feature helps with the adaptation of an application content to meet the language, cultural and other requirements of a specific target market (5). The "l10n" covers:

- Numeric, data and time formats;
- Use of currency;
- Keyboard usage;
- Collation and sorting;
- Symbols, icons and colors;
- Text and graphics containing references to objects actions or ideas which, in a given culture, may be subject to misinterpretation or viewed as insensitive;
- Varying legal requirements; and
- many more things.

And the "i18n" includes:

- Designing and developing in a way that removes barriers to localization or international deployment. This includes such things as enabling the use of Unicode, or ensuring the proper handling of legacy character encodings where appropriate, taking care over the concatenation of strings, avoiding dependance in code of user-interface string values, etc;

²To learn more visit: <http://searchdatamanagement.techtargget.com/definition/CRUD-cycle>

³To learn more visit: <http://msdn.microsoft.com/en-us/library/ff649643.aspx>

⁴To learn more visit: <http://tutorials.jenkov.com/java-persistence/dao-design-pattern.html>

⁵To learn more visit: <http://www.w3.org/International/questions/qa-i18n>

- Providing support for features that may not be used until localization occurs. For example, adding markup in your DTD (Document Type Definition) to support bidirectional text, or for identifying language. Or adding to CSS (Cascading Style Sheets) support for vertical text or other non-Latin typographic features;
- Enabling code to support local, regional, language, or culturally related preferences. Typically this involves incorporating predefined localization data and features derived from existing libraries or user preferences. Examples include date and time formats, local calendars, number formats and numeral systems, sorting and presentation of lists, handling of personal names and forms of address, etc;
- Separating localizable elements from source code or content, such that localized alternatives can be loaded or selected based on the user's international preferences as needed.

To create an application with the Yii Framework is very easy, the developer just needs to follow the simple steps below⁶:

1. Create the database;
2. Let the Yii generate the base PHP code;
3. Customize the code to fit the needs of the application.

4.3 The web-based SGPE information system

Now that Yii framework was introduced, in this section the details of SGPE information system will be given.

As shown previously, to create an application with Yii framework it is necessary to follow some steps. For the SGPE was not different. The first step is to create the database, which was already presented in Chapter 3.

The second step is to generate the basilar PHP code with the aid of the Yii framework. The framework has a module called Gii that provides code generation capabilities. The purpose of Gii is to create the fundamental model, view, and controller files required by the web-based application (6).

In the Gii module two options were executed: the generation of models; and the generation of CRUD operations. For the generation of models, the only thing that is needed is to enter the table names of the database and the result are PHP files with the name of the tables. Each PHP file created has a PHP class associated with the same name. The models in Yii store data and define the business rules for that data. To handle these features, each class has some methods, the most important are called `rules()` and `relations()`.

`rules()` method returns an array of rules by which the model data must abide. This method represents a key benefit of using a framework: built-in data validation. `relations()` method is used to indicate one model's relationship to other models, i.e., the relations between models with the foreign keys (6).

For instance, a model will receive the name `identificacaoEmigrante` if it represents the table `identificacaoEmigrante` of the database. From this model, Gii tool will create a file named `identificacaoEmigrante.php` that will contain the methods mentioned previously, as can be seen below.

⁶To learn more read the Larry's Ullman book entitled "The Yii Book: Developing Web Applications Using the Yii PHP Framework" available at yii.larryullman.com

rules():

```

1 public function rules()
2 {
3     // NOTE: you should only define rules for those attributes that
4     // will receive user inputs.
5     return array(
6         array('idEmigrante, nome', 'required', 'message'=>'O campo <b>{attribute
7             }</b> não pode ser vazio.'),
8         array('idEmigrante, idFiliacao, idNaturalidade', 'numerical', 'integerOnly
9             '=>true),
10        array('nome, nomeConj', 'length', 'max'=>64),
11        array('idConj', 'length', 'max'=>10),
12        array('dtNasc', 'safe'),
13        // The following rule is used by search().
14        // @todo Please remove those attributes that should not be searched.
15        array('idEmigrante, nome, dtNasc, idConj, nomeConj, idFiliacao,
16            idNaturalidade', 'safe', 'on'=>'search'),
17    );
18 }

```

rules() method presented above has the array to be returned with the attributes of the `identificacaoEmigrante` table with the rules for each attribute. For `idEmigrante` and `nome` attributes exists a rule that identifies these attributes as **required** by the model, i.e., in the final application, these attributes are mandatory to fill. If these attributes were not filled, the message: `O campo <idEmigrante OR nome> não pode ser vazio` will be shown. The other rules in this function are similar and their objective is to set the type and the maximum length (max) of all the table fields.

