Universidade do Minho
Escola de Engenharia
Departamento de Informática

Ricardo André Araújo Neves

Emigration Tales

automatic generation of texts, from ontological descriptions of emmigration resources, to tell life stories

May 2022

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Ricardo André Araújo Neves

Emigration Tales

automatic generation of texts, from ontological descriptions of emmigration resources, to tell life stories

Master dissertation
Integrated Master's in Informatics Engineering

Dissertation supervised by
Pedro Rangel Henriques
Alda Lopes Gancarski

May 2022

AUTHOR COPYRIGHTS AND TERMS OF USAGE BY THIRD PARTIES

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Insert name

Ricardo Neves

ACKNOWLEDGMENTS

To thank the two supervisors of this dissertation, Professor Pedro Rangel Henriques and Professor Alda Lopes Gancarski, for their availability and immense help throughout the whole development of this project. To thank Professor António Aragão for his constant help during the system's deployment. To thank my family, friends and colleagues for the positive motivation they conveyed.

ABSTRACT

This document presents a Master's Thesis on Informatics Engineering, at University of Minho, and it is focused on an automatic generator of texts that is responsible for telling life stories. The main goal of this project is to develop a system that is capable of analyzing ontological descriptions in order to build one or more stories related to individuals or families who emigrated from and to certain regions of the globe, in a given time space. This generation is based on rules of transformation and application of the ontology's data in grammatical structures, thus forming sentences which, in turn, form a text in free speech.

This generation of texts in natural language is far from being linear, due to the existence of some conditionings that can make the text incoherent and/or inconsistent. An example of this is the person's biological gender, which has to be calculated - if not indicated at the outset - through his or her first name. Other conditionings that will change the configuration of the generated text are, for example, the number of people involved in the story, their ages, and the quantity (and quality) of information available.

To connect the stories of a multiple number of emigrants, it is also necessary to find at least one point of information in common between them all, in order to offer a higher level of realism and cohesion to the generated text.

This project also aims at enriching the visits to virtual museums related to the history of social life and migrations, such as, for example, the Museum of Emigration and Communities or the Museum of the Person.

Keywords: ontology, virtual-museum, story-telling, emigration, generator

RESUMO

Este documento apresenta uma Tese de Mestrado em Engenharia Informática, na Universidade do Minho, e está focada num gerador automático de textos responsável por contar histórias de vida. O objetivo principal deste projeto é de desenvolver um sistema que seja capaz de analisar descrições ontológicas de modo a contruir uma ou mais histórias relacionadas com indivíduos ou famílias que emigraram de e para certas regiões do Mundo, num dado espaço de tempo. Esta geração é baseada em regras de tranformação e aplicação de informação ontológica em estruturas gramaticais, formando assim frases que, por sua vez, formam um texto em linguagem corrente.

Esta geração de textos em linguagem natural está longe de ser linear, devido à existência de algumas condicionantes que podem tornar o texto incoerente e/ou inconsistente. Um exemplo desta variância é o gênero biológico da pessoa, que terá de ser calculado - se não for indicado à partida pela fonte - a partir do seu nome próprio. Outras condicionantes que irão alterar a configuração do texto gerado são, por exemplo, o número de pessoas involvidas na história, as suas idades, e a quantidade (e qualidade) da informação disponível.

Para haver uma conexão das histórias de múltiplos emigrantes, é também necessário encontrar, pelo menos, um ponto de informação comum a todos, de modo a oferecer um nível mais elevado de realismo e coesão ao texto gerado.

Este projeto também tem o objetivo de enriquecer as visitaos aos museus virtuais relacionados com a história da vida social e migrações, tais como, por exemplo, o Museu das Emigrações e das Comunidades ou o Museu da Pessoa.

Palavras-Chave: ontologia, museu virtual, conto de histórias, emigração, gerador

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

M

MMC Museum of Migrations and Communities.

MP Museum of the Person.

N

NLG Natural Language Generation.

O

OWL Ontology Web Language.

R

RDF Resource Description Framework.

1

INTRODUCTION

This document is focused on an automatic generator of texts that is responsible for telling life stories. The main goal of this project is to develop a system that is capable of analyzing ontological descriptions in order to build one or more stories related to individuals or families who emigrated to other regions of the globe, in a given time space. This story generation is based on rules of transformation and application of the ontology's data in grammatical structures, thus forming a complete text in free speech. This project also aims at enriching the visits to virtual museums related to the history of social life and migrations.

1.1 MOTIVATION

Portuguese emigration is not a recent fact but has always been present in the Portuguese society whose evolution has become stronger at the end of the XIXth century and during the third quarter of the XX century. Economic reasons, among others of social, religious and political nature, are the main cause for the large Portuguese dispersion in the five continents [Arroteia, 2001].

Today, Portugal is the country of the European Union with more emigrants in proportion to the resident population, with the number of Portuguese emigrants currently exceeding the two million mark. The United Kingdom is today the country where Portuguese people emigrate the most: 61.000 emigrants between 2013 and 2014. Switzerland, France and Germany are the next preferable destination countries of the flow. Outside of Europe, the main countries of destination for Portuguese emigration are part of the CPLP (Community of Portuguese Speaking Countries): Angola, Mozambique and Brazil, in this order [1].

The Museum of Migrations and Communities (MMC) [2] was created on July 12, 2011 and is installed on the ground floor of the Municipal House of Culture in Fafe, Braga. At the request of the City Council, an online platform was developed to support the physical

[1] ACM, Saber mais sobre as migrações portuguesas: https://www.acm.gov.pt/-/saber-mais-sobre-as-migracoes-portuguesas

[2] Museu das Migrações e das Comunidades - CM Fafe: http://www.cm-fafe.pt/conteudo?item=31299

museum, designated by Museum of Emigration and Communities [3], which offers its users a virtual visit to the space and its historical content. This museum aims to expose knowledge in migratory movements, and, in particular, Portuguese emigration.

On the other hand, the Museum of the Person (MP) [45] is a virtual and collaborative museum of life stories, founded in São Paulo in 1991. Since its origin, it aims to record and preserve true information about life stories. Here, the user can have access to the data he wants, as well as upload his own records to populate the database, being able to contribute with texts, images and videos. At the moment, the collection has more than 18.000 life stories, 60.000 photos and documents, and 4.000 stories sent by visitors.

In a Computer Science environment, *an ontology provides a conceptualization of a knowledge domain by defining the classes and sub-classes of the individuals (entities), the types of possible relations between them, etc., and plays a central role in the Semantic Web* [Ion Androutsopoulos et al., 2013]. The ontology data model can be applied to a set of individual facts to create a knowledge graph - a collection of entities, where the types and the relationships between them are expressed by nodes and edges between those nodes. For this reason, it is appropriate to say that an ontology is an interesting way to describe and store the information collected from both museums, being able to easily be fed with more content from other sources later.

Therefore, all the data present in the databases of both the MMC and MP platforms will have to be analysed, studied and filtered, in order to build a rich and viable dataset. All this information will be described through a complete ontology about people who have migrated to another location: their personal data, place of origin and destination, among other details that will have to be reviewed in due course.

To have an idea about the kind of data that will be treated, there's an online database provided by CEPESE (Center for Population, Economics and Society Studies) [6] that provides thousands of records of passport holders and various informations related to them.

With the information saved and properly organized in the ontology, it is then possible to focus all the lights on the implementation of the automatic text generator, based on ontological data. To do so, it is essential to outline a series of grammatical rules, so that the text produced is as clear, fluent and grammatically rich as possible. To add a layer of interaction to the generated text, it's also possible to search for common spots of information between a couple of people and mix their stories together.

[3] Museu das Comunidades e da Emigração: https://epl.di.uminho.pt/ miguelcosta/EL/Code/Museu/

[4] Museu da Pessoa: https://museudapessoa.org/

[5] Núcleo Português do Museu da Pessoa: https://npmp.epl.di.uminho.pt/

[6] CEPESE: A Emigração de Portugal para o Brasil: http://www.remessas.cepese.pt/remessas/mod/itsdatabase/view.php?n=1&v=

In a very simplified way, a text can be considered as an articulated sequence of phrases that form a meaningful unit [Telmo Móia, 2014]. The texts vary widely in form and length, ranging from a simple phrase in an advertising speech to a novel, a story, scientific article, decree-law or sentence in a court. Among the many properties that characterize texts as linguistic objects, it's interesting to highlight these three ones: the grammar of their constituent elements (phrases), structural cohesion and conceptual coherence of the combination of these elements.

> *Story generation, storytelling and story understanding are examples of a phenomenon called narrative intelligence, which is defined as an entity's ability to organize and explain experiences in narrative terms, comprehend and make inferences about narratives that are told, and produce affective responses such as empathy* [Li Boyang et al., 2013].

1.2 OBJECTIVES

The main goal of this project is to develop a system that is capable of analyzing ontological descriptions in order to build one or more stories related to individuals or families who emigrated from and to certain regions of the globe, in a given time space. This generation is based on rules of transformation and application of the ontology's data in grammatical structures, thus forming sentences which, in turn, form a text in free speech.

The result of this project could become a valuable asset to enrich the visits to virtual museums related to the history of social life and migrations, such as the Museum of Emigration and Communities or the Museum of the Person.

The development of this project is divided in two different axis: a theoretical axis, based on the logical aspects of the problem, and a production axis, based on the production of the system itself.

In the theoretical axis, the objective is two-fold:

- Design a complete ontology to store and relate all the information;
- Identify a set of transformation and grammatical rules to automatically generate perceptible and clean texts.

In the software engineer axis the objective is three-fold:

- Build the ontology schema;
- Describe the data of the museums through the ontology;
- Plan and develop a system that automatically generates a text given the information stored in the ontology.

1.3 RESEARCH METHODOLOGY

To accomplish this master project, an iterative methodology based on documents' research, solution proposal, implementation and testing will be followed. To carry out this approach, the working plan, divided in the next topics, is:

- Do a general research about the Portuguese emigration to other places of the globe;
- Search about databases with information that can be valuable to feed the system;
- Do an in-depth study on automatic ontology-to-text generators that are already available;
- Design a complete ontology, which will serve as a slot for the extracted data;
- Export the information that was found previously and populate the ontology;
- Define the transformation rules;
- Implement the automatic text generator;
- Test the system and do the required adjustments.

1.4 RESEARCH HYPOTHESIS

With this project, it is intended to demonstrate that it is possible to generate the life story of a selected person, collecting data from an ontology, previously populated with filtered information about several individuals.

Particularly, in order to prove this hypothesis, a focus is made on databases of individuals who emigrated to other parts of the world, in a certain period of time.

1.5 DOCUMENT STRUCTURE

In this document, it was already presented, in the first topic, an Introduction [Section 1] to the work, going through the topics of the motivation that led to the realization of the project, the objectives outlined initially, the research approach methodology, and the research hypothesis.

Then, it is presented the State of Art [Section 2], where a few essential points are numbered regarding the project, based on bibliographic articles. This section is divided into five different topics: ontologies [Section 2.1], from ontology to text [Section 2.2] including an explanation about grammatical genders [Section 2.2.1], story planning and generation [Section 2.3], grammatically correct texts [Section 2.4], and related work [Section 2.5].

In third place, there is the Proposed System Architecture [Section 3], which discusses the initial planning of the system, as well as a simple scheme of the various components that compose it, detailing a little about each one, and the possible technologies that can be implemented.

Finally, the Conclusion [Section 6] of the document, where the reports that have happened during the writing of the document are described, how the project was developed, and some perspectives for future work. Here, it is presented the Work Plan [Section 6.2], with details of each of the phases that make up the entire project, a Gantt diagram to outline each of these phases, and a brief report that specifies whether this planning is being carried out, or not, explaining why.

At the end of the document, there is a Bibliography with the list of articles that helped in the writing of this document. It is important to note that each expression taken from these bibliographic pages is properly marked and linked to its correct reference.

2

STATE OF ART

The topic of State of Art serves as a starting point for the proposed project. During this chapter, the theoretical notions surrounding the project are presented, resulting from a dense introductory study and research. Here, a series of topics that relate to this project will be addressed: ontologies and their syntaxes, conversion of ontological data to text, grammatical gender, planning and story generation, natural language generation, and grammatically correct texts, namely at the grammatical level, coherence and cohesion. At the end, some existing systems will be listed, which are also based on the generation of natural language, describing and comparing each one to the Emigration Tales project.

2.1 ONTOLOGIES

In a Computer Science environment, *an ontology provides a conceptualization of a knowledge domain by defining the classes and sub-classes of the individuals (entities), the types of possible relations between them, etc., and plays a central role in the Semantic Web* [Ion Androutsopoulos et al., 2013]. The ontology data model can be applied to a set of individual facts to create a knowledge graph - a collection of entities, where the types and the relationships between them are expressed by nodes and edges between those nodes.

> *First, ontologies share a common conceptual structure which developers can work on, building shared and reusable knowledge bases. Second, they facilitate the inter-operability and merging of data (mash-ups), which enable the creation of powerful and intelligent computational applications* [Seiji Isotani and Ig Ibert Bittencourt, 2015].

Also according to this source, the design and the use of ontologies have always been part of the Semantic Web and, over the past decade, they have proven to be one of the key technologies in creating applications, as they are very suitable for handling large amounts of information in an intelligent way.

In this era of the data web, where information and knowledge are fragmented in the network and resources are constantly evolving, the development of systems based on open data cannot follow the paradigm in which databases are static and created for a very specific and restricted problem/domain. Currently, the best approach is to develop connected, highly shareable databases that allow inter-operability and the possibility to deal with the constant accumulation of knowledge available on the Web.

Formally, and according to Atanas Kiryakov (2006), we can define an ontology as a relationship of four elements, fundamental for the creation of a structure that represents the knowledge of a domain:

- C - set of classes that represent the concepts in a given domain
- R - set of relationships or associations between concepts
- I - set of instances derived from classes
- A - set of axioms of the domain, which serve to model restrictions and rules inherent to the instances

> *In recent years, the growth of data published on the web according to Semantic Web formalisms and data models (e.g. RDF(S) and OWL) has been exponential, leading to more than 30 billion Open Data cloud, which contains a wide range of factual knowledge that is very interesting to many applications and for many purposes* [Philipp Cimiano et al., 2013].

RDF Schema provides a data-modelling vocabulary for RDF data and a semantic extension for RDF. *An RDF graph is made up of semantic triples consisting of a subject, predicate and object* [1]. *It provides mechanisms for describing groups of related resources and the relationships between these resources. These resources are used to determine characteristics of other resources, such as the domains and ranges of properties* [2]. Also according to this source, RDF Schema differs from many other such systems in that, instead of defining a class in terms of the properties its instances may have, it describes properties in terms of the classes of the resource to which they apply. This is the role of the domain and range mechanisms described in this specification.

For example, we can define the :artist property to have a domain of :Albums and a range of :Person, whereas another type of system might typically define a class :Album with an attribute :artist of type :Person. There is a number of concrete syntaxes for RDF, such as Turtle, N-Triples, RDF/XML and JSON-LD, but for this project, and due to a previous knowledge of this syntax, the ontology is built with Turtle. A Turtle document is a textual representation of an RDF graph. *It allows writing down an RDF graph in a compact textual form.*

[1] W3C. Rdf 1.1 turtle. 2014b: https://www.w3.org/TR/turtle/

[2] W3C. Rdf schema 1.1. 2014: https://www.w3.org/TR/rdf-schema

2.2 FROM ONTOLOGY TO TEXT

In the *Emigration Tales* system, the domain is applied to the information of the group of people who emigrated to another point in the world, modeled as RDF data in the ontology, using the RDF Schema vocabulary.

For example, it is possible to store an RDF predicate about the fictional person João Ferreira, with certain information about him, like his name, birthplace and date of birth, like in the Figure 1.

```
:Joao_Ferreira rdf:type :Person,
               :name "João Ferreira",
               :birthPlace "Vila do Conde",
               :birthday "27-07-1925";
```

Figure 1: RDF example of emigrant João Ferreira

With only this excerpt from the ontology, we already have enough information to create a phrase like *"João Ferreira nasceu em Vila do Conde a 27 de Julho de 1925."*. In this case, an ontology is required to keep the relevant data of certain individuals, including the origin and destination of their migrations, as well as the date of departure and arrival. Thus, it is also possible to store this information in the same way as the previous one, like in the Figure 2.

```
:Joao_Ferreira :destination "Belo Horizonte",
               :origin "Lisboa",
               :departure "14-06-1959",
               :arrival "03-07-1959";
```

Figure 2: RDF example of the João Ferreira's migration

To show our system is capable of integrating life stories of several individuals who share a certain common category, we add another example of a Person. Thus, we store the information of a new individual named Maria Vieira, also of the Person type, as can be seen in Figure 3. Like our previous individual João, she also has the same origin and destination , with equal departure and arrival dates.

```
:Maria_Vieira rdf:type :Person,
              :name "Maria Vieira",
              :birthPlace "Viana do Castelo",
              :birthday "09-11-1929";
              :destination "Belo Horizonte",
              :origin "Lisboa",
              :departure "14-06-1959",
              :arrival "03-07-1959";
```

Figure 3: RDF example of emigrant Maria Vieira

In order to obtain all the people who left Lisbon with our individual João towards the same destination, at the same time, it is necessary to apply a SPARQL query to the ontology. Thus, as an answer we will obtain the records of João himself (who will be ignored) and Maria, which allows us, if desired, to combine the stories of these two people.

```
SELECT ?name ?birthPlace ?birthday
?destination ?origin ?departure ?arrival WHERE {
     ?p a :Pessoa.
     ?p :name ?name.
     ?p :birthPlace ?birthPlace.
     ?p :birthday ?birthday.
     ?p :origin "Lisboa".
     ?p :departure "14-06-1959".
}
```

Figure 4: SPARQL query to obtain the migrants that left Lisbon in June 14th 1959

With the combination of all these data obtained so far, we can generate a simple text with the information from these two individuals: *João Ferreira nasceu em Vila do Conde a 27 de Julho de 1925. Ele partiu de Lisboa a 14 de Junho de 1959 com destino a Belo Horizonte, onde chegou 19 dias depois, a 3 de Julho de 1959. Durante a viagem, conheceu Maria, uma emigrante que também tinha o mesmo destino que João. Maria Vieira nasceu em Viana do Castelo a 9 de Novembro de 1929.*

2.2.1 *Grammatical gender*

As we can see in the text reproduced above, there are two individuals of different biological genders: male, in the case of João, and female, in the case of Maria. Unlike the English language, used in the vast majority of story generators, Portuguese grammar lexically distinguishes the gender of words, as strange as it may seem to non-Portuguese speakers.

Grammatical gender plays a key role because it affects the generation of determiners and pronouns. It also influences the inflection of nouns and verbs [Diego Moussallem et al., 2018].

For example, in the text above, we can see a reference to "uma emigrante". This determinant takes the value of "uma" because it refers to Maria, who is female. However, if the roles were reversed, and a reference was made to João, the value of the determinant would be "um", since the individual is male.

In the case of the English language, the determinant "an" would be used for both the cases of Maria or João, since there's no such difference regarding biological gender: "an emigrant".

These lexical rules are not exclusive to determinants / pronouns. This gender differentiation is also applied to verbs (particularly, in the past participle) and adjectives, depending on the individual's biological gender.

The verbs articulated in the past particle depend on the gender of the individual to which they refer. For example, in the case of the verb "emigrar" ("emigrate" in English), its expression in the past particle is "emigrado" if the subject is male, and "emigrada" if the subject is female.