relations():

```

1 public function relations()
2 {
3     // NOTE: you may need to adjust the relation name and the related
4     // class name for the relations automatically generated below.
5     return array(
6         'idFiliacao' => array(self::BELONGS_TO, 'Filiacao', 'idFiliacao'),
7         'idNaturalidade' => array(self::BELONGS_TO, 'Localidade', 'idNaturalidade'
8             ),
9         'processos' => array(self::HAS_MANY, 'Processo', 'idEmigrante'),
10    );
11 }

```

relations() method presented above has the array to be returned with the attributes of the `identificacaoEmigrante` table with the relations between the `identificacaoEmigrante` model and other models, in this case, the `Filiacao` (`idFiliacao`), `Localidade` (`idNaturalidade`) and `Processo` (`processos`) models. Each line inside "return array" (line 5) describes the kind of the relation. Lines 6 and 7 have a kind of relation named `BELONGS_TO` that means that attribute "`idFiliacao`" and "`idNaturalidade`" belongs to the model `Filiacao` and `Localidade`, respectively. Line 8 has another kind of relationship. The relation established is `HAS_MANY` that represents the cardinality 1 to *, i.e., one emigrant (`identificacaoEmigrante`) can have many documents, but one document just refers one emigrant.

In addition to creating the model, the views and controllers also are created in the second step. The views are PHP files containing the code to display the forms and graphical components to the user, i.e., the code responsible for the user interaction with the application. In a web-based application, like SGPE, the views are a combination of HTML and PHP code that create the screen that the user will see in the browser (6).

A view of SGPE can be seen in Figure 4.1. This is a form for `identificacaoEmigrante` model. Note the red stars to advise the user that the field is mandatory (required), as seen previously in the rules of the model.

The screenshot shows a web form titled "Identificação do Emigrante". It contains the following elements from top to bottom:

- A text input field for "Identificação do Emigrante".
- A text input field for "Nome".
- A date selection dropdown for "Data de Nascimento" with the placeholder text "Selecione uma data".
- A text input field for "Idade do Cônjuge".
- A text input field for "Nome do Cônjuge".
- A dropdown menu for "Filiação" with the placeholder text "Selecione a Filiação", accompanied by a blue button labeled "Adicionar nova Filiação".
- A dropdown menu for "Naturalidade" with the placeholder text "Selecione a Naturalidade", accompanied by a blue button labeled "Adicionar nova Naturalidade".
- A black button labeled "Salvar Emigrante" at the bottom.

Figure 4.1: Screen for the register of the emigrants

As mentioned before, there are also the controllers, that are intermediary agents that handle user and other actions (6).

One controller also is a PHP file, like the models and views. The controller has some methods that can be understood as actions. These actions can be, for example, create, update, delete and show a register.

The corresponding method names always start with the word "action" as can be seen below (6):

- `actionCreate()`: for creating new model records;
- `actionIndex()`: for listing every model record;
- `actionView()`: for listing a single model record;
- `actionUpdate()`: for updating a single model record;
- `actionDelete()`: for deleting a single model record; and
- `actionAdmin()`: for showing every model record in a format designed for administrators.

To better understand the controller part of the MVC, an example of a controller (`identificacaoEmigranteController.php`) will be detailed showing the actions performed by the application. For the sake of space and simplicity, only `actionCreate()` will be shown.

actionCreate():

```

1   public function actionCreateEmigranteProcesso() {
2       $model = new identificacaoEmigrante;
3
4       // Uncomment the following line if AJAX validation is needed
5       $this->performAjaxValidation($model);
6
7       if((isset($_POST['idEmigrante']) && isset($_POST['nome'])))
8       {
9           $model->idEmigrante = $_POST['idEmigrante'];
10          $model->nome = $_POST['nome'];
11          $model->dtNasc = $_POST['dtNasc'];
12          $model->idConj = $_POST['idConj'];
13          $model->nomeConj = $_POST['nomeConj'];
14          $model->idFiliacao = $_POST['idCarregaFiliacaoEmigrante'];
15          $model->idNaturalidade = $_POST['idCarregaLocalidadeEmigrante'];
16
17          if($model->save()) {
18              $id = $model->getPrimaryKey();
19              $_SESSION['codEmigranteAjax'] = $id;
20          }
21      }
22
23      if (!isset($_POST['inserindoViaAjax'])) {
24          $this->render('createEmigranteProcesso', array(
25              'model'=>$model,
26          ));
27      }
28  }

```

As can be seen in the code above, first a new model is instantiated (in this case, the `identificacaoEmigrante` model). This is necessary to access the fields to create a new emigrant in the database (model). After that, if the `$_POST` values are not empty, each field of the model receives this value, that is the value of the components (html forms) and at last, the model is saved (line 14). Note the word "action" at the beginning of the method name.

After creating the fundamental models, views and controllers, in the third step, it is necessary to customize the application with the particular needs.

As can be seen in the figure 4.1, the components (buttons, textfields, dropdown menus, etc.) are not the built-in components of Yii. Yii framework has several extensions for various purposes; one to customize the components of the views of the application was used. The used extension is `Yii booster`⁷. This extension uses components similar to the well known framework `bootstrap`⁸.

Below are shown some screenshots of the SGPE application. The screenshots are related to the main flow of the application that is the creation, update, visualization and exclusion of the emigrant's documents.

When the user opens the SGPE, the first page that he accesses is a tutorial of "How to use the Management System of the Emigration Documents" with a timeline that shows the flow of the system, describing the functionality (creation, update and view) of all models of the application. These models are "identificacaoEmigrante", "Notas", "Filiacao", "Acompanhante", "Anexo", "Chamante", "Contratante", "Intermediario", "Localidade", "Lugar", "PessoasFamFicamPais" and "DeslAnteriores".

Figure 4.2 illustrates the main page. The navigation bar on the top of the page is the same to all views in the application. This navigation bar is a menu that gives access to the models "Processo", "identificacaoEmigrante" and "Notas" only, because the rest of the models are accessible from "Processo" model. There is still an item called "Report" that will be presented later.

⁷To learn more visit: <http://yii booster.cleverttech.biz/>

⁸To learn more visit: <http://getbootstrap.com/>



Figure 4.2: Timeline of the main page of the SGPE

From the main page, the user can access the views (pages) for creation, update, visualization or exclusion of any item of the navigation bar or yet, at the timeline, access any action except the creation of the models that are not in the navigation bar (Processo, Emigrante and Notas).

Following the timeline, the first thing to do is to access the documents of the emigrants ("Processo") page to create an emigrant record. Accessing "Processo" page, the first thing that appears is a list of emigrant's documents (see Figure 4.3) that shows a summary of the documents (10 per page).

Mostrando 1-10 de 233 resultados.

Código do Processo	Registro na Câmara Municipal	Registro na Junta da Emigração	Ano	País Destino	Localidade Destino
1	1/63	119/62	1962	E.U. Brasil	Rio de Janeiro
2	2/63	18864/62	1962	França	
3	3/63	161/63	1962	França	
4	4/63	446/63	1962	França	
5	5/63	766/63	1962	França	
6	6/63	22694/62	1962	França	
7	7/63	22136/62	1962	França	
8	8/63	18866/62	1962	França	
9	9/63	447/63	1963	França	
10	10/63	613/63	1962	França	

Navigation: << < 1 2 3 4 5 6 7 8 9 10 > >>

Figure 4.3: List of Emigrant's documents

There is also a generic menu, as illustrated in Figure 4.4, that shows the operation available to handle the emigrant's documents. However, this lateral menu is similar for all other models, always with the same options adapted to the model involved.

Depending on the page that the user is, the generic menu shows the appropriated options; for example, if the user is on the page to create an emigrant's documents, the options are 'list' or 'manage' documents, but if he is on the page to view an emigrant's document, the list of options are 'list', 'add', 'update', 'delete' the selected record. As the "Processo" model is the main of SGPE, there are two exclusive navigation items to facilitate the view of a record on the generic menu that

corresponds to the previous and the next document, respectively. Figure 4.4 shows these options. The next subsections will present the facilities available to the users: add, update, view or delete a record.



Figure 4.4: *Generic menu*

4.3.1 Adding a record

Figure 4.5 shows part of the form to create an emigrant's document, for the sake of space. To know all the data fields, see Chapter 2 in Section 2.2.

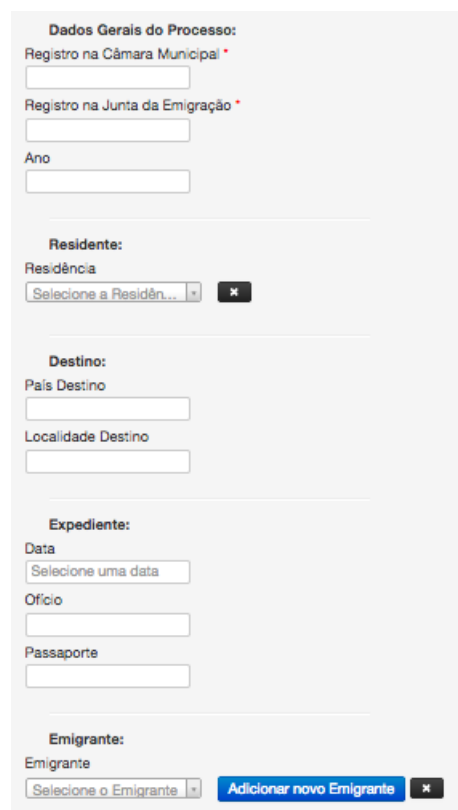
A web form titled "Dados Gerais do Processo:" (General Data of the Process). It is divided into several sections: "Registro na Câmara Municipal" with a text input field; "Registro na Junta da Emigração" with a text input field; "Ano" with a text input field; "Residente:" section with "Residência" and a dropdown menu labeled "Selecione a Residên..."; "Destino:" section with "País Destino" and "Localidade Destino" text input fields; "Expediente:" section with "Data" (dropdown "Selecione uma data"), "Ofício" (text input), and "Passaporte" (text input); and "Emigrante:" section with "Emigrante" (dropdown "Selecione o Emigrante") and a blue button "Adicionar novo Emigrante".