Adjectives associated with an individual are also changed under these rules. For instance, the adjective "bonito" ("beautiful" in English), is written in this way if the subject is male, but, otherwise, it must be "bonita", referring to a female individual.

With this being said, we can begin to notice a certain trend: if the subject is male, all determinants/pronouns/verbs/adjectives end up in the letter 'o', while, on the other hand, they end up in 'a' if the individual is female. This assumption is wrong and should not be used as a universal rule in all cases whatsoever.

Another concept inherent in the Portuguese language and which does not exist, in any form, in the English language, is that the common names - objects, animals, plants, etc. - also have an intrinsic gender. For example, in the expression "o navio" ("the ship" in English), the determinant used is 'o', since the object "navio" has a male gender associated with it. Again, this is a grammatical feature that is both interesting and complex, so it requires additional study - and some practice - for non-Portuguese speakers.

However, this last topic will not be explained in detail since it does not interfere or require additional work for the development of the system. Only words that are directly related to the individuals in the database - determinants, pronouns, verbs and adjectives - have to be worked on in order to respect the grammatical rules of the Portuguese language.

Now, there are two possibilities to calculate the gender of the individual (or individuals) by which the story will be shaped around: the first possibility is, at the outset, the source database already returning the person's gender, whether it is a man or a woman. This would be the simplest form, without a shadow of a doubt, whereas it would not need additional calculations, since the information is already there. However, our system does not control all the details of the source databases at all, so the second possibility will have to be taken into account, sooner or later.

Therefore, the second possibility is to calculate the individual's gender using the individual's first name. In both English and Portuguese, it's usual that a person's first name is adjusted in relation to his/her gender. So, this is a very accurate way to calculate this detail.

One possible approach to take is to check the last letter of the name. As has already been said, as a rule, men's names end in 'o', while women's names usually end in 'a'. Nonetheless, this "rule" is far from perfect, so there are countless cases of names that do not follow it. Therefore, the path will have to go through a much more assertive approach.

For this, it is necessary to find a list of names separated by gender, like in the Table 1. The greater the number of names available in this list, the more possible names are covered by this check, increasing the degree of system reliability.

Male	**Female**
André	Andreia
João	Joana
Miguel	Maria
Paulo	Paula
Pedro	Patrícia
Xavier	Sofia
(etc...)	(etc...)

Table 1: Examples of male and female names

Thereby, the approach involves comparing the person's first name (if available from the original database, of course) with the available list of names, associating the character with one of the known biological genders. With this, it is possible to shape the textual aspects so that the story is coherent with the gender of the individual (or individuals) involved.

If it is still not possible to calculate the actual person's gender, there is the possibility of shaping the text in a different way, so that this detail is not used. For example, instead of having

"Ele partiu de Lisboa a 14 de Junho de 1959 com destino a Belo Horizonte (...) Durante a viagem, conheceu Maria, uma emigrante que também tinha o mesmo destino",

we would obtain an excerpt like

"Partiu de Lisboa a 14 de Junho de 1959 com destino a Belo Horizonte (...) Durante a viagem, conheceu Maria, emigrante que também tinha o mesmo destino".

As we can see, in the second passage, there is no textual spot that makes reference to the gender of any of the intervening characters. However, this is an approach that we want to avoid at all costs, since it removes textual components that give life to the text, making it less fluid and interactive for the reader.

2.3 STORY PLANNING AND GENERATION

The challenge of the automated story generation is to automatically select a sequence of actions or events, performed by characters, that meet a set of criteria and forming a story. In the vast majority, a story can be divided into three different core parts:

1. introduction,
2. body and
3. conclusion.

The introduction refers to the initial situation of the story where the main elements of the narrative are presented: characters, temporal and physical space, and early plot. The introduction is very important because it gives the reader the necessary background information, building the foundations for the development of the plot that is coming next. In our case, this section presents the individual that the user has chosen previously in order to generate a story about his/her life. The information and characteristics of the chosen person depends, of course, on the quantity and quality of the data made available by the initial database.

The body of the narrative accounts for the bulk of the story. It's where the whole plot of the story takes place, and where the characters carry out actions, sometimes interacting with others, forming a chain of events. For the life story's body of the chosen person, a series of actions carried out by the protagonist are generated automatically, mostly written in the perfect and imperfect past tense. Whenever possible, and at the request of the user, the story of the main individual will be crossed with life stories of other characters present in the database. If the main individual made his migratory trip in 1920, for example, then it makes sense to relate his story with other people who made the same trip, at the same time, to the same destination (or geographically close).

The conclusion, also known as the outcome, resolution or denouement, is where the whole story stabilizes and refers to the fate of the characters, tying up all the loose ends. It is in this final section that the story of the main individual is completed, ending all the actions that were performed during the body. If the individual was related to other characters during the narrative, their stories should also be closed (if they were not previously closed already), thus ending the text.

The vast majority of story planning and case-based reasoning story generation systems require a domain model that provides knowledge about characters and places in a fictional world and a set of possible actions that those characters can perform [Li Boyang et al., 2013]. However, there are a few developed story generation systems that do not need a knowledge base to create a text. Examples of these type of systems are *SayAnything* [Reid Swanson and Andrew S. Gordon, 2012] and *MAKEBELIEVE* [Hugo Liu and Push Singh, 2002], which will be briefly covered later in the Related Work [Section 2.5].

A very common method for developing a story generation system is the production and use of plot graphics. *A plot graph defines the space of legal story progression and ultimately determines possible events at any given point in time* [Li Boyang et al., 2013]. They assemble a temporal schema with the sequential structure of the story that the author intends to create, forming a sequence of interconnected and relative events.

For instance, a plot event of a character entering his home must be preceded by other plot events where the person arrives at home and picks up his/her keys, opening the main door. In this very simple example, the action-consequence pair occurred very quickly, where the character took the keys from his/her pocket, opened the door and entered his house. However, there may (and certainly will be) cases where a character's action only has consequences/repercussions much later in the story.

We will show next an example of a plot graph in Figure 5, which depicts a fictional moment of when João arrives at home. Here, we can identify 3 components that make up the plot chart: events, precedent constraints and mutual exclusions.

Events are expressions, phrases or a set of phrases that tell an action of a character in the story. This action will trigger a series of new possible events, and may involve more characters from the story. An event can range from a simple action such as "entering the house" to an action with harmful consequences, which will affect the entire course of the story, such as killing another character.

When dealing with a plot graph, it is ideal to connect the possible sequences of events. Thus, precedent restrictions are used, forming a flow chart and where we can differentiate all possible paths of action of the character. From one event, we may have several precedent constraints going out to other events. Likewise, one event can be preceded by several other events. For instance, in the chart below [Figure 5], we can see that the only event that happens after "getting out of the car" is "taking the keys out of your pocket". However, also depending on whether "João's mother is at home", there is a bifurcation of events, where "João will open the door" or "his mother opens the house door" for him.

Finally, we can verify the existence of mutual exclusion links. As already said, and we reinforce it again, each action of the character can have greater or lesser consequences. Thus, two events are linked by a mutual exclusion link where, if the first is carried out by the character, the second will no longer be an option later in the history. Using the example below, João will only realize that he lost his wallet if he got home in a taxi. If João came home with his own car, it doesn't make sense that he lost his wallet in the taxi, since he never got into one in the first place. Likewise, if the taxi driver leaves the wallet at the police station, it means that João won't notify the police that he has lost his wallet - since it's already there to be picked -, creating another mutual exclusion on the story's plot graph.

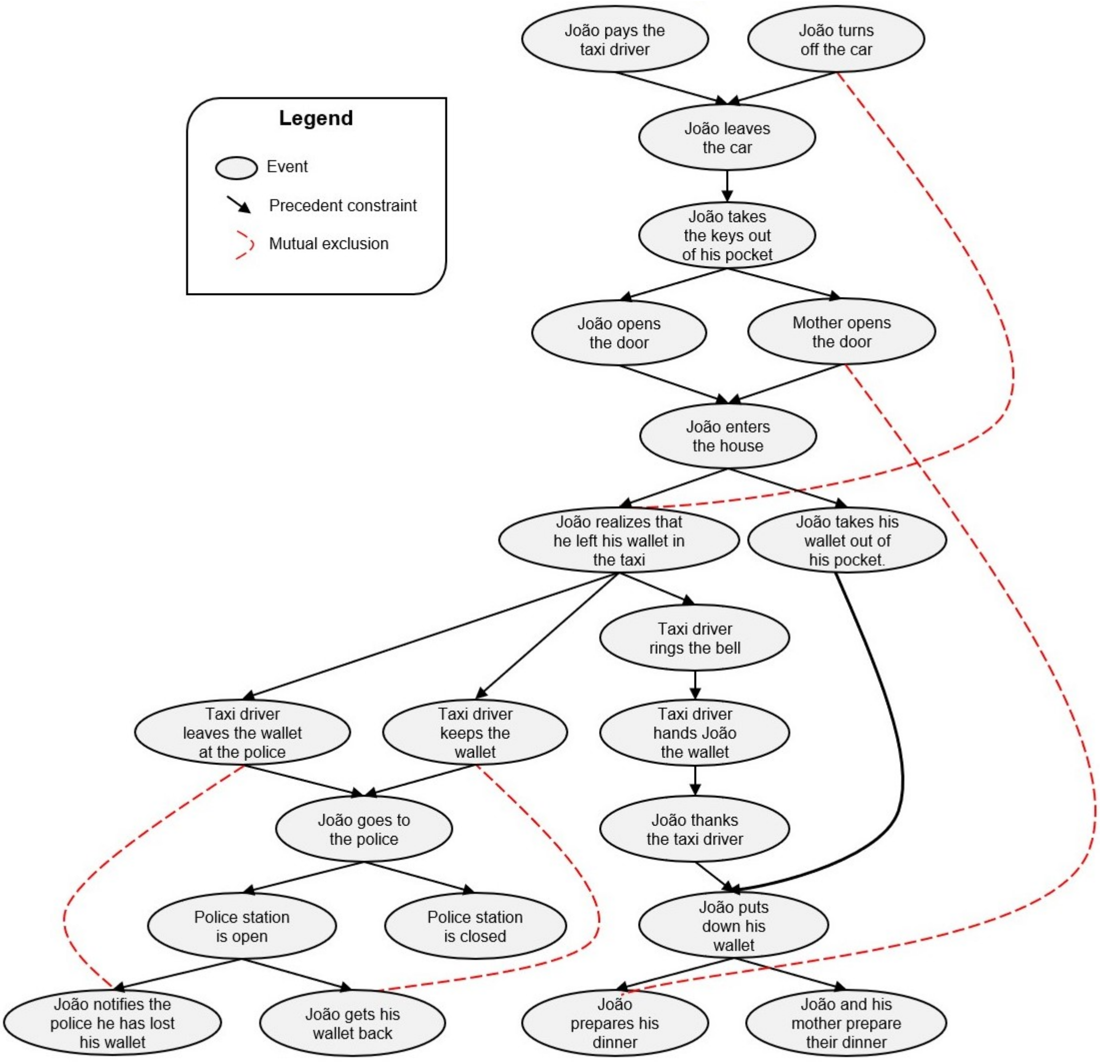

Legend
Event
Precedent constraint
Mutual exclusion
João pays the taxi driver
João turns off the car
João leaves the car
João takes the keys out of his pocket
João opens the door
Mother opens the door
João enters the house
João realizes that he left his wallet in the taxi
João takes his wallet out of his pocket.
Taxi driver rings the bell
Taxi driver leaves the wallet at the police
Taxi driver keeps the wallet
Taxi driver hands João the wallet
João goes to the police
João thanks the taxi driver
Police station is open
Police station is closed
João puts down his wallet
João notifies the police he has lost his wallet
João gets his wallet back
João prepares his dinner
João and his mother prepare their dinner

Figure 5: Plot graph example

A very practical example of using these plot graphics is the electronic video-game *Detroit: Become Human*, produced by the French developer Quantic Dreams. This is a game divided into 32 chapters where, in each one, the player has the possibility to choose one of several paths that the story offers. Therefore, depending on the choices made over the entire length of the game, players will reach different endings of the story.

An interesting feature that was designed into the game is the possibility for the player, when he finishes playing a chapter, to see the plot graph of this piece of the story. Here, the player is able to see all the forks in the story, the personal decisions he made, the paths he could have taken and the possible endings of the chapter. In Figure 6 below[3], we can see the plot graph of the first chapter:

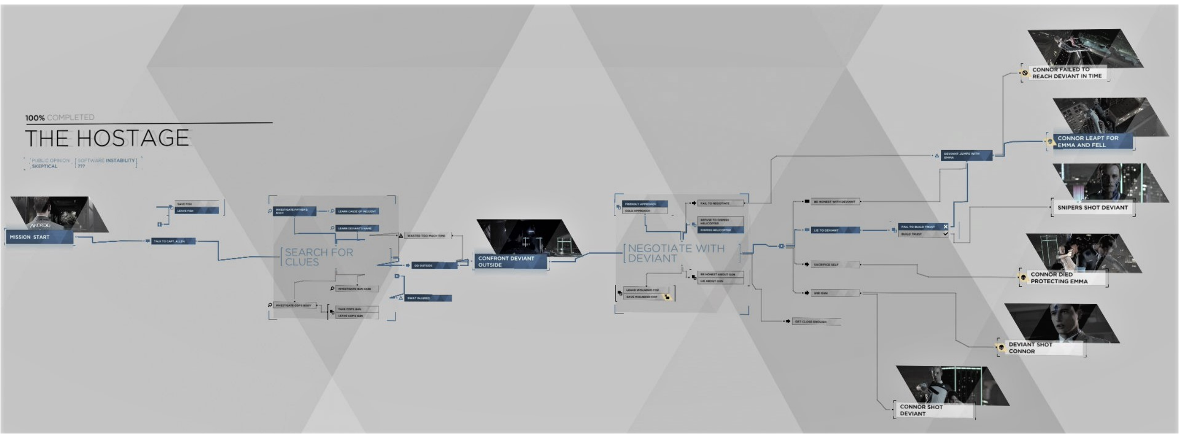


Figure 6: *Detroit: Become Human*'s first chapter plot graph

As we can see in the chart above, all players start at a single starting point, the beginning of the story. A little later, begins to appear the first forks in history, where the player is called upon to make the first personal choices. Here, the first mutual exclusion relationships of the chapter are created, disabling certain paths later on.

However, in the center of the graph, we can see that all possible paths that the player could have taken converge in a single common event in the chapter. Although all players are at the same point, mutual exclusions are already created and will take effect later.

Finally, on the right, there is another bifurcation of events, where the player is again called to take his own initiatives. This time, depending on all the choices he/she made throughout the chapter, we can analyze in the picture that there are 6 possible ends, which will have different consequences for what remains of the game.

Thus, the similarities that can be found between this electronic game and many story generators are that both have written, at the beginning, a predefined plot with all the possible paths, and have to keep track of the decisions that are made, in order to maintain the same coherency along its entire length. The difference lies in the fact that, in the video-game, the choices made are dependent on the player's decisions, while the story generators do not have this external element, so they are the ones who decide, in a random but controlled way, the path that the story will take.

[3] Figure 6 taken from the videogame *Detroit: Become Human*

With all this being said, it's fair to say that a well thought and coherent chart helps to visualize all the relationships between events in a simple and quick way to read. The greater the number of possibilities for pre-defined events in the story generation system, the greater and more complex this type of plot graphics will be.

2.3.1 *Natural Language Generation*

> *NLG is the sub-field of artificial intelligence and computational linguistics that is concerned with the construction of computer systems that can produce understandable texts in English or other human languages from some underlying non-linguistic representation of information.* [Ehud Reiter and Robert Dale, 2000]

Still according to this source, NLG techniques can be used to:

- generate textual weather forecasts from representations of graphical weather maps,
- summarize statistical data extracted from a database or spreadsheet,
- explain medical information in a patient-friendly way,
- produce answers to questions about an object described in a knowledge base.

This list of possible applications of NLG systems is only indicative and is, by no means, complete. In the case of our project, the NLG technique serves the purpose of building a life story about a certain individual.

Depending on the purpose and detail of each NLG system, most of them adopt a pipeline architecture, which may vary slightly from system to system. To illustrate one of these architectures, take the example of the *NaturalOWL* project [Ion Androutsopoulos et al., 2013], divided into three key stages, that can be generically, but never strictly, adopted into the Emigration Tales system, illustrated in the Figure 7:

1. document planning,
2. micro-planning and
3. surface realization.

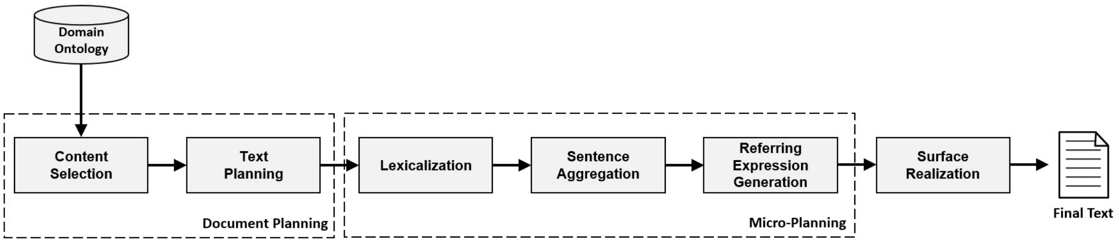


Figure 7: NLG process

First of all, document planning is a junction of two minor phases: content selection, where the system filters and selects the information to be used later, and text planning, where it plans the text structure that will be generated.

During content selection, the system first retrieves, from the ontology, all the statements that are relevant to the text, converting them into message triples, which are easily expressible as sentences. Next, in the planning stage, *the text planner orders the message triples, in effect ordering the corresponding sentences* [Ion Androutsopoulos et al., 2013].

The next stage of micro-planning consists of three sub-steps: lexicalization, sentence aggregation, and generation of referring expressions.

Firstly, during the lexicalization step, *NLG systems usually turn the output of content selection (the message triples) into abstract sentence specifications* [Ion Androutsopoulos et al., 2013]. In the case of more robust projects, this stage can be further developed, where *the domain author may specify one or more template-like sentence plans to indicate how message triples involving every property can be expressed* [Ion Androutsopoulos et al., 2013].

Next, the sentence aggregation is responsible for merging a certain number of phrases - usually around 2 and 4 - that refer to the same subject, generating a more user-friendly text and easier to read, never losing its precious coherence. For instance, in some museum contexts, the system must be able to set this parameter to 3 or 4 sentences, in order to generate a reasonable text for adult visitors, able to understand somewhat long texts with the least of difficulties. Whereas, a value of 2 is best suited for children visitors, where the text must be as clear as possible, with shorter and simpler expressions/phrases. In our Emigration Tales project, this number is usually set around 2 and 3, except for some rare exceptions, where this number may rise, best balancing the degree of complexity of the text and providing a better reading experience for the end-user. What follows is an example with the parameter set to 2:

"Ele nasceu em Lisboa. Ele nasceu em 1937." -> "Ele nasceu em Lisboa, em 1937."

The last step of this micro-planning stage is the generation of referring expressions, aggregating smaller sentences into longer ones, where the subject's name is removed from the phrase, being replaced by a reference to him/her/itself. Depending on the context, it may be better, for instance, to use the name of the subject (e.g., "João"), a pronoun (e.g., "ele"), a demonstrative noun phrase (e.g., "esta pessoa"), etc. This topic is detailed more deeply later in this document, specifically in the next section [Section 2.4], addressing the issues of textual cohesion and referential chains.