Figure 4.5: *Form to create an emigrant document*

Note in Figure 4.5 that field "Emigrante" has a button to create a new emigrant (the button "Adicionar novo Emigrante") associating it to the emigrant's record that is being registered, in case

he does not exist yet. The action of the button "Adicionar novo Emigrante" lets the user to create a new emigrant without the need to leave the current page. This facility is available to all the other models that have a relation (foreign key) with "Processo" model.

4.3.2 Viewing a record

There are two ways to view the data recorded in the table. One is by completing the registration of the emigrant's document: a view with the recently included data is shown, like a summary, for the user to analyze and then take the next action. Figure 4.6 shows a part of this summary.

Código do Processo	1
Registro na Câmara Municipal	1/63
Registro na Junta da Emigração	119/62
Ano	1962
Idade	24
Estado Civil	casada
Pais Destino	E.U. Brasil
Localidade Destino	Rio de Janeiro
Data	1963-01-02
Ofício	5
Passaporte	1899/63
Tipo de Transporte	navio
Designação de Transporte	Corrientes
Companhia	Argentina Dederó
Data	logo que obtenha autorização e passaporte
Passagem Paga?	sim
Local	Lisboa
Porto Desembarque	Rio de Janeiro

Figure 4.6: Review of a registered emigrant document

The second way to view the recorded data is selecting the wanted record. To do this it is necessary to select the *manage* option (gerenciar) on the generic menu (see Figure 4.4). Selecting the option, the following page appears (see Figure 4.7).































Mostrando 1-10 de 233 resultados.						
Código do Processo	Registro na Câmara Municipal	Registro na Junta da Emigração	Ano	Idade	Estado Civil	Ação
1	1/63	119/62	1962	24	casada	  
2	2/63	18864/62	1962	38	casado	  
3	3/63	161/63	1962	33	casado	  
4	4/63	446/63	1962	34	casada	  
5	5/63	766/63	1962	37	casado	  
6	6/63	22694/62	1962	17	solteiro	  
7	7/63	22136/62	1962	38	casado	  
8	8/63	18866/62	1962	29	casado	  
9	9/63	447/63	1963	39	casada	  
10	10/63	613/63	1962	32	casada	  

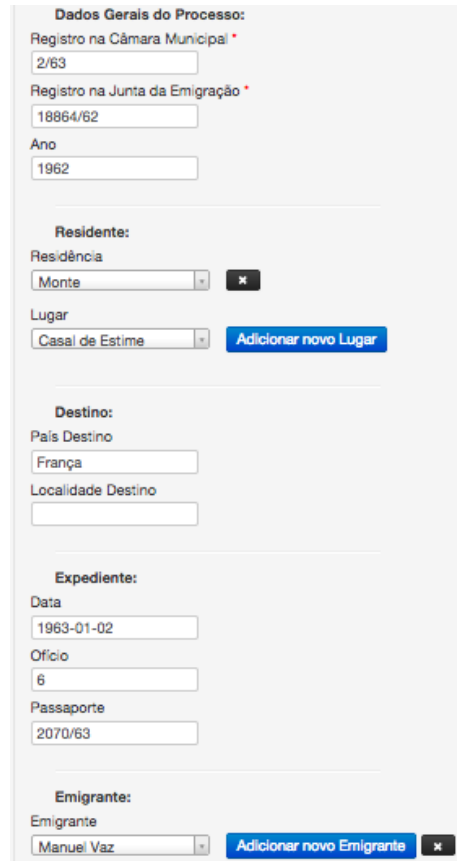
Figure 4.7: Management options

On this page there is a table with a list of documents and the last column ("Ação") contains three icons that correspond to the operations that user can do. The icon that triggers the action to

visualize a record is the "eye". When this icon is clicked, the page is presented as shown in Figure 4.6.

4.3.3 Updating a record

Another feature that SGPE has, is the option to update a record. To access this option, "pencil" icon (displayed in Figure 4.7) shall be selected. After that selection, a page like the one exhibited in Figure 4.8 appears with the data of the selected record to be updated.