Finally, for the third and last stage of the NLG process, in some systems, *the sentences at the end of micro-planning are under-specified: for example, the order of their constituents or the exact* *forms of their words may be unspecified. Large-scale grammars or statistical models can then be used to fill in the missing information during surface realization* [Ion Androutsopoulos et al., 2013].

By contrast, at this point, other projects - most template-based systems - have already an ordered and aggregated set of sentences, making the stage of surface realization simpler and faster. Here, this is mostly a mechanism of converting a set of phrases into a final flowing text, adding the necessary punctuation and capitalization, thus ending the process of its generation.

2.4 GRAMMATICALLY CORRECT TEXTS

In a very simplified way, a text can be considered as an articulated sequence of phrases that form a meaningful unit [Telmo Móia, 2014]. The texts that we find all over the place everyday can vary greatly in form and length, ranging from a simple and short phrase, like in an advertising speech (e.g., "O que é Nacional é bom."), to a more extensive and elaborated text, for instance, a scientific article, a novel, a decree-law or a court sentence.

Among the many properties that characterize texts as linguistic objects, it is smart to highlight three: the grammaticality of their constituent elements (phrases), the structural cohesion, and the coherence of the combination of these elements, as shown in the Figure 8.

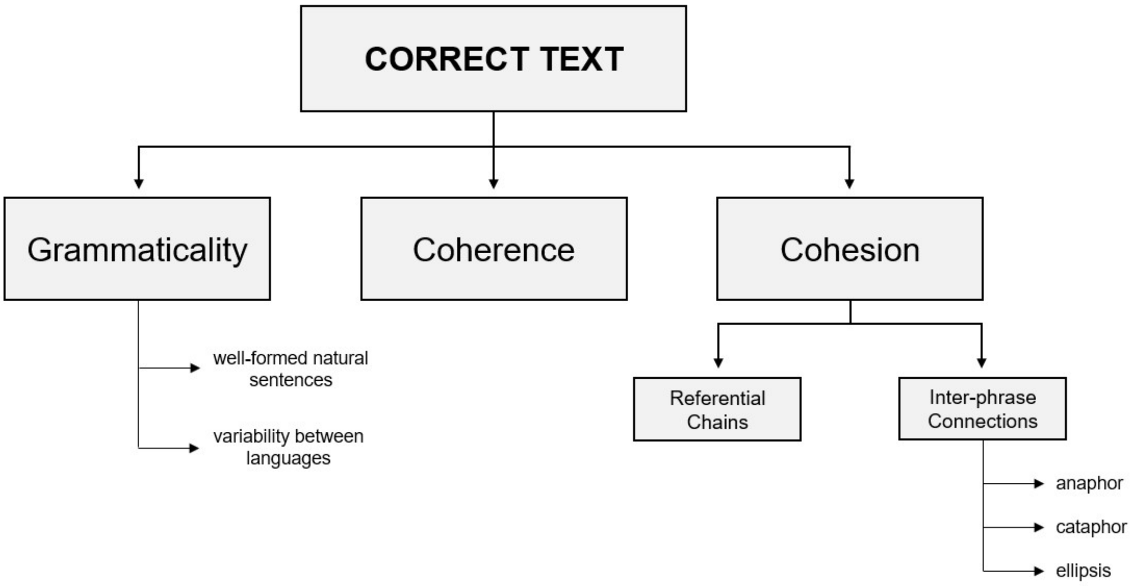


Figure 8: Correct text diagram

2.4.1 *Grammaticality*

In the linguistic prism, grammaticality is determined by the conformity of a sentence to the rules defined by the specific grammar of a language. The notion of grammaticality rose alongside the theory of generative grammar, which primary aim is to formulate rules that define well-formed, grammatical sentences. Although, a sensitive issue regarding grammaticality is the variability within a certain language. For example, some sentences are grammatically correct for one regional or social variety, but are incorrect for another.

With this being said, it is essential to remember that natural languages are systems of regularities. *The existence of (combination) rules is not something imposed from the exterior, but an intrinsic characteristic of all languages* [Telmo Móia, 2014]. Therefore, any speaker of the Portuguese language recognizes this expression

"Um sistema operativo é um programa ou um conjunto de programas cuja função é gerenciar os recursos do sistema (...)" (in Sistema Operativo - Wikipedia)

as part of their language and the sequence

"Um sistema operativo um programa é função cuja ...."

as an illegitimate combination of the same Portuguese words.

Grammaticality and its opposite (agrammaticality) are scalar concepts. A combination can range from full acceptability and naturalness, as in the first example above, to total rejection, as in the second one. But, in between, there are linguistic productions that have a greater or lesser degree of strangeness for most speakers. These are areas of instability in the linguistic system, which is in constant, however very slow, transformation.

"Um sistema operativo é um programa ou conjunto de programas cuja função ***é de*** *gerenciar os recursos do sistema...*
"Um sistema operativo é um programa ou conjunto de programas ***que a sua*** *função é gerenciar os recursos do sistema...*
"Um sistema operativo é um programa ou conjunto de programas ***cuja*** *sua função é gerenciar os recursos do sistema...*

It should now be noted that certain grammatical anomalies - although they should be avoided - do not compromise, or do little, the transmission of information and the unity of meaning that the text is supposed to constitute. Other anomalies, however, can have a more negative impact, like the second example we've seen before on this topic.

Let's now consider the issues of textual coherence and cohesion. In a very simplified way, we can consider that coherence is established on a conceptual plane, involving the possibility of finding meaning in statements, so that the reader interacts with the text in a logically correct way; and cohesion is established on a more structural/grammatical plane, involving the establishment of lexical links between different elements and sub-parts of the text, holding it together

2.4.2 *Coherence*

First of all, we talk about coherence. Each word has its individual meaning, but when they relate, they can create another completely different meaning. The same logic applies to sentences, and paragraphs. Each of these elements has an individual meaning and a type of relationship with the others. When these relations are done correctly, the reader gets a message, an understandable semantic content. Coherence is the factor that enables the understanding of this message transmitted by the text. As the text is read, ideas are concatenated, forming a chain of understanding. A text that obeys consistency conveys a logical relationship of ideas that complement and do not contradict each other, giving it meaning.

2.4.3 *Cohesion*

Besides being coherent, a text also needs to be cohesive. Cohesion concerns all the means by which, in a text, the connection between its components (words, expressions, phrases, paragraphs) is processed, in order to correctly convey the idea presented. When the text is incoherent, it harms the communication process. Relative to cohesion, it's pivotal to differ two important pillars: the referential chains and the inter-phase connections.

The referential chains are interpretive dependencies that are established between elements of the discourse, often at a great distance, allowing expressions not to be repeated and, therefore, that the statements are more economical [Telmo Móia, 2014] and lighter for the reader.

"Depois de um momento de reflexão, ***os seus amigos*** *lá aceitaram a sua decisão de partir, prometendo que voltaria para visitar todos* ***os seus amigos****."*
"Depois de um momento de reflexão, ***os seus amigos*** *lá aceitaram a sua decisão de partir, prometendo que voltaria para {****os visitar/visitar todos estes****}."*

Each language has its own set of expressions - like *os* or *estes* - whose interpretation is not autonomous. This interpretation depends on the capacity of association between this expression and the element of the speech. Thus, before using any referential expression, it is necessary to pay attention to whether the subject is well implicit. Otherwise, the referential chain will be deformed, giving rise to cases of grammatical anomalies and causing a feeling of strangeness to the reader, which is never recommended under any circumstances.

Most of times, the subject to which the reference is made precedes the non-autonomous expression (anaphor), as can be seen in the previous example. However, the subject can be found after the referential element (cataphor), as can be seen in the example below. Nonetheless, it is important to pay attention and be brief in exposing the subject to which the reference is made, so as not to deform the referential chain.

*"O João reuniu-**os** e contou - Eu vou emigrar brevemente, mas voltarei para **vos** visitar regularmente. - Os **seus amigos** refletiram e tristemente aceitaram a decisão."*

Another type of referential chain can occur, where there is no referential element in a second sentence, but the subject continues to be interpreted anaphorically (ellipsis).

"O João ganhou coragem e {o João} contou a notícia."

The second pillar, also very important for textual cohesion, is *the possibility of establishing links of meaning between phrases, technically called inter-phrase connections* [Telmo Móia, 2014]. These connections can be anterior-posterior, cause-effect, object-end, part-whole relationships. Some examples follow.

- causality-effect links:

*"O João precisava de contar a notícia. **Consequentemente**, reuniu os amigos em casa."*
*"O João precisava de contar a notícia, **pelo que** reuniu os amigos em casa."*
*"O João reuniu os amigos em casa **porque** precisava de contar a notícia."*

- contradition links:

*"**Apesar de** terem aceitado a decisão do João, os amigos ficaram bastante tristes."*
*"Os amigos aceitaram a decisão do João. **No entanto**, ficaram bastante tristes."*
*"Os amigos aceitaram a decisão do João. **Porém**, ficaram bastante tristes."*

- conditional links:

*"**Se** o João não estivesse com dificuldades financeiras, não teria emigrado."*
*"**Estando** financeiramente estável, o João não teria emigrado."*
*"O João não teria emigrado, **caso** estivesse financeiramente estável."*

Following all these mentioned topics, we obtain a grammatically correct, coherent and structurally cohesive text.

2.5 RELATED WORK

Throughout the bibliographic research, we found some systems that also focus on the area of natural language generation:

- *NaturalOWL* [Ion Androutsopoulos et al., 2013] - produces fluent and coherent multi-sentences texts describing individuals or classes of OWL ontologies;
- *RDF2PT* [Diego Moussallem et al., 2018] - verbalizes RDF data to Brazilian Portuguese language;
- *SayAnything* [Reid Swanson and Andrew S. Gordon, 2012] - interactively writes coherent and entertaining textual narratives that mines personal stories from web-blogs;
- *MAKEBELIEVE* [Hugo Liu and Push Singh, 2002] - interactive agent that generates short fictional texts when the user supplies the first sentence of the story;
- *Scheherazade* [Li Boyang et al., 2013] - creates a fictional narrative about a simple user-provided topic;
- a Cognitive Interaction Technology – Center of Excellence (CITEC) system [Philipp Cimiano et al., 2013] - converts RDF data into natural language text based on an ontology and an associated ontology lexicon

Only one of these listed systems generates texts in Portuguese - the *RDF2PT* system - while all the others are focused on the English or Greek language (in the case of *NaturalOWL* system), which does not satisfy the purpose of our project. This *RDF2PT* system is capable of generating texts written in Portuguese but, since it is developed by a Brazilian team, these are written, more specifically, in Brazilian Portuguese. The grammatical and lexical difference between the Portuguese language of Portugal and Brazil is not as evident in written texts as it is in oral speech, but there are terms and expressions that are not used simultaneously by both countries. Being said, to the best of our knowledge, our system is the only one capable of generating complete Portuguese Portuguese texts.

The Emigration Tales system, as already mentioned in the Introduction [Section 1], has a very specific prior domain which focuses on migrants and the emigration process. This is an advantage due to the possibility, when generating the life stories of individuals, to detail certain aspects more deeply and rigorously, deepening the level of detail to the texts produced. All other systems address more generic and broad domains.

Also related to the previous paragraph, in which we affirm that the domain of the project is based on emigration, all the generated text will have the format of a story, as we find in several books that we read in our daily life, characterized by a clearly narrative tone. This results from the final purpose outlined in the Introduction to the document [Section 1], where we claim that it is possible to automatically generate, in natural language, a life story of a certain person (or group of people). As the other systems are more comprehensive and open, the final texts generated will be way more rigid and less flexible, without recreating this narrative and careful tone.

Some of these natural language generators, also due to not having a specified previous domain, require the users to enter their own information domains, for example, an ontology already populated with ready-to-use data. Without this, the system will not be able to generate output texts, since it has no information to base on. Since the large part of the target audience of the system does not have any knowledge in programming, let alone in ontologies, it makes the use of these software very difficult, or even impossible. In the case of our system, the process of data extraction and development of the ontology is already done from scratch, and the end-user will not be asked to enter his/her own domain.

During the research, we tried to test these existing systems. However, we only managed to install some of them and, from the systems that were possible to test, we could notice that the instructions for their installation / utilization are not very clear, need external tools to be executed that the vast majority of common end-users do not have and never heard of, and/or their interface is not very intuitive and little visually improved. Therefore, it is very important to be clear and consistent in the instructions for installation and use, so that the end-user feels comfortable with the handling of the system. This is an important feature we take into account in the development of our Emigration Tales project.

3

PROPOSED SYSTEM ARCHITECTURE

In the initial phase of the project, a possible design was developed for the architecture of the final system, which is presented just below. In this architecture scheme [Figure 9], we can identify seven main interconnected components: the initial database, a parser, an ontology, the system backend, a set of previously defined phrases, the application frontend and the end-user. We will go on to detail each of these components: what's their role, possible programming languages to use, and their inputs and outputs.

All these components belonging to the Emigration Tales system must be used together, inserted in a functional set, in order to take full advantage of their maximum capacities. This is due to the fact that all components are completely interconnected, and, through these connections, there is transfers of formatted and schematized data particularly for their reading and analysis. The only Emigration Tales' component that is suited to used individually and, if needed, easily transferred to any other external system is the ontology: this is the ontological database that contains information on all emigrants, their made transits and their birth locations, which are used by the other system components, illustrated in the following scheme [Figure 9].

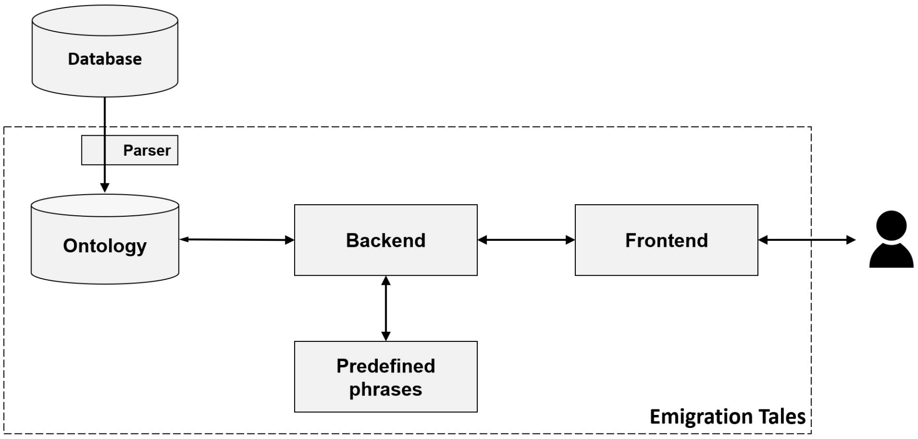


Figure 9: System architecture

3.1 DATABASE

The database of this project, as already mentioned in the introduction to the document, is based on personal information about emigrants, such as name, date of birth, origin, destination, etc. Depending on the source of this data, the available amount (and, sometimes, quality) of information about the individuals can vary significantly. Also depending on the source, the database can be written in several different ways: XML, CSV, JSON, among many other ways to store data. The greater the level of abundance and detail of the data files provided, the richer and deeper the ontology will be, which will result in a better final output text. This data can be arranged in different formats, depending on each source that shares it: plain text, listings, tables, graphics, and therefore it is necessary to create a custom processor that analyzes these sources of information and transforms the data into an homogeneous set.

3.2 PARSER

The parser of the Emigration Tales system is the processor that reads the information present in the data source and transforms it into a new scheme, analyzing and filtering the information as it is read. For each source of information that is considered, it is necessary to create, from scratch, a new customized parser, or adjust an existing one if the data structure is similar, in order to adjust to the information schema of the source. In the end, all the data from the different sources will be combined in a single, homogeneous structure of information, ready to be used later.

3.3 ONTOLOGY

The system ontology is where the data extracted from the initial database will be stored. For this, it is necessary to develop a parser that transforms the entire data file into an ontological basis, relating each personal information to the correct individual. As already mentioned in the topic of Ontologies [section 2.1] earlier on this report, this will be developed using the Turtle syntax, due to the fact that it's a simple and intuitive way of describing/representing an ontology, and also due to previous knowledge acquired during the university program. The ontology, already populated with the necessary information, is then loaded into a database that supports RDF formats and allows the execution of queries for data retrieving, as is the case with the GraphDB tool.

3.4 BACKEND

As we can analyze by the scheme above [Figure 9], the system's Backend interconnects directly with all the remaining components of the system, functioning as a data bridge between them. It is the "brain" of the entire NLG process, illustrated in the Figure 10, and is responsible for carrying out several tasks, including:

- analysis of the input introduced by the user through the Frontend, which will be covered later on the Frontend topic;
- generation of SPARQL queries to obtain certain ontology information (an example of one of these queries is shown in the Ontology to Text section [Section 2.2]);
- parsing the ontology output, in response to the previously sent query;
- verification of biological gender, where the system checks whether the individual in question is male or female: this is a very important calculation in the generation of natural language, since most of the words that make up the final text, such as determiners, pronouns and adjectives, depend on the biological gender of the emigrant.
- plot development, where the most appropriate expression/phrase in the context of the story will be chosen, taking into account all the information available about that emigrant. In this way, one of the most important properties of a grammatically correct text is maintained: its coherence, from the beginning to the end of the story.

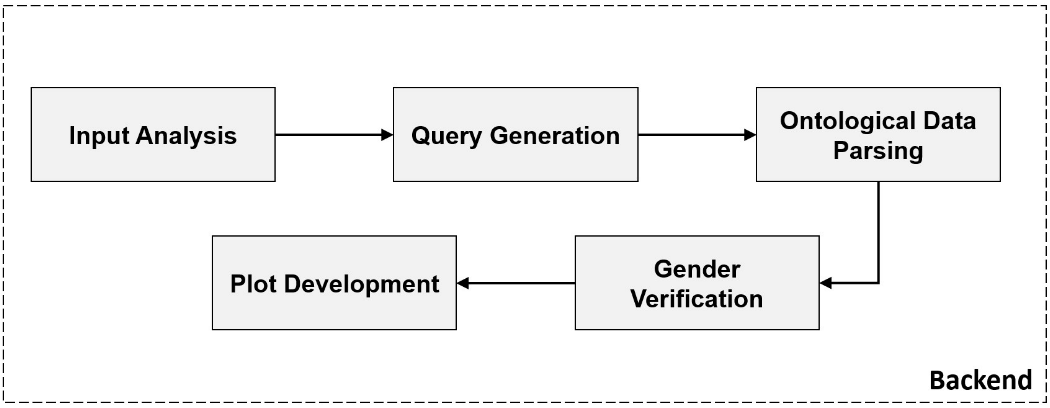


Figure 10: Backend processes

The Backend component can be developed in either Python or JavaScript. Both of these programming languages are powerful tools for building websites, web applications or software programs, and have a huge support community all over the world.

3.5 PREDEFINED PHRASES

In this component, all the preconceived phrases are kept and will give shape to the person's story. Here, there's also the process of filling these sentences with the data provided by the Backend component: since each sentence contains blank spaces intended for the name, date of birth, place of birth, nationality, etc., these are filled in according to the data of the individual to which the sentence relates. Python and JavaScript are some of the possible backend languages that could be used for its development.

3.6 FRONTEND

The Frontend is the graphic component that connects the user to the Backend. This is where the end-user enters his inputs - when asked, through clicks on the interface or entering text, for example - and where the final story is presented back. The Frontend is directly connected to the system's Backend, where it is sent, filtered and analyzed after a user action, such as clocking on a button or a link to another page. This data, as already informed, is transferred to the Backend and, after all the processes involved are completed, a response is sent back to the Frontend, which is analyzed and properly schematized on the page where the is at the moment, making it available for consulting, be it information about a person, a multi-entry table, or the story of a specific emigrant.

To develop the Frontend, the main goal is to design a graphical interface that is simple and interactive for the clients, improving their user experience to the maximum. As it is a system focused on the consulting of emigrants' personal data, mostly from the 20th century, it is expected that the users of the platform are already at an adult age, so all visual components must be discreet and minimalist, using a small palette of colors that should not distort the reading of the textual content. There is also no need for elaborate visual animations as, once again, the main focus of the platform is the quick and effective availability of historical data.

There are some possible Frontend languages for its development, such as HTML or CSS for example, or frameworks already well established in the market, such as Vue, React, Angular, among others.

3.7 END-USER

The end user is the human client of this entire system, who the story is generated for. He/she is also responsible for the introduction of inputs in the Frontend component such as, for example, the choice of the person the story will be shaped around, when the system is initialized.

4

DEVELOPMENT

In this section, all the logic and development work carried out in the various constituent parts of the system will be addressed, from the initially built prototype, ontology schema, backend and frontend of the application, among others. Throughout the sections, all the technologies used to complete the different requirements will also be indicated.

4.1 EARLY PROTOTYPE

During the initial phase of the project, a prototype was developed, using the Python programming language, in order to have a very rudimentary and preliminary idea of the final system, and to identify some early problems and challenges going forward.

This initial prototype was made up of a simple architecture with 3 main elements: a database in XML, with small pieces of information about a few people for testing, partially extracted from the CEPESE online database [1]; a Python script with some similar, but more simple, processes that will be present in the Backend of the final system; and a small set of pre-determined phrase/expression templates ready to be filled in with details about the individual (or individuals). Surely, a template will only be selected to incorporate the text if all of its blank spaces can be filled with information about the individual.

Considering the following example, there are two pre-determined phrases, whose function is to start the introduction to the main character of the story, and only one of them must be chosen to integrate the final text.

*"O seu nome era **João Ferreira**, **português**, nasceu em **Vila do Conde** a **27 de Julho de 1925**, e trabalhava como ________."*

*"**João Ferreira** foi um emigrante **português** nascido a **27 de Julho de 1925**, em **Vila do Conde**."*

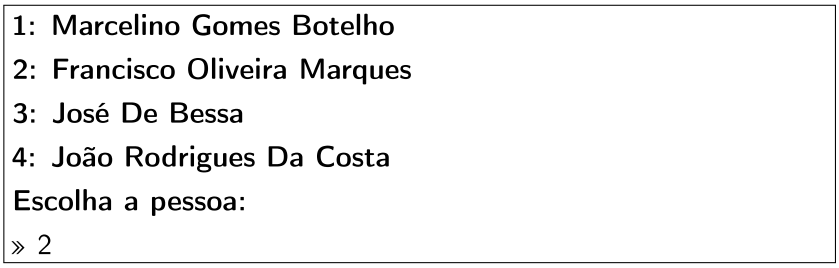
[1] CEPESE: A Emigração de Portugal para o Brasil: http://www.remessas.cepese.pt/remessas/mod/itsdatabase/view.php?n=1&v=

As can be seen, only one of the sentences can be entirely completed with data about that person, since there's no information about his job/occupation. Thereby, the sentence on the right becomes the most suitable choice for this individual, and is the one that will be used. This calculated selection is directly related to the number of pre-defined expressions/phrases, and the number of attributes that define an individual and that are made available by the data source: the more options there are to choose from, the more calculations will have to be performed to determine, in fact, the best sentence to compose the final text, and bring more variety to the stories.

Right here, there arises the challenge of the grammatical gender, which has already been explained in detail in the Grammatical gender section [Section 2.2.1]: if the expression/phrase refers to an individual of a certain biological gender, then all textual components must be grammatically adjusted in order to make it properly coherent. Since the data source does not inform the biological gender of each person listed, this binary attribute needs to be calculated automatically by the system: whether it is male or female.

For this, and after a brief search, two extensive lists of Portuguese names for male and female were found, with thousands of entries. These lists correspond to the names registered in 2013, in Portugal, by the IRN (Instituto dos Registos e Notariado). Both of these lists are stored in separate files, so it is easier to distinguish them later on. This subject will be relevant again later in this document.

With this being said, the first user step for this prototype is to select the person that he/she wants to generate a story about, from the given list, as shown in the Figure 11. Every person has its own information stored in the XML file built earlier, generating stories that will be different from each other.



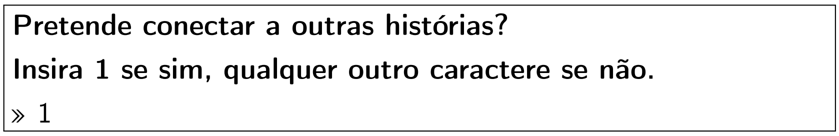
```
1: Marcelino Gomes Botelho
2: Francisco Oliveira Marques
3: José De Bessa
4: João Rodrigues Da Costa
Escolha a pessoa:
» 2
```

Figure 11: Prototype first user step

Clearly, the character inserted by the user must be valid: if the user inserts, for example, the number 0 (zero) or 5 (five), or even a letter, etc., an error message will appear, informing that the input is not valid and it was not possible to infer the chosen individual.

Here, when the user inserts a valid character, the system recognizes the main individual for whom the story will be built around and, in a simple scenery like this prototype, it would be possible to generate a life story. However, in order to explore even deeper the system as a whole, some additional settings were added, so that the generated story can be a little more customizable, and that will be used as test cases for the main system.

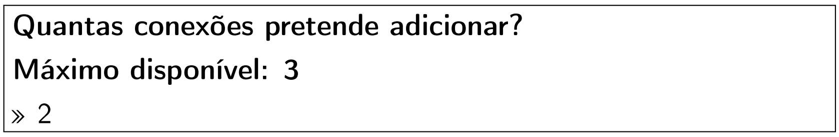
The first setting is tied up with the possibility of connecting the main individual's story to other people in the initial database, through a common point between them. Because the user may not want to connect the main individual's story to other people, it's important to ask this question before the generation of the text, like shown in the Figure 12.



```
Pretende conectar a outras histórias?
Insira 1 se sim, qualquer outro caractere se não.
» 1
```

Figure 12: Prototype second user step

If the user intends to add more people to the story, another question will be prompted, where it is asked about the number of new connections, as shown in the Figure 13.



```
Quantas conexões pretende adicionar?
Máximo disponível: 3
» 2
```

Figure 13: Prototype third user step

Naturally, if the user specified before that he/she did not want to add any connection to the story, this question will be skipped.

As we can see above, the number of available new connections is 3. This means that, even if it is possible to find a large amount of common points between the individuals and tie all their stories together, we decided to set a limit of connections, for the following reasons:

- limited number of individuals (and information about them) in the prototype database,
- little variation of pre-defined phrases in the prototype, which can lead to many repetitions of expressions/phrases throughout the text, making it look more robotic and less human,
- an unnecessary high complexion of the story.

After completing all this process, the story is generated and shown to the user. In order to try again, the user must exit the application and run it one more time.

4.2 ONTOLOGY

For the organization and relationship of data, the approach taken was to build an RDF ontology around the emigrants and the information available about them, as well as relevant data about their travels in various temporal spaces. As already mentioned in the Ontologies topic of the State of the Art [Section 2.1], the Turtle syntax of the RDF Schema was chosen for the development of the ontology, due to its easy interpretation and customization, and also due to previous knowledge about this language.

Regarding the created ontology, the Protégé tool was used to design its scheme, and is composed of 3 main classes: Person, Place and Transit, that will be detailed next, and are illustrated in the Figure 14 below. Each class comprises a series of properties that add relevant information about each individual record in the ontology.

The Person class represents a singular emigrant and contains all the biological and social information associated with that person. The properties of this class are:

- Age - age of the person at the time of registration;
- Birth date - date of birth;
- Comments - additional comments about the person;
- Country - country of birth;
- Destination occupation - person's job/occupation in the destination;
- Father - name of the father;
- Gender - biological gender, male or female;
- ID - unique text associated with that person, composed by the name of the person and a random integer;
- Literacy - literary ability, literate or illiterate;
- Marital status - civil status;
- Mother - name of the mother;
- Name - full or abbreviated name of the person;
- Occupation - main job/occupation;
- Passport number - number of the passport used for the transits;
- Source - source of the person's data.

The Place class represents the birthplace of the emigrants present in the ontology. The existence of this class will be important, later, to be able to more easily aggregate all the People who were born in a given Place. The properties of this class are:

- County - name of the county;
- District - name of the district;
- ID - unique text associated with that place, composed by the concatenation of the parish, county and district names;
- Parish - name of the parish.

For context, the Portuguese territory can be administratively divided into three tiers: districts, municipalities (or municipalities), and parishes. Each district is sub-divided into several counties, while these are also divided into parishes, with rare exceptions.

The last class present in the ontology is the Transit, which represents a travel made by one or more emigrants. This class has the following properties:

- Arrival date - date the transit arrived at the destination;
- Departure - place of origin;
- Departure date - date the transit departed from the origin;
- Destination - place of destination;
- ID - unique text associated with that transit, composed by the concatenation of departure, destination and both the dates of departure and arrival;
- Transport - transportation used.

With the three fundamental classes already created and analyzed, it is necessary to relate them, in order to combine information and create a knowledge network. Therefore, three types of relationships between classes were created:

- Person **born in** Place - relates the emigrant with his own place of birth
- Person **traveled** Transit - relates the emigrant with a transit he took
- Transit **carried** Person - inverse relation to "traveled", it relates a transit with the emigrants it transported.

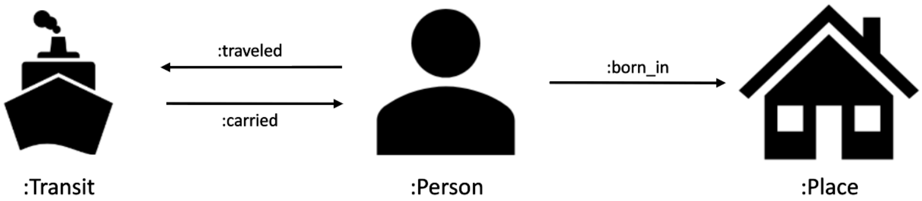


Figure 14: Ontology relations

The choice of these classes, properties and relationships was not by chance: beforehand, the available data sources, that will be talked about in the next session, were analyzed and, from there, the necessary information was collected for the creation of the ontology structure.

However, other properties were added to some classes that are not found in any of the sources used, such as the "destination occupation", "passport number", "gender" (which is a key attribute in story generation, and which will be covered in more detail in the topic of Story Generation [Section 4.5]), "passport number" attributes for the Person class; and "transport" for the Transit class. From the data sources used, it is not possible to populate any of these class properties, however they were added to the ontology structure, as they will, although, be used later.

Naturally, during the development of the project, certain properties of the ontology were added, modified and removed in order to reach a functional and stable structure, having, in the end, reached this final result that was presented previously.

4.3 DATA SOURCES AND EXTRACTION

For data population, two different sources are used, with varied information about emigrants and their travels: CEPESE's online database [2] and Fafe's emigration records [3]. While CEPESE's database mostly contains generic information about emigrants who traveled, mainly, to Brazilian cities (Rio de Janeiro, São Paulo, Manaus, Rio Grande do Sul, among others), Fafe's registries contain more detailed and in-depth data about their emigrants who, for the most part, were natives of the municipality of Fafe.

With regard to the time periods covered by these two sources, the CEPESE database does not provide any temporal context for their data; on the other hand, a large part of Fafe's records have the date of departure associated with the migrant's transit, so the time span of this data source is within the years of 1834 to 1915.

[2] CEPESE: A Emigração de Portugal para o Brasil: http://www.remessas.cepese.pt/remessas/mod/itsdatabase/view.php?n=1&v=

[3] Museu das Migrações e das Comunidades - CM Fafe: http://www.cm-fafe.pt/conteudo?item=31299

In the Table 2 below, we can see which ontology properties can be populated, with real data, using the CEPESE and Fafe records.

Property	**CEPESE**	**Fafe**
Age	X	X
Birth date		
Comments		X
Country		
Destination occupation		
Father	X	X
Gender		
Literacy		X
Marital status	X	X
Mother	X	X
Name	X	X
Occupation	X	X
Passport number		
County	X	X
District		
Parish	X	X
Arrival date		
Departure		
Departure date		X
Destination	X	X
Transport		

Table 2: CEPESE and Fafe data properties

As can be seen from the Table 2 above, all properties covered by the CEPESE database are also present in Fafe's records. However, the opposite is not true. It is also verified what was said at the end of Section 4.2: in which some properties were created which, even though they are not covered by either of these two databases, can still be populated later.

That said, and having found the common points between them, it was possible to merge these two distinct information sources, and thus build the database that was used to populate the ontology, with all this data passing through a series of processors described in Figure 15.

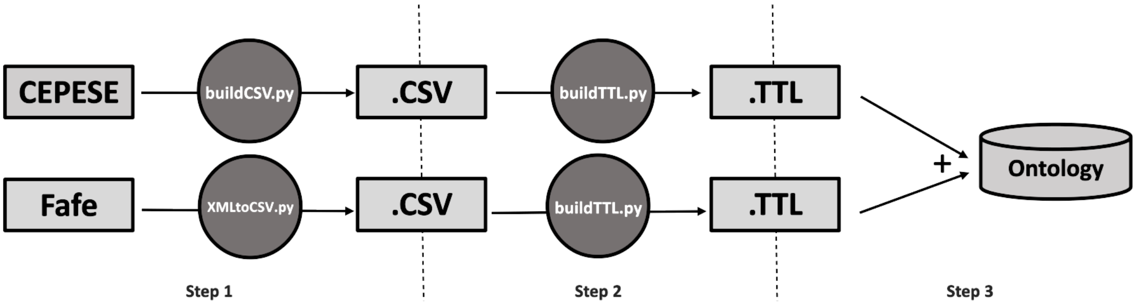


Figure 15: The steps of data extracting

4.3.1 *Step 1 - Extracting the data*

Regarding the data on emigrants available on the CEPESE website, a small web-scrapper in Python (*buildCSV.py*) was developed. A web-scrapper, in short, is a web data collector that allows an automated extraction of information available on online sites, converting it into structured information for later use. In the case of this web-scrapper, it was used to extract information from emigrants, which, through a brief visit to the CEPESE website, we can verify that this data is in the form of a table, spread over multiple pages.

Thereby, as mentioned, a web-scrapper was designed in Python that runs through all the pages available on the CEPESE website, converting this data into a CSV file.

To calculate the emigrant's ID, to be used later in the ontology, the person's name (without spaces) was concatenated with the number present in the "ID" column of the CEPESE database. Taking the example of the first emigrant present in the table, his full name is "Marcelino Gomes Botelho" and his number in the "ID" column is 2, therefore his ID in the system ontology will be "MarcelinoGomesBotelho2".

Adding this integer to the end of the emigrant's name prevents there being two different people with the same names, and therefore with the same ID in the ontology, which could cause some problems at a later stage. Thus, an integer was added to the end of the emigrant's name, avoiding this problem: even if two or more emigrants have exactly the same name, the number associated with each one will be different, making this ID unique for each entry.

Now, regarding Fafe's records, they were made available through an XML file, with all the information already structured, so no scrapper extraction was necessary. However, in order to standardize all the information available, a second Python script (*XMLtoCSV.py*) was built, capable of converting an XML file into a CSV file.

Unlike CEPESE records, emigrants from this data source do not have a unique number associated with them. Thus, for each emigrant registered in Fafe's data source, it was generated a random number between 1 and 4999 and concatenated to the person's name.

4.3.2 *Step 2 - Convert to Turtle*

So far, we have 2 CSV files (one with data from CEPESE, the other with data from the Fafe records) but, in order to populate the already created ontology, it is necessary to convert these files once again into ontological structures, in this case, into the Turtle syntax.

For this task, and for the third time, a new Python script (*buildTTL.py*) was created in order to transform the data structured in CSV into an ontological format in Turtle and that fits into the previously created ontology. This script is responsible for reading the CSV file line by line - each line represents an emigrant - and filling in the blanks with the information coming from that file. In the Figure 16, it is possible to analyze an example of an emigrant insertion into Turltle format, where the words marked by '<>' are replaced by that person's real information.

```
:<NAME+ID> rdf:type :Person,
              :name "<NAME>",
              :age "<AGE>",
              :father "<FATHER>",
              :mother "<MOTHER>",
              :occupation "<OCCUPATION>",
              (...)
```

Figure 16: RDF template for the insertion of a Person

The reading of the CSV file is performed twice for each execution:

1. the first iteration is responsible for only reading the data referring to the Person class (that is, the emigrant's personal data) and inserting that emigrant into the ontology;
2. in the second reading, the data referring to the Place and Transit classes is read (that is, the information about the person's birthplace and the trip taken) and also inserted into the knowledge base, alongside its relations with the corresponding emigrant.

This is also where the information is filtered before being inserted into the ontology: for example, in the CEPESE repository, when there is no information about a certain attribute of an emigrant, this information is filled in with "N/a", "&bnsp;", "?", "Si" or "Sem Indicação" ("No Indication" in English), among others. Naturally, it doesn't make sense to load these indications into the ontology, so they are filtered and removed.

4.3.3 *Step 3 - Concatenate the data*

Once this whole process is completed, the 3 Turtle files:

- the ontology structure,
- CEPESE data,
- and Fafe records

are concatenated together in a single .TTL file, and loaded into the GraphDB tool. This concatenation was made manually, but can be automated quickly with another Python script.

4.3.4 *Duplicate entries*

Since the beginning of data extraction, extra care has always been taken in order to not duplicate information. As the ontology comprises three main classes (Person, Place and Transit), it is very important that each entity is unique, in order to be able to correctly relate to the other entities of the ontology. Therefore, it is necessary to create a unique ID for each entry in the ontology, each containing pertinent information and that is able to clearly identify the correct individual record.

Thus, as already mentioned in the ontology structure chapter [Section 4.2], the IDs of the entities belonging to those three classes are calculated as follows:

Person - name of emigrant, no spaces + random integer
Example: JoãoFerreira294

Place - parish + county + district, except empty values
Example: JunqueiraViladoCondePorto

Transit - place of departure + place of arrival + departure date + arrival date, no spaces or '/' and except empty values
Example: PortoRiodeJaneiro2021050720210515

Knowing this, as the CSV file is read by the program, all IDs are saved as the lines are traversed. In this way, taking note in memory of all IDs, the duplication of Places and Transits is avoided:

- if the ID does not yet exist, the information about this Place/Transit and its relationship with the corresponding emigrant are inserted in the ontology, as shown in the Figure 17.

```
:<PARISH + COUNTY + DISTRICT> rdf:type :Place,
              :parish "<PARISH>",
              :county "<COUNTY>",
              :district "<DISTRICT>";
:<PERSON ID> :born_in :<PARISH + COUNTY + DISTRICT>;
```

Figure 17: RDF template for the insertion of a Place and its relation with an emigrant

- if the ID already exists, it means that its information has already been inserted in the file (there is no point in inserting it again), and only the relationship with the respective emigrant is inserted in the ontology, like in the Figure 18.

```
:<PERSON ID> :born_in :<PARISH + COUNTY + DISTRICT>;
```

Figure 18: RDF template for the insertion of a Place - Person relation

An example of a modification that was carried out during the development of the system was, namely, in the ID property of the Transit class. As mentioned above, this ID must represent a unique value for a specific transit, carried out by one or more emigrants. At first, the ID of a Transit was only defined by the value of "place of arrival + departure date" (for example "RiodeJaneiro20210515"). However, it was determined that these two values alone are not enough to characterize a single transit: two transits may have the same destination and the same departure date, but may have departed from different places and/or arrived at different times. This makes them not the same transit at all. Therefore, as mentioned above, the attributes "place of departure" and "arrival date" were added to the ID, in order to effectively define a single transit.