The form is titled "Dados Gerais do Processo:" and contains several sections of input fields:

- Dados Gerais do Processo:**
 - Registro na Câmara Municipal: 2/63
 - Registro na Junta da Emigração: 18864/62
 - Ano: 1962
- Residente:**
 - Residência: Monte (dropdown menu)
 - Lugar: Casal de Estime (dropdown menu) with a button "Adicionar novo Lugar"
- Destino:**
 - Pais Destino: França
 - Localidade Destino: (empty text field)
- Expediente:**
 - Data: 1963-01-02
 - Ofício: 6
 - Passaporte: 2070/63
- Emigrante:**
 - Emigrante: Manuel Vaz (dropdown menu) with a button "Adicionar novo Emigrante"

Figure 4.8: Form to update an emigrant document

4.3.4 Deleting a record

To delete a record, there are also two ways to do it. The first mode is selecting the "trash" icon on the management page (see Figure 4.7), the second mode is to use the generic menu, and then chose the delete record option. Both the first and the second modes require confirmation before removing the selected record (see Figure 4.9).

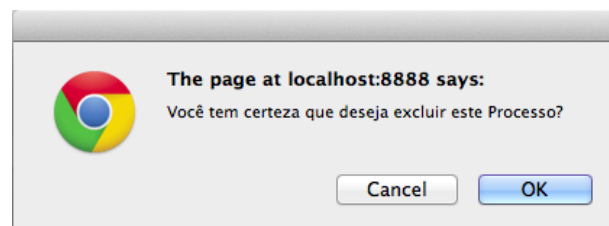


Figure 4.9: Pop-up of exclusion confirmation

4.3.5 Report menu

The last item in the navigation bar (on the top of SGPE main screen) gives access to reports and queries in the database. One feature that SGPE provides, is a summary of the number of the records in the database. Figure 4.10 shows the graphic with the number of records in each table of the database.

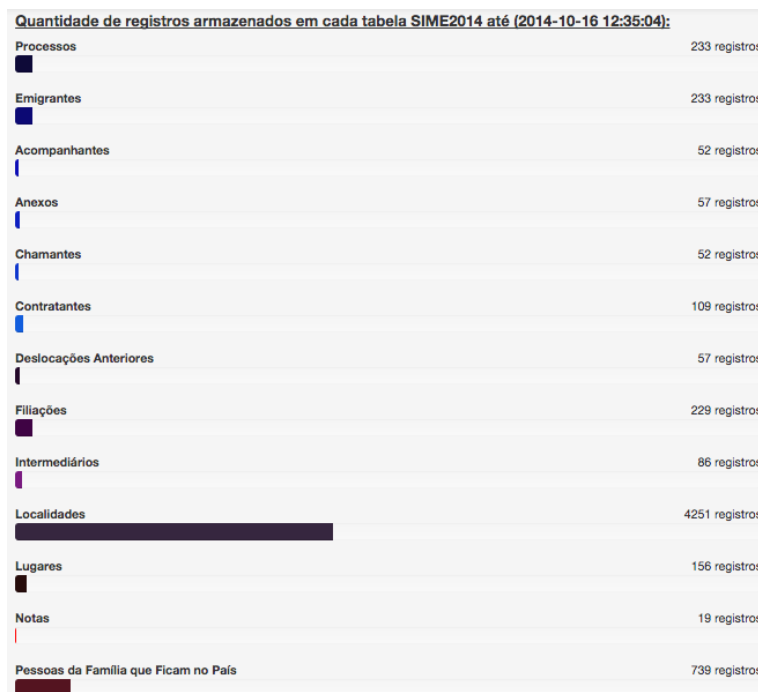


Figure 4.10: Graphic showing the count of the records in the database

The same page also exhibits another way (percentage's stack) to view the number of records in the database. It can be seen in Figure 4.11.



Figure 4.11: Stack percentage of records

Another feature that SGPE provides are queries to the database. Accessing the "Report" menu, it is possible to elaborate queries to retrieve information from the database. Figure 4.12 shows two kinds of queries that are possible to do. The first is a simple query to find records of emigrants between the years 1960 and 1962. The second query is to search for documents of the emigrants that leave your family in the country of the origin.

▼ Consultar Processos entre anos

até

► Consultar processos de emigrantes que deixam pessoas da família no país

Figure 4.12: Example of queries

4.3.6 Problems found

Yii framework helps to reduce the development time, however the use of such tools has some drawbacks. Mainly when these tools are opened for developers extend the framework. In the devel-

opment of SGPE, most of the troubles found were related to extensions namely concerned with the use of jQuery.

In the forums, many problems are reported with Yii framework and jQuery. As SGPE has large forms to be filled, a way to fill them easily is using ajax/jQuery, because the content of the pages (forms) can be rendered without the need to reload the entire page — this is a very important feature of SGPE.

In Chapter 1 was mentioned that Yii framework has some known bugs and problems with some extensions. In Section 4.3 also was mentioned one extension similar to the known Bootstrap framework used to work with customization of the components, the Yiibooster.

The Yiibooster extension working together with the jQuery/Ajax, brings some problems to the render of the components. Figure 4.13 shows in (a), the correct form of the presentation of the components and in (b), the incorrect form, without the render of the extension component.



Figure 4.13: Rendering the correct component (a) and rendering the incorrect component (b)

Looking to Figure 4.13 can not be seen much differences between the component rendered in (a) and that rendered in (b), but in Figure 4.14 can be seen that the component rendered in (a) has some features to help the user to enter with the correct information and in (b), this does not occur. The extensions, frameworks and applications should provide these mechanisms to actually aid the end-user or else, it would not be necessary to have an application.

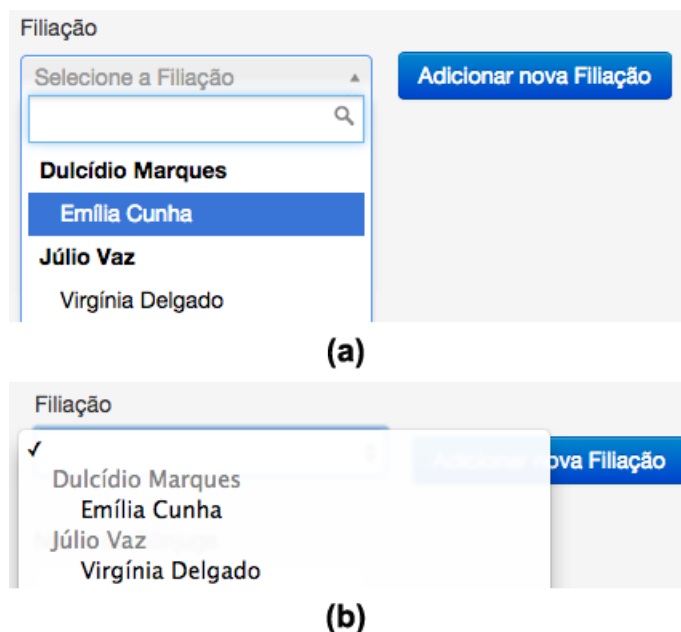


Figure 4.14: Correct component (a) and incorrect component (b)

The problem illustrated in Figures 4.13 and 4.14 happened because Yii framework was loading jQuery twice or more. This problem was circumvented not permitting Yii framework to load it more than once. In the listing below, there is a code snippet showing the resolution for this stated problem.

```
1 public function actionCarregaFiliacoes() {
2     $model = new filiacao;
3
4     Yii::app()->clientscript->scriptMap['jquery.js'] = false;
5     $this->renderPartial("carregaFiliacoes", array('model'=>$model), false,
6         true);
7     Yii::app()->clientscript->scriptMap['jquery.js'] = true;
8 }
```

The code above presents a function to:

1. disable jQuery:
line `Yii::app()->clientscript->scriptMap['jquery.js'] = false;`
2. render a view, not rendering the entire layout:
line `$this->renderPartial("carregaFiliacoes", array('model'=>$model), false, true);`. The two last parameters of `renderPartial` function serve to control whether or not the result is returned or displayed to end users. So, this could be used to produce some solutions on Ajax/jQuery requests;
3. enable jQuery:
line `Yii::app()->clientscript->scriptMap['jquery.js'] = true;`

Thus, disabling, rendering a view, and enabling the jQuery again, the components that are called by Ajax, appear correctly. The only thing to do is to create a separately php file with the widget to be rendered and in the Ajax/jQuery code, call in Ajax success function, the jQuery `load` method with the url to the php file of the widget to be rendered.

Chapter 5

Recovery of the physical documents of the municipal Archive

At a first stage (previous to SGPE development) the data contained in the emigration documents of the Municipal Archive of Fafe, was uploaded to spreadsheets. In this chapter we discuss how information was transferred from the spreadsheets to the final database used by SGPE application.

5.1 Gathering data from Emigration Passport files

In an initial stage of this digitalization project aimed at the recovery of emigration data contained in passport application forms, it was decided to create a book of spreadsheets to store in a structured way that data. Each record of the application form collection will be stored as a line; Columns identify each record field, as partially shown in the screenshot of Figure 5.1.