Nonetheless, the problem of duplicate emigrants may still exist with this approach, as, up to this point, there is still no logical check to detect duplicate migrants.

4.3.5 *Emigrants with multiple transits*

During the development of the system, it was noticed that there were several entries of the same emigrant in the ontology. This is due to the fact that, both in CEPESE's database and in Fafe's records, each entry does not correspond to a single emigrant, but rather to a transit made by an emigrant. Therefore, it was necessary to make some adjustments to the Python script that converts CSV data into Turtle format (*buildTTL.py*). Thus, it was necessary to find characteristics intrinsic to each emigrant that would define them as a unique person, filtering out duplicate entries.

The attributes belonging to the Person entity of the ontology that can define a unique person are: full name, father's name and mother's name. If these three pieces of information are exactly the same, there is a significant possibility that it is the same emigrant. Since, with the information available, it is not possible to determine 100% whether there are repeated emigrants in the data source, the choice of these three attributes was the most viable alternative to apply this filter. Once again, it should be noted that these three attributes (full name, father's name and mother's name) need to be exactly identical between two or more records for the unification of emigrants to occur, given the delicacy of the process, since it occurs automatically during the extraction of data which, to date, assembles more than 8000 entries.

At the time of the first reading of the CSV (*buildTTL.py*), when emigrants are being entered into the ontology, their names are also inserted into a memory data structure, along with the names of their father and mother. This allows to control which unique emigrants have or have not already been included into the ontology.

If, at the time of entering an emigrant, a similar emigrant already exists in this data structure (with the same name and parents' names), then this new record will be ignored, as it is assumed here that it is the same person.

Before the duplicate emigrants filter, all entries present in the data sources created a new emigrant in the ontology, as shown in the Table 3.

Name	Father	Mother	Birthplace	Transit
João Ferreira	António	Luísa	:Vila do Conde	:Belo Horizonte
João Ferreira	António	Luísa	:Vila do Conde	:São Paulo
João Ferreira		Luísa	:Vila do Conde	:Lisboa
Maria Vieira	Paulo	Fátima	:Viana do Castelo	:Rio de Janeiro
Maria Vieira	Paulo	Fátima	:Viana do Castelo	:Sevilha

Table 3: Emigrant records before the filter

After the duplicate emigrants filter, emigrants with the same name and the same parents are merged, which avoids the creation of duplicate entries. As we can see, the third entry of the emigrant João Ferreira was not merged with the other emigrants with the same name, since the father's name is not in agreement with the other two, and a new record was created.

Name	Father	Mother	Birthplace	Transit
João Ferreira	António	Luísa	:Vila do Conde	:Belo Horizonte :São Paulo
João Ferreira		Luísa	:Vila do Conde	:Lisboa
Maria Vieira	Paulo	Fátima	:Viana do Castelo	:Rio de Janeiro :Sevilha

Table 4: Emigrant records after the filter

Taking into account the available information on these emigrants, it is not possible to develop a fully competent filter: there may be several entries of emigrants with the same name and who, at first glance, appear to be the same person, but for some incongruence between the data, it is not possible to filter them out (for example, the father's names may be the same, but the mother's names may be empty or spelled differently from entry to entry), as can be seen in the Table 4.

4.3.6 *Other considerations*

As stated already, the same emigrant may appear multiple times in the same data source. Nonetheless, it should be noted that, whenever a duplicate emigrant appears, it is necessary to add the transit that he took, but his place of birth should not be added again. Thus, it was also necessary to create a logic to only associate the first place of birth that appears associated with this migrant; all followed birthplaces will be ignored and will not be inserted into the ontology.

In the CEPESE database, there are multiple references to the destination "Rio", an abbreviation of the city "Rio de Janeiro" in Brazil, which has always attracted a lot of Portuguese emigrants. In order to group the "Rio" and "Rio de Janeiro" entries into the same ontology entity - as they are in fact the same location - a filter was created that detects the "Rio" entry and modifies it to "Rio de Janeiro".

4.4 EMIGRANT GENDER CALCULATION

As already mentioned in the Grammatical Gender chapter [Section 2.2.1], the Portuguese language uses a series of determinants and pronouns that depend on the biological gender of the referred subject. For example, in the sentence "O João e a Maria emigraram para França" (in English, "João and Maria emigrated to France"), we can observe the existence of the determinants 'o' and 'a' before the names of individuals. In English, this differentiation doesn't exist in grammar (which does not prevent from also having to calculate the biological gender of the subjects).

This grammatical gender is not only related to determinants and pronouns: as mentioned in the referred Section 2.2.1, some verbs and adjectives also depend on the gender of the individual being referred to.

Therefore, in order to build a coherent story around one or more emigrants, it is almost mandatory to calculate their biological gender and then shape the entire sentence around that information. Bearing in mind that none of the data sources used so far (CEPESE and Fafe) present information about the biological gender of their emigrants, it is necessary to find an alternative way to make this important calculation.

Of all the information available about the emigrants from the two sources utilized, the only piece of data that is useful for calculating the biological gender is the person's first name. Baring any rare exceptions, people's first names are divided into two well-defined groups : male names and female names, usually given by the parents at the child's birth. In Portugal, generally speaking, with a person's first name it is already possible to determine their gender and, therefore, this was the single route that was taken.

To perform this calculation at the system's Backend level, two listings of names were found - one with male names [4], the other with female names [5] - referring to all names registered in Portugal during the year of 2013, with data from the Instituto dos Registros e Notariado. The list of male names has 1800 entries, and the list of female names contains almost 2100 entries, which makes a total of approximately 3900 different Portuguese names.

These are available online in PDF format, so it was necessary to convert them manually into two basic TXT files, which offer faster computation reading. Some less pertinent

[4] Nomes registados em 2013 - Masculinos, Instituto dos Registos e Notariado: https://www.dn.pt/DNMultimedia/DOCS+PDFS/2013/Nomes%202013%20M%20(at%C3%A9%2020%20dez.).pdf

[5] Nomes registados em 2013 - Femininos, Instituto dos Registos e Notariado: https://www.dn.pt/DNMultimedia/DOCS+PDFS/2013/Nomes%202013%20F%20(at%C3%A9%2020%20dez.).pdf

information for the final result was also filtered out: were removed the titles and page numbering, as well as all information relating to the number of registered names by entry.

Finally, it is obtained two distinct TXT files (*malenames.txt* and *femalenames.txt*) with thousands of Portuguese names, separated by the biological gender to which they are normally associated with, like shown in the Table 5.

malenames.txt	**femalenames.txt**
João	Maria
Rodrigo	Matilde
Martim	Leonor
Francisco	Mariana
Santiago	Carolina
Tomás	Beatriz
(etc...)	(etc...)

Table 5: Excerpts from *malenames.txt* and *femalenames.txt* files used for gender calculation

Here, it is now made possible to calculate with precision the biological gender of the emigrant, during the moment of the generation of his/her life story at the Backend level.

4.5 STORY GENERATION

The generation of a story for a specific emigrant can be divided into 4 distinct steps:

1. get the emigrant data;
2. calculate the age group;
3. calculate the gender;
4. build the story.

4.5.1 *Get the emigrant data*

In the first place, it is retrieved all the information about the emigrant the story is shaped around, including his personal data, place of birth and transit (or transits) that he may have taken. This is done through two SPARQL queries to the previously populated database: one to search the emigrant's personal data and place of birth, and the second to search the list of transits he has taken.

4.5.2 *Calculate the age group*

With the data on the emigrant in possession, the next step is to calculate the generation to which the emigrant belongs: if he is still a child, a young adult, adult or elderly person. This age group calculation is not at all a priority when compared to the other three steps, as it is only performed to increase the level of detail of the generated story: if this calculation is not performed, the generation of the story still is perfectly possible. In the Figure 19, it can be seen the logic used to calculate the generation of the migrant.

```
if(age < 8)
     generation = 'kid'
else if(age >= 8 AND age < 26)
     generation = 'young'
else if(age >= 26 AND age < 60)
     generation = 'adult'
else generation = 'old'
```

Figure 19: Generation calculation logic

Regarding the age limits for each designation, the following values were defined, as it is schematized in the Figure 20:

- child - under 8 years old;
- young adult - from 9 to 25 years old;
- adult - from 26 to 59 years old;
- elderly - over 60 years old.

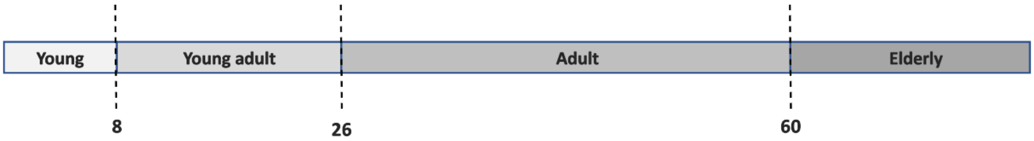


Figure 20: Generations and age limits used

4.5.3 *Calculate the gender*

After calculating the individual's age group, it is then necessary to determine the biological gender of the emigrant, taking into account his or her first name, as already explained in Section 4.4. The logic of the gender calculation used is presented in the Figure 21, and will be explained below.

Before verifying whether the emigrant's first name is present in any of the male or female name listings, a quick check is made on the last letter present in that name. In the Portuguese language, there is a percentage of proper names that end with the letter 'o' (for example "António", "Francisco", "Rodrigo") or with the letter 'a' (for example "Carolina", "Lara" , "Mariana"). As a general rule, with exceptions that are extremely rare, most of them very uncommon in Portugal, names ending in the letter 'o' are associated with men, while names ending in 'a' are associated with women. That being said, the last letter of the emigrant's first name is checked, and, if it is an 'o' or an 'a', it is then determined that his/her biological gender is male or female, respectively. If the emigrant's name ends with any other letter, his/her gender remains undefined. Noteworthy, this value is, for now, provisional, given that another additional verification will be carried out through the listings of male and female names.

The search is then performed in the lists of male and female names, even if the gender of the emigrant has already been provisionally calculated. If the emigrant's first name is found in the male names file, then it is defined as a male emigrant; if it cannot be found, the name is searched again in the female names file. If found here, then it is defined as a female emigrant.

This second calculation of the gender takes priority over the first: if the name of the emigrant is found in one of the two available files, any result of the calculation performed previously ('M', 'F' or undefined) will be replaced by the search result. If the name cannot be found in either of the two files, the result of the first calculation prevails. If, in both calculations, it is not possible to find the biological gender of the given emigrant, then this value will remain undefined.

```
const female_names = fs.readFile('femalenames.txt')
const male_names = fs.readFile('malenames.txt')
if(first_name.last_letter == 'o')
     gender = 'M'
else if(first_name.last_letter == 'a')
     gender = 'F'
if(male_names.includes(first_name))
     gender = 'M'
else if(female_names.includes(first_name))
     gender = 'F'
```

Figure 21: Gender calculation logic

4.5.4 *Build the story*

At this point, it is already possible to build a story given the information collected so far: all the emigrant's personal information, transits taken, age group to which he belongs, and biological gender.

The generation of the story then begins with the introduction of the main emigrant. Depending on the age group to which belongs and his/her biological gender, which were calculated beforehand, the emigrant is presented in a different way: he can be presented as a boy/girl, a young person or a man/woman. Initially, it is presented the most personal information of the emigrant: full name, father and mother's names and where is from. Then, it moves on to secondary information about the emigrant: profession/occupation, marital status, and literacy (i.e., if has any literary skills). The final part touches the person's emigration process: whether kept the same job or if changed his/her occupation, passport number and details about the transits made. Objectively, the more information about the individual is available, the more complete the final text will become.

This generation of natural language is carried out in the same way as in the Prototype developed in a previous phase [Section 4.1]: through the use of predetermined phases, containing blank spaces, which can be filled in with real information about the emigrant. Thus, a large set of pre-defined phrases was built, which is used to describe each single information available about the emigrant.

On a more technical level, in the instant before the construction of the story, one of two possible paths is taken, chosen at random: the first path generates the introduction of the emigrant with a certain structure, and the second path generates another introduction but with a different structure from the first. These structures share the same predefined phrases pool with each other, only changing the order in which the information about the emigrant is presented in the story.

The entire code of story generation is made up of a large number of if-else statements, which filter the predefined sentences in order to choose the one that best suits the context. In addition to numerous if-else statements, the generator also presents several random choices of numbers, whose responsibility is to pick a predefined phrase randomly from the set and, thus, make each text as unique as possible.

To report the parents of an emigrant in several different ways, predefined phrases were coded, such as:

- "sendo os seus pais o sr. ________ e a dona ________"
- "filho/filha do sr. ________ e da dona ________"
- "o nome do seu pai era ________, e o nome da sua mãe era ________"
- "o nome do seu pai era ________, enquanto que a sua mãe se chamava ________"
- "sendo o seu pai o sr. ________"
- "o nome do seu pai era ________"
- "sendo a sua mãe a dona ________"
- "o nome da sua mãe era ________"

To report the occupation of an emigrant in several different ways, predefined phrases were coded, such as:

- "trabalhava como ________"
- "trabalhava como ________ na sua terra"
- "a sua ocupação principal era ________"
- "________ era a sua ocupação principal"
- "exercia a profissão de ________"
- "ocupava os seus dias a exercer a profissão de ________"

In each of these phrases/expressions, the blank space is replaced by the real information about the emigrant's profession. Only one of these phrases is chosen to integrate the final text, chosen at random. This process is repeated for each information available about that emigrant.

One way to increase the complexity of a computer generated text, in order to make it as natural as possible, imitating the writing of a human being, is to combine different data about the emigrant within the same sentence. For this, more pre-defined phrases were coded, with a greater degree of complexity, that combines two or more information about the emigrant, such as:

- "o seu nome era ________, sendo os seus pais o sr. ________ e a dona ________"
- "antes de emigrar com ____ anos de idade, era natural do concelho de ________"
- "o seu estado civil era ________ e trabalhava como ________ na sua terra"
- "aos ____ anos de idade, a sua ocupação principal era ________"

Naturally, for choosing one of these complex sentences, none of the information that completes the blank spaces can be null.

Another component of natural language generation that was added to this system was textual props, that is, small expressions, concatenated to the pre-defined phrases already shown above, which add fictional information to the life story, also increasing the complexity of the generated text. These textual props are added to the beginning or the end of sentences, in order to increase the degree of immersiveness for the reader. It should be noted, once again, that these props may contain fictitious information about the emigrant, as they do not make use of attributes present in the ontology. They have also been carefully created and revised so that they do not contain negative information that could be harmful to the emigrant involved in the story.

Examples of these textual props are:

- "bastante conhecidos na sua terra por serem tão amigáveis e sempre dispostos a ajudar", when talking about the parents of the emigrant;
- "local onde sempre viveu antes de emigrar", when talking about the emigrant's place of birth;
- "profissão que sempre desempenhou com o maior empenho e dedicação", when talking about the emigrant's occupation
- "sendo que não era capaz de ler nem escrever", when talking about the emigrant's literacy.

As this is a natural language text generation, the story should be as cohesive as possible, but also coherent, having to make sense when read by the user. Thus, some constraints in the generation of the text begin to appear here. Some examples of conditions used in this generator are presented:

1. if there is no information about an emigrant's attribute, nothing should be wrote about it: the Emigrant class in the ontology is composed of a series of attributes, each associated with a set of pre-defined sentences that can be completed with the value of that attribute. Naturally, if there is no data about this attribute of the emigrant, none of these phrases should be chosen to be part of the text, as it would leave a blank space in the final text.
2. if the emigrant's father and/or mother's name is not known, nothing should be written about this unknown person or people: there are several textual props that refer to the family of the emigrant which the story unfolds around. These props are usually related to the relationship that the emigrant had with his/her parents, if they worked together in the same business, etc. Bearing this in mind, these textual props should not mention people who are not identified in the ontology: if, for example, the name of

the emigrant's father is unknown, there will be no textual reference to that person, and this reference may be replaced by the name of the mother, if it is known and makes sense in the context of the story.

3. use of the term "ainda" (in English, "still"), in the sentence "antes de emigrar, ainda com ____ anos de idade": the term "ainda" can be used as something that happened before its time, that is unusual for it to happen so soon. Accordingly, this term "ainda" is only used when the emigrant in the story is still a child, since it is not very common for a youngster to be an emigrant at such a young age, commonly because the parents decided to carry out this emigration. As stated earlier in this section, it was defined that the "young" generation comprised ages 0 to 8 years old, and therefore the term "ainda" is added to this expression whenever this condition is met.

4. if the emigrant is under a certain age, his/her marital status will not be mentioned: marital status is one of the attributes of the Emigrant class in the ontology but, like any other attribute (except for the migrant's name), it can be found empty. If the value of the marital status is not empty, a verification on the emigrant's age is carried out first. A limit of 16 years old is defined so that, if the emigrant is younger than this age, the marital status is omitted. This is due to the fact that, up to this age, cases where young people have a marital status other than "single" are extremely rare, taking into account the demography of migrants present in the database. If there was no such limit, the text would contain a sentence such as "o emigrante era solteiro" (in English, "the emigrant was single") even if he/she was still a child, this being redundant information and could very well be omitted from the life story.

4.5.5 *Story generation with multiple emigrants*

In order to create an interaction between the emigrants present in the ontology, the possibility of creating a story with several people was developed, as was also briefly mentioned in the Section 4.1. The maximum number of migrants in the same generated story is three. To join the story, the additional emigrants must necessarily have a common point with the main emigrant, previously chosen by the user. This connection between emigrants can be made through:

- parents: all emigrants present in story must have the same father and mother, that is, be siblings. Neither father's name nor mother's name can be unknown.
- place of birth (parish): all emigrants present in story must have been born in the same parish.
- place of birth (county): all emigrants present in story must have been born in the same county.

- transit: all emigrants present in story must have traveled on precisely the same transit.
- destination: the last destination of all emigrants present in story must be the same, regardless of whether they traveled on the same transit.

During the story generations, depending on the connection criteria chosen by the user, a query is made to the database that returns all emigrants - including the main emigrant in the story - that satisfy the defined criteria. Then, depending also on the number of connections desired, that same number of emigrants are chosen at random. Here, it is necessary to be extra careful not to choose 1. the main emigrant again, given that he/she is already in the story, and 2. repeated emigrants. This, whenever an emigrant is chosen to integrate the new story, his/her name is saved in memory. When choosing another migrant, the saved names are scrolled through to verify that the new emigrant is not already in the story.

In these stories generated with multiple emigrants, they take the following base structure:

1. presentation of the main emigrant;
2. what is the connection between the main emigrant and the other emigrants;
3. presentation of secondary emigrants;
4. small plot involving all emigrants in the story.

4.5.6 *Examples*

The following are real examples of life stories generated by the system:

ID: JosédaRocha1665
Name: José da Rocha
Age: 26
Parish: Passos
County: Fafe
Marital status: Solteiro
Transits: Departure date: 1884/02/28, Destination: Rio de Janeiro
Source: Fafe
Generated story: "**José da Rocha**, um senhor de **26** anos de idade, foi um emigrante oriundo da freguesia de **Passos**, **Fafe**. **José** viajou para **Rio de Janeiro** no dia **28 de Fevereiro de 1884**."