1	A	B	C	D	E	F	G	H	I	J	K	L
2	IdEmigrante	NumCM	NumJE	ANO	LUGAR	FREGUESIA	CONCELHO	PAÍS	LOCALIDADE	Data	Ofício	Passap
220	2821026	231/63	13196/63	1963	Lugar de Igreja	Monte	Fafe	França		1963.07.18	2885	15697/63
221	1767206	214/63	12452/63	1963	Lugar de Avejão	Arões S. Romão	Fafe	França		1963.07.09	2757	15641/63
222	704756	215/63	12385/63	1963		Monte	Fafe	França		1963.07.18	2762	14547/63
223	757690	217/63	10770/63	1963	Rua Albino Guimarães	Fafe	Fafe	França		1963.07.13	2802	15146/63
224	998467	216/63	12386/63	1963	Lugar de Queimadela	Fafe	Fafe	E. U. Brasil	S. Paulo	1963.07.10	2763	14519/63
225	2888060	243/63	13934/63	1963	Lugar do Barreiro	Serafão	Fafe	França		1963.07.26	2986	16397/63
226	760391	278/63	15875/63	1963	Lugar dos Pinheiros	Medelo	Fafe	França		1963.08.19	3165	18567/63
227	2889163	259/63	15386/63	1963	Lugar do Aldro	Vilacova	Fafe	França				19025/63
228	748687	288/63	13195/63	1963	Lugar da Poça	Golães	Fafe	França		1963.08.30	3240	19100/63

Figure 5.1: Initial spreadsheet that stores data about emigration documents

In Figure 5.1 is shown the Excel book for the aggregation, in a normalized format, of emigration documents' data. In the bottom line, it can be seen that there are multiple worksheets, each one grouping related data according to the information structure described in Section 2.2.

At this phase, 240 emigration documents were recovered and uploaded to the spreadsheet. This manual process of directly filling the spreadsheet cells with values read in the original source document is error-prone, and some cells were incorrectly filled. This happens because there is a clear lack of a program to deal with this kind of scenarios. There are some data values (in this domain) difficult to treat, because they are text fields (open fields), i.e., fields that can have any text, but there are also fields with values contained in enumeration sets, and those can be controlled forcing a selection in a restricted list of options.

It is important to notice that this error control is provided by an interface like the one implemented by SGPE, but it was not available when the spreadsheet was fulfilled directly.

5.2 Data recovery from the spreadsheets to the database

After the development of SGPE and the construction of its database, it was necessary to perform an important task to avoid losing all the data stored in the spreadsheet book during the first stage. To accomplish this task, two scripts were written, one in Java and other in Perl. A piece of each script is presented in the two listings below.

Java:

```

1  public String readXLS(int pNumSheet) {
2      //Initialize vars
3      list.removeAll(list);
4      cellCount = 0;
5      DecimalFormat df = new DecimalFormat("#"); //To format cell whit .## to # (
        Trim number)
6
7      try {
8          FileInputStream file = new FileInputStream(new File("SIME2014.xls"));
9
10         //Get the workbook instance for XLS file
11         HSSFWorkbook workbook = new HSSFWorkbook(file);
12
13         //Get first sheet from the workbook
14         sheet = workbook.getSheetAt(pNumSheet);
15
16         //Iterate through each rows from first sheet
17         Iterator<Row> rowIterator = sheet.iterator();
18         while(rowIterator.hasNext()) {
19             Row row = rowIterator.next();
20             //For each row, iterate through each columns
21             Iterator<Cell> cellIterator = row.cellIterator();
22             while(cellIterator.hasNext()) {
23
24                 Cell cell = cellIterator.next();
25
26                 switch(cell.getCellType()) {
27                     case Cell.CELL_TYPE_BOOLEAN:
28                         //System.out.print(cell.getBooleanCellValue() + "\t\t");
29                         list.add(cell.getBooleanCellValue());
30                         break;
31                     case Cell.CELL_TYPE_NUMERIC:
32                         //System.out.print(cell.getNumericCellValue() + "\t\t");
33                         list.add(df.format(cell.getNumericCellValue()));
34                         break;
35                     case Cell.CELL_TYPE_STRING:
36                         //System.out.print(cell.getStringCellValue() + "\t\t");
37                         list.add(cell.getStringCellValue());
38                         break;
39                     case Cell.CELL_TYPE_BLANK:
40                         if (row.getRowNum() != 0)
41                             list.add("");
42                         break;
43                 }
44                 if ((row.getRowNum() == 0) && (cell.getCellType() != Cell.
                    CELL_TYPE_BLANK)) {
45                     cellCount++;
46                 }
47             }
48             System.out.println();
49             storeFirstRowCells = cellCount;
50         }
51         file.close();
52     } catch(Exception e) {
53         e.printStackTrace();
54     }

```

```

55     return (String) sheet.getSheetName();
56 }

```

The Java script was made for two purposes. First, to create the database (structure) and second, to populate it with the 240 records to the emigration documents previously processed. The code above shows a function in Java that creates and activates an iterator through each row in each worksheet to populate the database.