—

ID: AntónioMartinsGomes503

Name: António Martins Gomes
Father: Custódio Pereira Gomes
Mother: Albina Gomes Martins
Parish: Arouca
County: Arouca
Source: CEPESE

Generated story: "O nome do senhor era **António Martins Gomes**, filho do sr. **Custódio Pereira Gomes** e da dona **Albina Gomes Martins**. **António** era natural da freguesia de **Arouca**, concelho de **Arouca**."

ID: ManuelDomingueirosCabre2919
Name: Manuel Domingueiros Cabre
Age: 30
Parish: Monte
County: Fafe
Marital status: Casado
Transits: Departure date: 1872/07/31, Destination: Rio de Janeiro
Source: Fafe

Generated story: "O nome do senhor era **Manuel Domingueiros Cabre**, conhecido pelo seu grupo de amigos por apenas **Cabre**, como aliás já era usual na sua família. Antes de emigrar com **30** anos, era natural da freguesia de **Monte**, **Fafe**. **Manuel** viajou para **Rio de Janeiro** no dia **31 de Julho de 1872**."

—

ID: RodrigoAlvesMarinho1710
Name: Rodrigo Alves Marinho
Age: 14
Father: José Alves Marinho
Parish: Arnozela
County: Fafe
Marital status: Solteiro
Literacy: Literado
Transits: Departure Date: 1862/03/20, Destination: Rio de Janeiro
Source: Fafe

Generated story: "O nome do jovem era **Rodrigo Alves Marinho**, sendo o seu pai o sr. **José Alves Marinho**. Antes de emigrar ainda com **14** anos de idade, Rodrigo era natural da freguesia de **Arnozela**, concelho de **Fafe**. Na viagem que realizou, **Rodrigo** viajou para **Rio de Janeiro** no dia **20 de Março de 1862**."

—

ID: MariaRosaLopes2434
Name: Maria Rosa Lopes
Age: 21
Father: José Lopes
Mother: Zabelina Rosa
Occupation: Jornaleiro
Parish: Serafão
County: Fafe
Marital status: Casada
Literacy: Literado
Transits: Departure date: 1910/10/05, Destination: Brasil
Source: Fafe

Generated story: "O nome da jovem era **Maria Rosa Lopes**. O nome do seu pai era **José Lopes**, enquanto que a sua mãe se chamava **Zabelina Rosa**, sempre prontos a ajudar a sua filha em tudo o que ela precisasse. Antes de emigrar ainda com **21** anos de idade, **Maria** era natural de **Serafão**, **Fafe**. O seu estado civil era **casada**, sendo que ocupava o seu dia a exercer a profissão de **jornaleiro**, no entanto, considerava que esse não era o seu trabalho de sonho. **Maria** viajou para **Brasil** no dia **05 de Outubro de 1910.**"

—

ID: ManuelJoaquim1076
Name: Manuel Joaquim
Age: 46
Occupation: Proprietário
Parish: Monte
County: Fafe
Marital status: Casado
Literacy: Iletrado
Transits: Departure date: 1898/11/05, Destination: Rio de Janeiro
Source: Fafe

Generated story: "**Manuel Joaquim**, um senhor de **46** anos de idade, foi um emigrante proveniente de **Monte**, concelho de **Fafe**, mas que foi a sua casa por pouco tempo, devido aos seus pais se terem mudado para outra cidade, ainda quando era muito jovem. Trabalhava como **proprietário** na sua terra, no entanto, as suas ambições apontavam para algo mais. Era **iletrado** devido à pouca motivação de ir à escola, e nenhum incentivo por parte dos pais para alterar isso. Na viagem que realizou, viajou para **Rio de Janeiro** no dia **05 de Novembro de 1898**."

—

ID: AlbertinoMachado3325
Name: Albertino Machado
Age: 25
Father: Albino Machado
Mother: Maria de Freitas
Occupation: Agricultor
Parish: Estorãos
County: Fafe
Marital status: Casado
Literacy: Iletrado
Transits: Departure date: 1913/08/02, Destination: Brasil
Source: Fafe

Generated story: "**Albertino Machado**, um jovem de **25** anos de idade, conhecido pelo seu grupo de amigos por apenas **Machado**, foi um emigrante natural de **Estorãos**, **Fafe**. O nome do seu pai era **Albino Machado**, enquanto que a sua mãe se chamava **Maria de Freitas**, que formavam uma família bastante conhecida na sua terra pela sua humildade e trabalho árduo. **Albertino** era casado, sendo que ocupava o seu dia a exercer a profissão de **agricultor**, no entanto, as suas ambições apontavam para algo mais. Era um jovem **iletrado** devido à pouca motivação de ir à escola, e nenhum incentivo por parte dos pais para alterar isso.

Albertino e **José** eram irmãos.

O seu nome era **José Machado**, sendo os seus pais o sr. **Albino Machado** e a dona **Maria de Freitas**, que sempre apoiavam o seu filho em todas as decisões que ele tomasse para a sua vida. **José** era natural da freguesia de **Medelo**, concelho de **Fafe**. Era **solteiro**, sendo que ocupava os seus tempos a exercer a profissão de **agricultor**, juntamente com a sua mãe no negócio da familia.

Albertino e **José** trabalhavam juntos como **agricultores** no concelho de **Fafe** até que, devido a alguns problemas financeiros, ambos decidiram tomar um novo rumo e tentar a sua sorte num outro país. Juntamente, resolveram então viajar para **Brasil** no dia **02 de Agosto de 1913**."

—

ID: ManuelMachado1941
Name: Manuel Machado
Age: 27
Father: José Antonio Machado
Mother: Rosa Maria de Sampaio
Occupation: Agricultor

Parish: Regadas
County: Fafe
Marital status: Casado
Literacy: Iletrado
Transits: Departure date: 1908/01/16, Destination: Rio de Janeiro
Source: Fafe

Generated story: "**Manuel Machado**, um senhor de **27** anos, foi um emigrante proveniente da freguesia de **Regadas**, concelho de **Fafe**. O nome do seu pai era **José Antonio Machado**, enquanto que a sua mãe se chamava **Rosa Maria de Sampaio**, sempre prontos a ajudar o seu filho em tudo o que ele precisasse. Trabalhava como **agricultor**, que sempre exerceu com a maior paixão. Era **iletrado** pois não teve a oportunidade de frequentar uma escola.

Manuel e **Antonio** eram irmãos.

O seu nome era **Antonio Joaquim Machado**, filho do sr. **José Antonio Machado** e da dona **Rosa Maria de Sampaio**, que formavam uma família bastante conhecida na sua terra pela sua humildade e trabalho árduo. Era natural da freguesia de **Regadas**, **Fafe**, local onde sempre viveu antes de emigrar. **Antonio** era solteiro.

Manuel e **Antonio** foram sempre irmãos muito próximos, ajudando sempre o outro quando fosse necessário. Quando a oportunidade apareceu, apesar de estarem financeiramente estáveis, ambos decidiram mudar um pouco o rumo às suas vidas e tentar construir algo novo num outro local, longe de casa e das suas zonas de conforto. Assim, **Antonio** tomou a decisão de viajar para **Rio de Janeiro** no dia **28 de Novembro de 1907**. Seguindo o concelho do seu irmão, **Manuel** decidiu também ele viajar para **Rio de Janeiro** no dia **16 de Janeiro de 1908**."

—

ID: IdalinoDomingues4379
Name: Idalino Domingues
Age: 29
Father: José Domingues
Mother: Maria Vaz Branco
Occupation: Agricultor
Parish: Monte
County: Fafe
Marital status: Casado
Literacy: Letrado
Transits: Departure date: 1924/04/01, Destination: França
Source: Fafe

Generated story: "O nome do senhor era **Idalino Domingues**, filho do sr. **José Domingues** e da dona **Maria Vaz Branco**. **Idalino** era natural de **Monte**, concelho de **Fafe**. **Idalino** ocupava o seu dia a exercer a profissão de **agricultor**, que sempre exerceu com a maior paixão.

Idalino, **José** e **Joaquim** foram juntos na mesma viagem para **França** no dia **01 de Abril de 1924**.

O nome do senhor era **José de Freitas Ferreira**, conhecido pelos seus familiares, amigos e conhecidos por apenas **'Zé'**, diminutivo que sempre o acompanhou desde que era criança. O nome do seu pai era **Antonio de Freitas Fe**, enquanto que a sua mãe se chamava **Emilia Fernandes**. **José** era natural da freguesia de **Serafão**, concelho de **Fafe**. Trabalhava como **agricultor**, no entanto, as suas ambições apontavam para algo mais.

Joaquim Vaz Branco, um senhor de **26** anos de idade, foi um emigrante oriundo de **Monte**, **Fafe**, localidade que sempre gostara bastante e visitava frequentemente com a sua familia. O nome do seu pai era **Francisco Vaz Branco**, enquanto que a sua mãe se chamava **Balbina Vaz**. **Joaquim** era **solteiro**, sendo que ocupava os seus tempos a exercer a profissão de **agricultor**, juntamente com o seu pai no negócio da familia.

No dia **01 de Abril de 1924**, dia em que se iniciava a viagem para **França**, **Idalino**, **José** e **Joaquim** conheceram-se na fila para o embarque. Fizeram a viagem juntos, na qual agradavelmente partilharam histórias das suas vidas."

ID: MadalenaFernandesdaCosta236
Name: Madalena Fernandes da Costa
Age: 24
Father: José Fernandes da Costa
Mother: Maria Rodrigues
Occupation: Doméstica
Parish: Rio de Janeiro
County: Rio de Janeiro (brasil)
Marital status: Casado
Transits: Destination: Montevideu
Source: CEPESE

Generated story: "**Madalena Fernandes da Costa**, uma jovem de **24** anos, foi uma emigrante natural de **Rio de Janeiro**, **Rio de Janeiro (brasil)**. O nome do seu pai era **José Fernandes da Costa**, enquanto que a sua mãe se chamava **Maria Rodrigues**, que sempre apoiavam a sua filha em todas as decisões que ela tomasse para a sua vida. O seu estado civil era **casado**, sendo que ocupava os seus tempos a exercer a profissão de **doméstica**, apesar de sentir que essa não era a sua ocupação preferida, sendo que procurava ativamente outro cargo, quer em jornais, quer através de pessoas conhecidas que, esperançosamente, pudessem oferecer outro trabalho.

Madalena e **Lídia** foram juntas na mesma viagem para **Montevideu**.

Lídia de Jesus Freire, uma jovem de **11** anos, foi uma emigrante proveniente de **Aveiro**, concelho de **Aveiro**. O nome do seu pai era **Agostinho Nunes Freire**, enquanto que a sua mãe se chamava **Maria de Jesus**. Lídia trabalhava como **estudante** na sua terra.

Durante a viagem que realizaram juntas, **Madalena** e **Lídia** conheceram-se e, já no destino, juntavam as suas famílias num jantar anual em que comemoravam o dia que chegaram a **Montevideu**."

4.6 EMIGRATION TALES SYSTEM

The Emigration Tales system was built to host this generation of life stories, so that it would be available for the user to explore this feature. In addition to generating stories, the system works as an interactive database, where the user can find out all about the emigrants present in the ontology, visit their individual profiles, add new entries to the knowledge base (if they have permission to do so, through Admin Mode), get to know all the birthplaces spread mostly throughout Portugal, and consult all the transits taken by one or more passengers.

Thus, the Emigration Tales system can be divided into three main components, which interconnect with each other to provide the best possible usage experience for the end user: the ontology, the backend (or data API) and the frontend.

4.6.1 *Ontology*

As already mentioned in Section 4.2, the various processors for extracting, filtering and constructing the RDF ontology were built in the Python programming language.

Regarding the creation of the ontology scheme, the Protégé tool was used. Protégé is an open source ontology editor and knowledge management system, capable of inferring information based on the analysis of the ontology, and later exporting it in various RDF syntaxes. To take advantage of the built ontology, the GraphDB tool was used, a database compatible with RDF, where a repository was created and the ontology loaded. With GraphDB, it is possible to apply SPARQL queries to the knowledge base, which allows to extract the necessary information at any time, in a fast and structured way.

4.6.2 *Backend*

For the Backend development of the Emigration Tales project, was used the JavaScript programming language, both for the data API and for the story generator.

The data API, mounted on NodeJS, is the bridge that connects the system's Frontend to the ontology data: when the application user wants to access specific information through its interface, a request is send to the Backend which, in turn, applies the desired query to the GraphDB, which contains the ontology. If the response arrives, the Backend send it back to the Frontend, which processes the data and shows it to the final user in a structured manner.

4.6.3 *Frontend*

Frontend is the visible layer for the end use, and was developed using Vue.js, a JavaScript framework that provides useful features for building custom interfaces.

4.6.4 *Additional features*

Next, some additional features of the system are presented, which, despite not directly serving the purpose of story generation, improve the overall use of the Emigration Tales application from an aesthetic and security point of view.

Admin Mode

In order to increase the security and integrity of the data in the ontology, access to functionalities that allow the insertion of new information into the knowledge base is protected. Therefore, the insertion of new emigrants and new transits to existing migrants are exclusive to users who manage to access the Admin Mode available in the application. The Admin Mode can be accessed through the Home Page of the system, where the user is redirected to a new page, where he/she will have to enter the correct password. If the user enters a wrong password, an error will appear and prompt to try again because the password is incorrect. If the user enters the correct password, the Insert Emigrant page and the button to insert a new transit in the emigrant profile will be available.

This Admin Mode was implemented by the use of Local Storage available in the internet browsers. If the user provides the correct password, a key-value pair is created in the Local Storage of the user's web browser, containing the name of the key and its value, that represents the password entered previously). Since the Insert Emigrant and Emigrant Profile pages are dependent on this password, an initial check is made whether the key-value pair matches the correct password. If so, then the user is in Admin Mode, and will be able to insert information into the ontology. This pair has no expiration date: once the user enters the correct password, whenever he/she visits the system again, even having closed the browser previously, will always have access to the Admin Mode of the application.

Emigrant profile pictures

On the emigrants' individual page, where the migrant's information is presented, including the transits taken and a short story about him/her, there is a small portrait of a person's face. This portrait is merely aesthetic in the context of the page, as it does not reflect, in any way, the actual physical image of the corresponding emigrant.

The choice of the portrait contains a previously built logic, that depends on two very important factors: the migrant's age and biological gender. As mentioned earlier in this section, these two emigrant attributes are obtained through custom calculations. It should be noted that, although a scale of age groups was already presented in Section 4.5.2, it was not put to use in order to calculate the best portrait. Here, a different scale was used that defined, in a more realistic and reliable way, the alteration of the human physiognomy.

Therefore, when the user opens the individual page of an emigrant, the age group and biological gender of that person are calculated, and then, based on these data, a portrait is chosen at random, among the possibilities that fit the mentioned requirements.

Insert new data into ontology

In the Emigration Tales system, a password-protected page is available in order to maintain the integrity and reliability of the information presented to all the users, where it is possible to add knowledge to the ontology: add a new emigrant, and their personal information, as well as, optionally, the transit carried out by this migrant. Here, the only fields that are defined to be mandatory are the name of the emigrant and the travel destination, if the user wants to associate it. This information filled in by the user is schematized in a SPARQL query, which is sent to GraphDB through a POST request.

If the user is authenticated as a system administrator, he/she can also enjoy the functionality of adding a transit to an existing emigrant, on his or hers individual page. This system capacity is similar to the insertion of a new emigrant described before. Here, the only mandatory field is the travel destination due, once again, to the ID generation of this new record. Upon successful entry of a new transit, the associated emigrant's profile page is automatically refreshed.

4.6.5 *Loading times*

In this section, some loading times of the various features of the system will be presented. These times were determined from the server where the system is currently deployed on. For each feature present in the following tables, were collected 5 samples of execution times and, in the bottom row, the averages of all these times were calculated.

First, in the Table 6, the loading times of the three tables referring to the three classes present in the ontology are presented: Emigrants, Places and Transits. Each of these tables contains all the records registered in the database. In the last column, the loading times of 5 pages of emigrant profiles are presented.

Emigrants	Places	Transits	Emigrant profile
840	55	477	1980
691	56	366	2280
684	47	312	1890
703	48	328	1990
844	57	357	1880
752,4	**52,6**	**368**	**1990**

Table 6: Loading times for record tables and emigrant profiles (in milliseconds)

As it is possible to see by comparing the averages between the first 3 columns of the Table 6, their loading times are clearly distinct. This is due to the fact that the number of emigrants present in the ontology is much greater than the number of transits which, in turn, is also greater than the number of birthplaces. Therefore, as there are more records to be loaded into each table, the longer it will take for it to be completely loaded.

In the last column of the table, as said, are the loading times for 5 different emigrant profiles, all of which are around 2 seconds of total page load.

Still in the Places and Transits table of the system, when clicking on an entry, a new table is displayed containing the emigrants related to that record: if a certain Place is chosen, all emigrants who were born there are listed; if a Transit is chosen, the emigrants who took part in that trip are presented. In the Table 7, response times for this feature are shown, where 5 Places and 5 Transits were randomly chosen, and their loading times determined.

Place	Transit
65	83
108	63
60	32
63	58
64	34
72	**54**

Table 7: Loading times for place and transit related emigrants (in milliseconds)

Finally, response times to the feature of creating stories were determined, and noted in the Table 8. In the first column of this table, 5 emigrants were chosen randomly and the response time calculated for retrieving all the information available about them. It is possible to see that, in any of these samples, none of the waiting times exceeded 100 milliseconds.

In the following 3 columns, the execution times for the generation of stories are presented, based on a main Emigrant. For this, the same main emigrant was used, to obtain more reliable data and, like this, enabling the comparison between these samples. Here, too, a different connection criteria was used between emigrants for each sample, as the Emigration Tales system offers 5 options: parents, place of birth (by parish or county), transit and destination.

Emigrant information	Story w/ 1 emig.	Story w/ 2 emig.	Story w/ 3 emig.
76	80	89	93
52	29	100	123
82	33	74	127
51	63	146	148
64	62	240	273
65	**53,4**	**129,8**	**144,8**

Table 8: Loading times for emigrant information and life stories (in milliseconds)

By analyzing the Table 8, it is possible to verify that, as expected, the greater the number of emigrants who participate in the story, the longer the text generation takes. This is due to the fact that a search is carried out in the database, that consults all emigrants who meet the requirement imposed by the connection criteria, previously defined by the end-user. This search, however, does not take place when the user only wants to generate the life story of a single emigrant only, saving some loading time.