Perl:

```

1 for my $worksheet ($workbook->worksheet('Ident')){
2     my ($row_min, $row_max) = $worksheet->row_range();
3     my @identificacaoEmigrante_cols = (0,1,5,11,10,6,8,3,4);
4
5     for my $row ($initialRowExcel .. $row_max) {
6         foreach my $col (@identificacaoEmigrante_cols) {
7             my $cell = $worksheet->get_cell($row, $col);
8
9             if (($col == 0) && (defined $cell && $cell->value() ne '')) {
10                #grava o idEmigrante no recordsOfDb
11                push(@recordsOfDb, $cell->value());
12            } elsif (($col == 0) && (!defined $cell || $cell->value() eq ''))
13                {
14                #grava no log e salta para próxima linha, pois sem
15                idEmigrante
16                #não há o que fazer
17                writeTxtFile("log_files", "log_emigrante", ">>", "—>
18                idEmigrante vazio ou não numérico.\tCélula ". ($row+1)."A
19                \n\n");
20                $idEmigranteNotExist = 1;
21                last;
22            }
23        }
24    }
25 }

```

The code above shows a piece of the Perl script to get the values of `identificacaoEmigrante` fields from the spreadsheet. Each row of the spreadsheet is read, always checking the cell value. In the Perl snippet code above, the cell `idEmigrante` is being checked if it is empty or non-numeric; if not, the record is stored in a list to be committed in the database. Otherwise, a log file is created with the error to be manually analyzed later. This type of validation is made at fields like `dtNasc`, `nomeEmigrante`, and location (`freguesia`, `concelho` and `distrito`). This is important, because, as mentioned previously, the original workbook contains several inconsistencies.

As said above, to circumvent this problem the log file created by the Perl script registers the line and the description of each error found. An example of this log file is shown in Figure 5.2.

```

--> idEmigrante vazio ou não numérico.  Célula 75A
--> dNasc (Data Nascimento) não definida ou incorreta.  Célula 115F
--> idEmigrante vazio ou não numérico.  Célula 130A
--> dNasc (Data Nascimento) não definida ou incorreta.  Célula 152F
--> dNasc (Data Nascimento) não definida ou incorreta.  Célula 163F
--> idEmigrante vazio ou não numérico.  Célula 194A
--> dNasc (Data Nascimento) não definida ou incorreta.  Célula 221F
--> idEmigrante vazio ou não numérico.  Célula 231A
--> idEmigrante vazio ou não numérico.  Célula 232A

```

Figure 5.2: Log file

In this way, 234 out of a total of 240 records were transferred completely and populate the

initial state of the database. So, all the items uploaded during the first stage, can be now consulted, edited, and explored through SGPE interface.

Chapter 6

Conclusion

The goal of this work was to develop an web-based application (SGPE) to upload into a computer database the data of the physical documents of the emigration focused in a Municipal Archive to later use these data to several purposes. The main aim is to use the information of the emigration documents to create virtual Learning Spaces as already mentioned in Chapter 1, but other important purposes also should be considered in the future.

With the collection of these data several social and cultural benefits can be promoted, such as:

- Potentiate the demand for cultural issues;
- Make available the documentary archive about the problematic of the emigrations in Portugal and in the world;
- Develop studies and research, in collaboration with scientific and academic institutions, to share the knowledge about the problematic of the emigrations;
- Promote seminars, conferences, debates and other activities related to the phenomena associated with population movements;
- Promote the research about the emigrants in the host territories and in their return to Portugal, treating questions about architecture, industry, commerce, philanthropy, journalism, arts, among others.

In fact, the cultural recognition of society can only be enhanced if studies showing the importance of the various historical subjects present in training the people and culture are developed. Thus, follow the route of the emigrants that depart to abroad is important, because this allows the recognition of the cultural trades, the difficulties in the adaptation, the prejudices that they suffered, among other relevant subjects. To sum up, the study of the emigrants is essential to understand the development of the destination areas as well as the influences that the emigration causes in the society in the return of the emigrant to his country.

To conclude, as mentioned in Chapter 1, the SGPE application can be useful for another city council. The study made and here reported was about the emigration in the 1960s and 1970s, using Fafe's files, but the application can be used by other Archives of Portugal for the collection and exploitation of the emigration documents from other eras.

Bibliography

- [1] Young, S., of Illinois at Chicago. Library. Special Collections Department, U.: Don't Throw it Away!: Documenting and Preserving Organizational History. Special Collections Department, University Library, University of Illinois at Chicago (1995) [6](#)
- [2] ico. Information Commissioner's Office: Guidance on data security breach management (1998) [13](#)
- [3] Matoria, R.K., Upadhayay, P.: Design and development of web-enabled databases in libraries with special reference to rdbms: Selection of tools and technologies. DESIDOC Journal of Library & Information Technology **22** (2003) [14](#)
- [4] Fowler, M.: Gui architectures (2006) [15](#)
- [5] Richard Ishida, W3C, S.K.M.B.: Localization vs. internationalization (2010) [15](#)
- [6] Ullman, L.: The Yii Book: Developing Web Applications Using the Yii PHP Framework. Self-published (2013) [16](#), [17](#), [18](#)