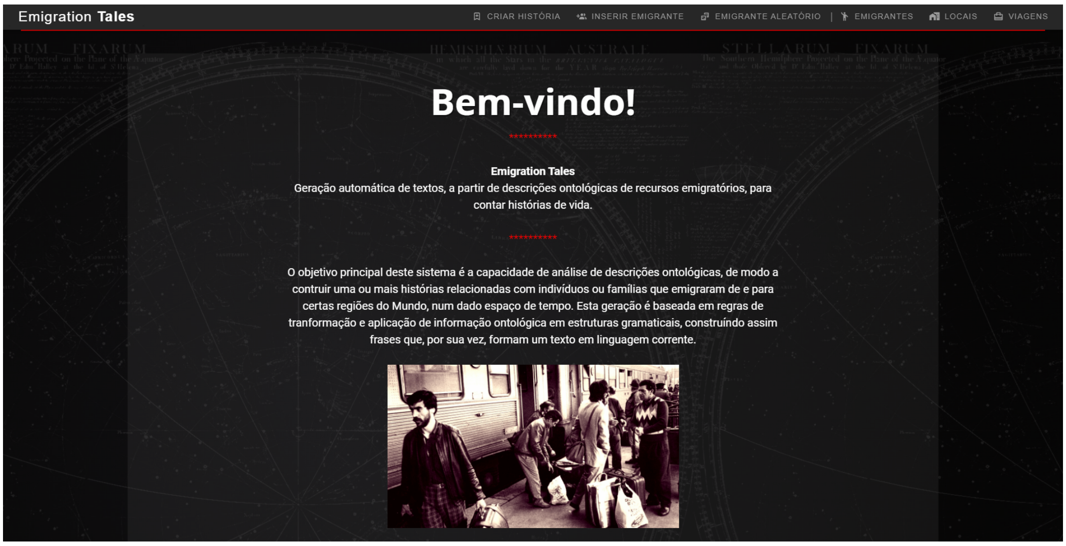
5

A LOOK OVER THE EMIGRATION TALES SYSTEM

In this chapter, the various constituent pages of the web application are presented, including a detailed description of what the main user is capable of seeing and doing in each one, as well as a print-screen of that component. In general terms, a complete "guided tour" is made to the whole system developed in the scope of the master thesis over the last months, and that can be visited at `http://epl.di.uminho.pt:50502`.

5.1 HOME-PAGE

The home page of the application is presented to the user when he enters the system. Here, are presented a brief summary about the motivation behind this project, the system capabilities, a simple scheme of the ontology used and some other useful links, like the GitHub page of the developer, the CEPESE online emigration database, and the Fafe Museum. A screenshot of the home page can be seen in the Figure 22.



Bem-vindo!

Emigration Tales
Geração automática de textos, a partir de descrições ontológicas de recursos emigratórios, para contar histórias de vida.

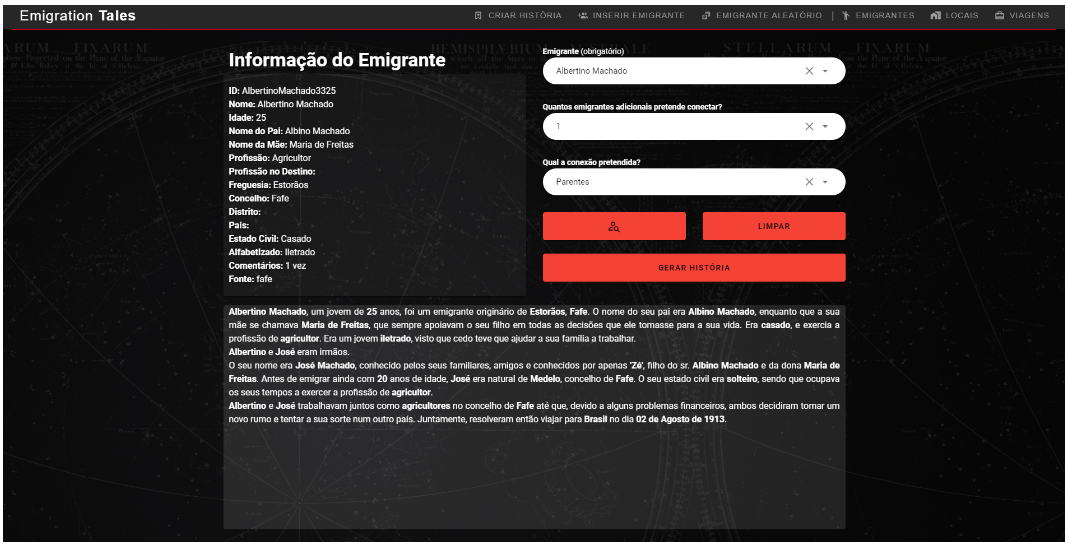
O objetivo principal deste sistema é a capacidade de análise de descrições ontológicas, de modo a contruir uma ou mais histórias relacionadas com indivíduos ou famílias que emigraram de e para certas regiões do Mundo, num dado espaço de tempo. Esta geração é baseada em regras de tranformação e aplicação de informação ontológica em estruturas gramaticais, construíndo assim frases que, por sua vez, formam um texto em linguagem corrente.

Figure 22: Home page

5.2 CREATE STORY

The Create Story page is where the user can generate a life story of an emigrant, previously defining a series of criteria: the main emigrant of the story, the number of connections intended to add (in other words, how many additional emigrants will be added to this story), and the type of connection pretended (that is, the common link between the participating emigrants). Only the choice of the main emigrant is mandatory; the number of connections and the search criteria are both optional, as the user can leave them blank. A screenshot of the Create Story page can be seen in the Figure 23.

When the user clicks on the "Gerar História" button (in English, "Generate Story"), the created life story appears in the text box below. The words in bold correspond to real information about the emigrant, provided by the ontology. The "Limpar" button (in English, "Clean") serves the purpose of clearing any content present in the bottom text box: if the user wants to delete a generated story, this button is clicked and it is erased.



Informação do Emigrante

ID: AlbertinoMachado3325
Nome: Albertino Machado
Idade: 25
Nome do Pai: Albino Machado
Nome da Mãe: Maria de Freitas
Profissão: Agricultor
Profissão no Destino:
Freguesia: Estorãos
Concelho: Fafe
Distrito:
País:
Estado Civil: Casado
Alfabetizado: Iletrado
Comentários: 1 vez
Fonte: fafe

Emigrante (obrigatório): Albertino Machado

Quantos emigrantes adicionais pretende conectar?: 1

Qual a conexão pretendida?: Parentes

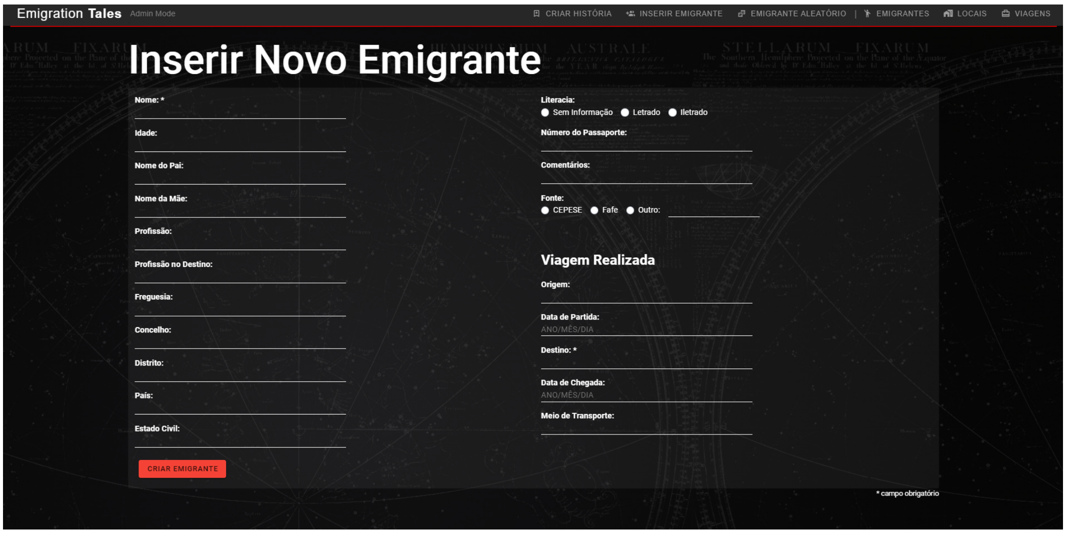
Albertino Machado, um jovem de **25** anos, foi um emigrante originário de **Estorãos, Fafe**. O nome do seu pai era **Albino Machado**, enquanto que a sua mãe se chamava **Maria de Freitas**, que sempre apoiavam o seu filho em todas as decisões que ele tomasse para a sua vida. Era **casado**, e exercia a profissão de **agricultor**. Era um jovem **iletrado**, visto que cedo teve que ajudar a sua família a trabalhar.
Albertino e **José** eram irmãos.
O seu nome era **José Machado**, conhecido pelos seus familiares, amigos e conhecidos por apenas **'Zé'**, filho do sr. **Albino Machado** e da dona **Maria de Freitas**. Antes de emigrar ainda com **20** anos de idade, **José** era natural de **Medelo**, concelho de **Fafe**. O seu estado civil era **solteiro**, sendo que ocupava os seus tempos a exercer a profissão de **agricultor**.
Albertino e **José** trabalhavam juntos como **agricultores** no concelho de **Fafe** até que, devido a alguns problemas financeiros, ambos decidiram tomar um novo rumo e tentar a sua sorte num outro país. Juntamente, resolveram então viajar para **Brasil** no dia **02 de Agosto de 1913**.

Figure 23: Create story page

On the left, there is a panel that is filled with all the information about the emigrant chosen by the user, clicking on the button with the magnifying glass. This panel is very useful to help the user to get to know better the emigrant he has previously chosen, as it contains all of his/her personal information. If the user clicks the search button without first picking an emigrant from the dropdown list, an error will be presented.

5.3 INSERT EMIGRANT

The Insert Emigrant page is blocked for general users, therefore it is only possible to access it if the system is in Admin mode, for reasons of security and data integrity. Here, it is possible to insert a new emigrant into the ontology, filling in a series of available fields. When inserting an emigrant, the user can, right from this page, associate a transit that he/she has taken. The only fields that are required and must be written are the Name of the emigrant to be inserted, and the Destination of the transit, if the user wants to associate one. These fields are mandatory due to the fact that they are essential to determine the object IDs, within the ontology. A screenshot of the Insert Emigrant page can be seen in the Figure 24.



Inserir Novo Emigrante

Nome: *

Idade:

Nome do Pai:

Nome da Mãe:

Profissão:

Profissão no Destino:

Freguesia:

Concelho:

Distrito:

País:

Estado Civil:

Literacia:
● Sem Informação ● Letrado ● Iletrado

Número do Passaporte:

Comentários:

Fonte:
● CEPESE ● Fafe ● Outro:

Viagem Realizada

Origem:

Data de Partida:
ANO/MÊS/DIA

Destino: *

Data de Chegada:
ANO/MÊS/DIA

Meio de Transporte:

* campo obrigatório

Figure 24: Insert Emigrant page

When the user fills in all the required data and clicks on the button to create the emigrant, he is directly redirected to the created migrant's profile page, in case of a successful insertion in the ontology.

5.4 RANDOM EMIGRANT

A very useful feature for users who want to browse the emigrants database, but are not quite sure which of the thousands of emigrants to choose, or just don't want to go through the work of clicking on each emigrant at a time. By clicking on "Random emigrant", this loading page is displayed, which is shown to the user while the system finds a random emigrant from the ontology. This calculation is not very extensive, therefore, under normal conditions, this page should not take more than 2 seconds to be replaced by the profile of the emigrant chosen at random. A screenshot of the loading page of a random emigrant profile can be seen in the Figure 25.

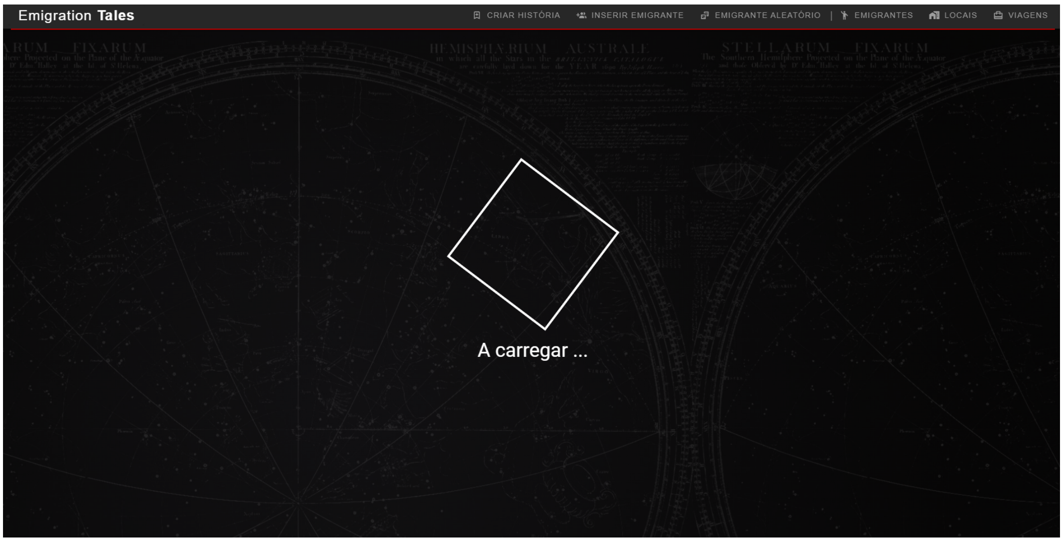
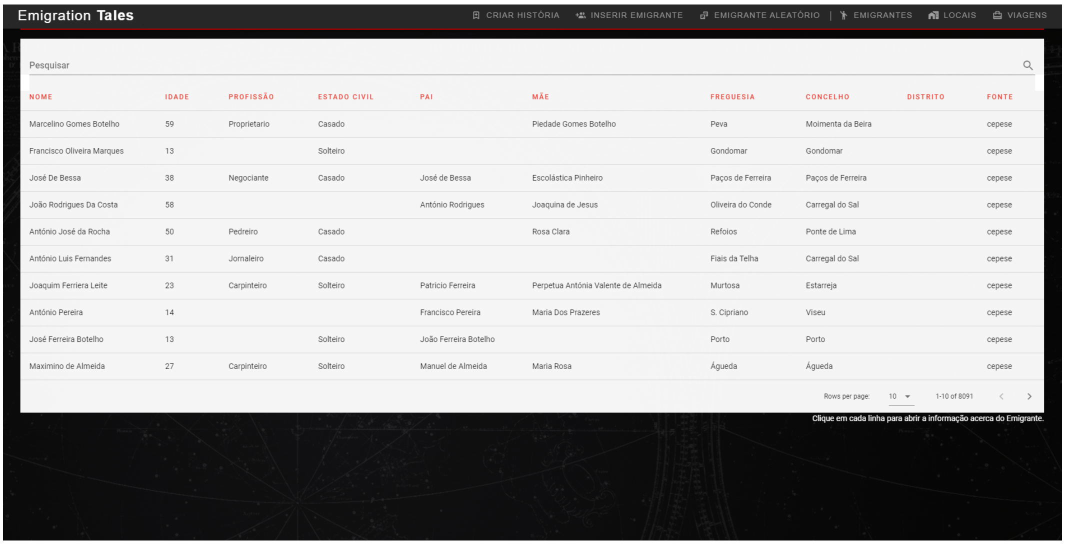


Figure 25: Page presented while the emigrant is loaded

5.5 EMIGRANTS

Here is where it can be found, organized in a table, all the emigrants registered in the ontology. From this table alone, it is possible to know some key information about each emigrant, namely his/her name, age, occupation, marital status, father and mother's names, parish, county, district of birth, and the source of this data. By clicking on each row of the table, a new tab opens in the browser where the individual profile of the emigrant chosen previously by the user is opened. By default, 10 emigrants are shown per table window, but the user has the option to display more or less records, and to navigate forward and backwards in this table as well, in order to see more entries. A screenshot of the emigrants table can be seen in the Figure 26.



NOME	IDADE	PROFISSÃO	ESTADO CIVIL	PAI	MÃE	FREGUESIA	CONCELHO	DISTRITO	FONTE
Marcelino Gomes Botelho	59	Proprietario	Casado		Piedade Gomes Botelho	Peva	Moimenta da Beira		cepese
Francisco Oliveira Marques	13		Solteiro			Gondomar	Gondomar		cepese
José De Bessa	38	Negociante	Casado	José de Bessa	Escolástica Pinheiro	Paços de Ferreira	Paços de Ferreira		cepese
João Rodrigues Da Costa	58			António Rodrigues	Joaquina de Jesus	Oliveira do Conde	Carregal do Sal		cepese
António José da Rocha	50	Pedreiro	Casado		Rosa Clara	Refoios	Ponte de Lima		cepese
António Luis Fernandes	31	Jornaleiro	Casado			Fiais da Telha	Carregal do Sal		cepese
Joaquim Ferriera Leite	23	Carpinteiro	Solteiro	Patricio Ferreira	Perpetua Antónia Valente de Almeida	Murtosa	Estarreja		cepese
António Pereira	14			Francisco Pereira	Maria Dos Prazeres	S. Cipriano	Viseu		cepese
José Ferreira Botelho	13		Solteiro	João Ferreira Botelho		Porto	Porto		cepese
Maximino de Almeida	27	Carpinteiro	Solteiro	Manuel de Almeida	Maria Rosa	Águeda	Águeda		cepese

Clique em cada linha para abrir a informação acerca do Emigrante.

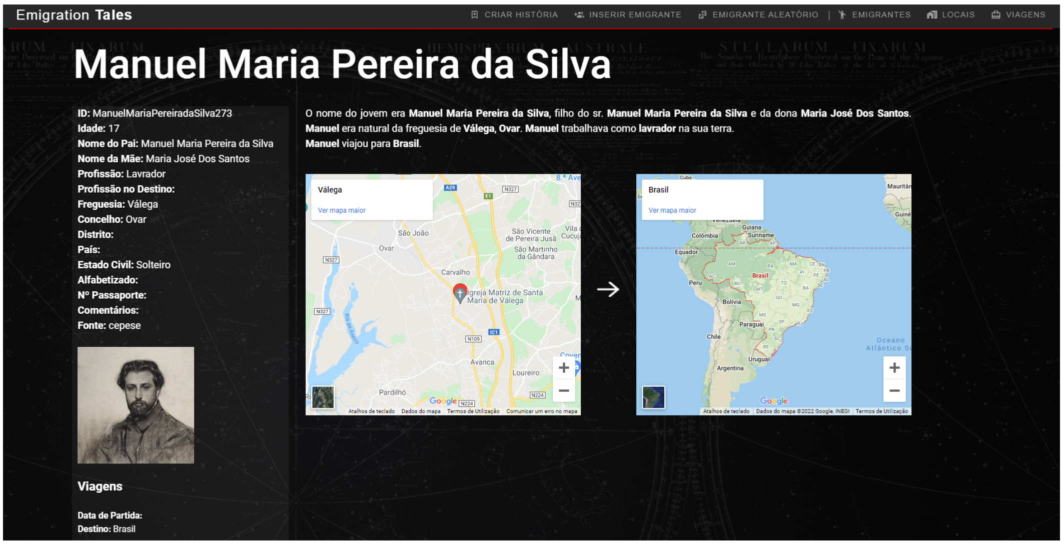
Figure 26: Emigrants page

At the top there is also a search bar. This search is not only limited to the names of the emigrants, it works for any information listed in the table below. It is possible to search, for example, for all emigrants called "Manuel", all emigrants aged 40 or all emigrants that were born in "Fafe", just by typing these key-words. This is a real-time search: whenever the users enters a character, a search is performed and the table records are updated according to what has already been written.

5.6 EMIGRANT PROFILE

To open the individual page of an emigrant, the users can access it by either clicking the Random Emigrant button on the top navigation bar, or by the listing of all the emigrants present in the ontology. In the Emigrant profile page, the full name of the emigrant is displayed at the top and, further down, in the left column, all the available information about that person. Below, we can see a picture chosen through the calculation of the age and biological gender of the emigrant, as already mentioned in the Section 4.6.4. Still, at the end of the column, it is presented the complete list of transits made the migrant, also containing all the available data about them. A screenshot of an emigrant profile can be seen in the Figure 27.

Also on this page, there is a brief story about the emigrant. Unlike the stories created on the Create Story page, this story does not contain inferences about the person: it only contains information known and made available by the data sources.



Manuel Maria Pereira da Silva

ID: ManuelMariaPereiradaSilva273
Idade: 17
Nome do Pai: Manuel Maria Pereira da Silva
Nome da Mãe: Maria José Dos Santos
Profissão: Lavrador
Profissão no Destino:
Freguesia: Válega
Concelho: Ovar
Distrito:
País:
Estado Civil: Solteiro
Alfabetizado:
Nº Passaporte:
Comentários:
Fonte: cepese

Viagens

Data de Partida:
Destino: Brasil

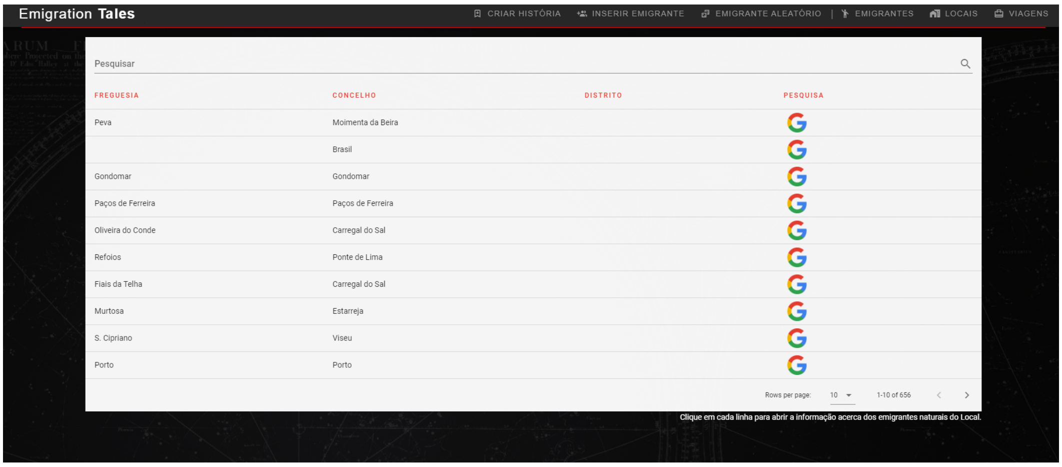
O nome do jovem era **Manuel Maria Pereira da Silva**, filho do sr. **Manuel Maria Pereira da Silva** e da dona **Maria José Dos Santos**.
Manuel era natural da freguesia de **Válega, Ovar**. **Manuel** trabalhava como **lavrador** na sua terra.
Manuel viajou para **Brasil**.

Figure 27: Emigrant profile

To complete the profile page, interactive world maps point to the places the emigrant lived or visited. In the map on the left, the parish or municipality where the emigrant was born. On the right, the destination where the person emigrated, in case he/she made only on transit, as shown in the Figure 27. If the emigrant has more records of trips taken, the layout of the maps is different: the maps are aligned vertically, where the place of birth is at the top, and the different destinations of the transits at the bottom. These interactive world maps are provided by Google Maps, where the user can move and zoom the map without having to leave the system.

5.7 PLACES

On the Places tab, there is a table that contains all the unique locations from which the emigrants in the ontology are native. This table consists of 4 columns: parish, county and district of the locality from which one or more emigrants were born, and, more to the right, a button with the Google logo, in which the user can search for this same locality in the search engine. A screenshot of the places table can be seen in the Figure 28.



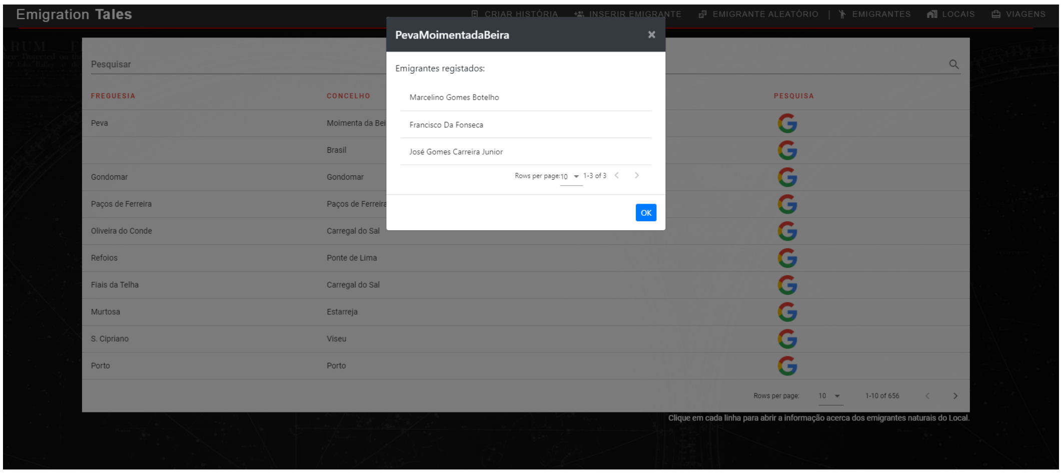
FREGUESIA	CONCELHO	DISTRITO	PESQUISA
Peva	Moimenta da Beira		G
	Brasil		G
Gondomar	Gondomar		G
Paços de Ferreira	Paços de Ferreira		G
Oliveira do Conde	Carregal do Sal		G
Refoios	Ponte de Lima		G
Fiais da Telha	Carregal do Sal		G
Murtosa	Estarreja		G
S. Cipriano	Viseu		G
Porto	Porto		G

Rows per page: 10 1-10 of 656

Clique em cada linha para abrir a informação acerca dos emigrantes naturais do Local.

Figure 28: Places page

All the lines in the table are clickable: when the user clicks on an entry, a modal appears on the screen that displays a list of all emigrants who were born in that location. This list of emigrants is also interactive, so it is made possible for the user to quickly access the profile of an emigrant listed there. A screenshot of this modal can be seen in the Figure 29.



PevaMoimentadaBeira

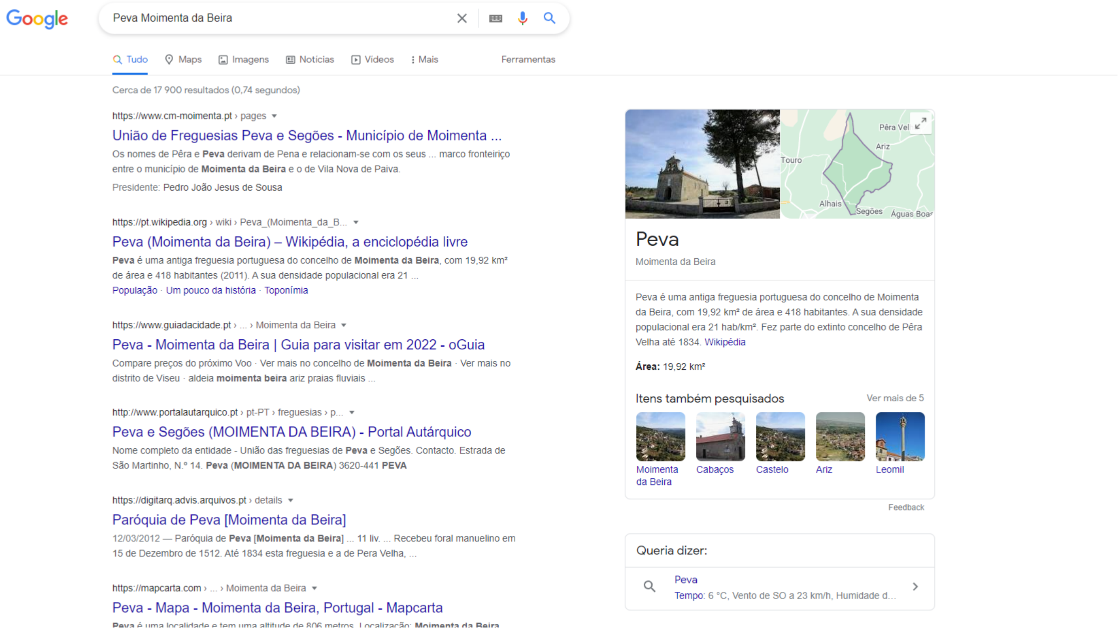
Emigrantes registados:

- Marcelino Gomes Botelho
- Francisco Da Fonseca
- José Gomes Carreira Junior

Rows per page:10 1-3 of 3

Figure 29: Place information

By clicking on the Google button associated with each entry, a new tab is opened in the user's web browser and an automatic search is carried out in this search engine, using the name of the location previously chosen. A screenshot of the Google page that is opened can be seen in the Figure 30.



Peva Moimenta da Beira

Cerca de 17 900 resultados (0,74 segundos)

https://www.cm-moimenta.pt › pages
União de Freguesias Peva e Segões - Município de Moimenta ...
Os nomes de Pêra e **Peva** derivam de Pena e relacionam-se com os seus ... marco fronteiriço entre o município de **Moimenta da Beira** e o de Vila Nova de Paiva.
Presidente: Pedro João Jesus de Sousa

https://pt.wikipedia.org › wiki › Peva_(Moimenta_da_B...
Peva (Moimenta da Beira) – Wikipédia, a enciclopédia livre
Peva é uma antiga freguesia portuguesa do concelho de **Moimenta da Beira**, com 19,92 km² de área e 418 habitantes (2011). A sua densidade populacional era 21 ...
População · Um pouco da história · Toponímia

https://www.guiadacidade.pt › ... › Moimenta da Beira
Peva - Moimenta da Beira | Guia para visitar em 2022 - oGuia
Compare preços do próximo Voo · Ver mais no concelho de **Moimenta da Beira** · Ver mais no distrito de Viseu · aldeia **moimenta beira** ariz praias fluviais ...

http://www.portalautarquico.pt › pt-PT › freguesias › p...
Peva e Segões (MOIMENTA DA BEIRA) - Portal Autárquico
Nome completo da entidade - União das freguesias de **Peva** e Segões. Contacto. Estrada de São Martinho, N.º 14. **Peva** (**MOIMENTA DA BEIRA**) 3620-441 **PEVA**

https://digitarq.advis.arquivos.pt › details
Paróquia de Peva [Moimenta da Beira]
12/03/2012 — Paróquia de **Peva** [**Moimenta da Beira**] ... 11 liv. ... Recebeu foral manuelino em 15 de Dezembro de 1512. Até 1834 esta freguesia e a de Pera Velha, ...

https://mapcarta.com › ... › Moimenta da Beira
Peva - Mapa - Moimenta da Beira, Portugal - Mapcarta
Peva é uma localidade e tem uma altitude de 806 metros. Localização: **Moimenta da Beira**

Peva
Moimenta da Beira

Peva é uma antiga freguesia portuguesa do concelho de Moimenta da Beira, com 19,92 km² de área e 418 habitantes. A sua densidade populacional era 21 hab/km². Fez parte do extinto concelho de Pêra Velha até 1834. Wikipédia

Área: 19,92 km²

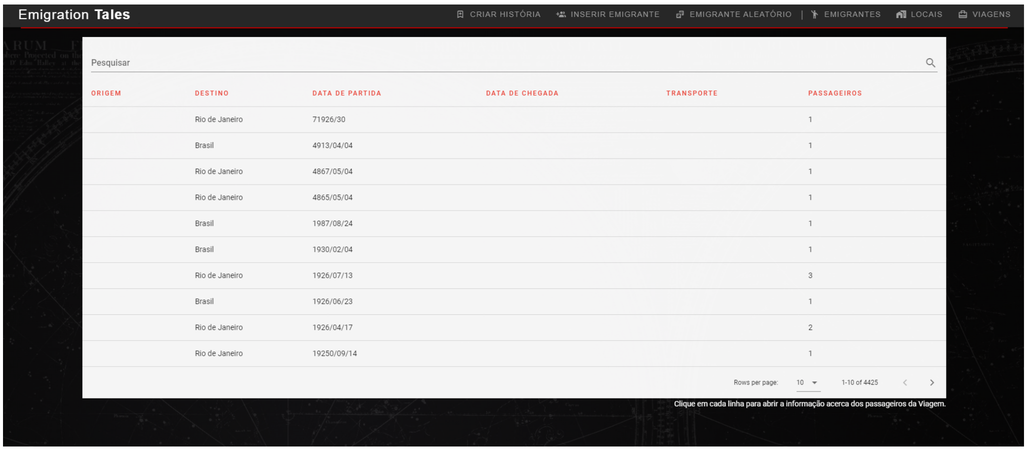
Itens também pesquisados Ver mais de 5
Moimenta da Beira · Cabaços · Castelo · Ariz · Leomil

Queria dizer:
Peva
Tempo: 6 °C, Vento de SO a 23 km/h, Humidade d...

Figure 30: Place Google page

5.8 TRANSITS

The last listing present in the application is the Transits table. Here, all the unique trips made by the emigrants in the ontology are listed. This table is made up of 6 columns: place of origin, place of destination, date of departure and arrival, mean of transport and number of emigrants who made this same trip. A screenshot of the transits table can be seen in the Figure 31.



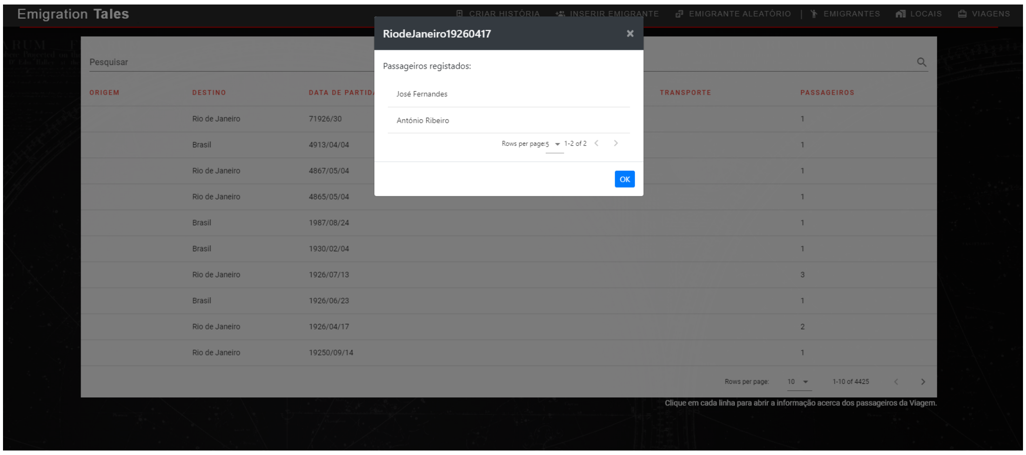
ORIGEM	DESTINO	DATA DE PARTIDA	DATA DE CHEGADA	TRANSPORTE	PASSAGEIROS
	Rio de Janeiro	71926/30			1
	Brasil	4913/04/04			1
	Rio de Janeiro	4867/05/04			1
	Rio de Janeiro	4865/05/04			1
	Brasil	1987/08/24			1
	Brasil	1930/02/04			1
	Rio de Janeiro	1926/07/13			3
	Brasil	1926/06/23			1
	Rio de Janeiro	1926/04/17			2
	Rio de Janeiro	19250/09/14			1

Rows per page: 10 1-10 of 4425

Clique em cada linha para abrir a informação acerca dos passageiros da Viagem.

Figure 31: Transits page

Similar to the other tables available and previously mentioned in the sections, this listing is also clickable: if the user selects an entry in the table, a modal will be displayed on the screen, containing the list of emigrants who participated in this transit, similar to what happens in the Places table. Also here in this modal, all emigrants are clickable, opening their individual profile in a new browser tab. A screenshot of this modal can be seen in the Figure 32.



RiodeJaneiro19260417

Passageiros registados:

José Fernandes

António Ribeiro

Rows per page:5 1-2 of 2

ORIGEM	DESTINO	DATA DE PARTID		TRANSPORTE	PASSAGEIROS
	Rio de Janeiro	71926/30			1
	Brasil	4913/04/04			1
	Rio de Janeiro	4867/05/04			1
	Rio de Janeiro	4865/05/04			1
	Brasil	1987/08/24			1
	Brasil	1930/02/04			1
	Rio de Janeiro	1926/07/13			3
	Brasil	1926/06/23			1
	Rio de Janeiro	1926/04/17			2
	Rio de Janeiro	19250/09/14			1

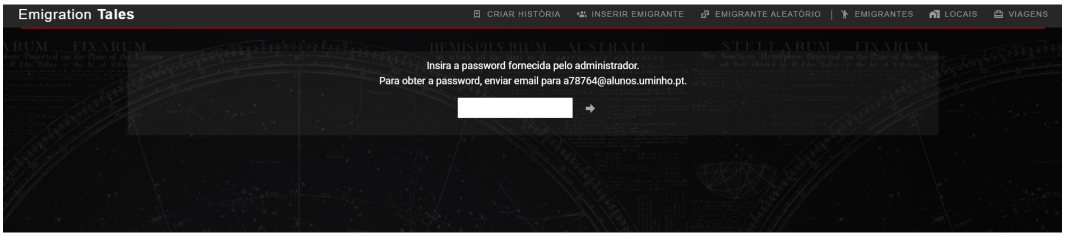
Rows per page: 10 1-10 of 4425

Clique em cada linha para abrir a informação acerca dos passageiros da Viagem.

Figure 32: Transit information

5.9 ADMIN

In order to log into the system as Admin, it is necessary to enter the correct password on this page, as already mentioned in Section 4.6.4. If a wrong password is entered, the user will be notified with an error message, informing to try again as it was entered an incorrect password. A screenshot of the admin page can be seen in the Figure 33.



Insira a password fornecida pelo administrador.
Para obter a password, enviar email para a78764@alunos.uminho.pt.

Figure 33: Admin page

5.10 REPOSITORY DATA

During this section, some interesting data about the emigrants present in the Emigration Tales' system is presented, in the form of Q&A (questions and answers). Next to each answer, there is also a graphic that visually represents the distribution of this information among the different possibilities.

How many emigrants are there in the ontology?

There are 8,091 emigrants included in the ontology. As mentioned in Section 4.3.5, a filter was applied that removed entries from repeated emigrants, through their name, father's name and mother's name; however, taking into account the available data on these emigrants, it is not possible to develop a 100% competent filter. Among all these emigrants, 961 entries (11.9%) correspond to data from CEPESE [1], while most records were taken from the Fafe database [2], accounting for the remaining 7,130 emigrants (88.1%). This information is graphically represented in the Figure 34.

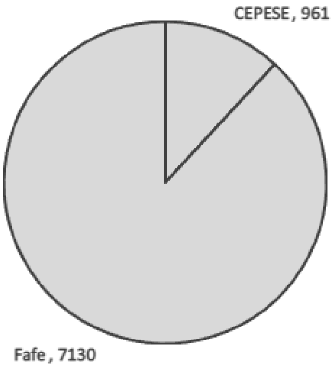


Figure 34: Emigrants data sources

What is the most common birthplace?

In the ontology, emigrants come from 655 different places. Among the localities with more emigrants born there, we find the municipality of Fafe with 6,815 emigrants (84.2%), Porto with 179 emigrants (2.2%), Guimarães with 83 emigrants (1%) and Celorico with 65 emigrants (0.8%). Other places present in the ontology are Felgueiras, Vila Nova de Gaia, Póvoa de Varzim and Viseu. This great difference from the first to the second most popular location is entirely due to the use of Fafe records for the ontology population, where the overwhelming majority of these entries are from this municipality. Going deeper into this topic, the most popular parishes in the municipality of Fafe are: Fafe with 756 emigrants, S. Gens with 553 emigrants, Moreira with 500 emigrants, Quinchães with 413 emigrants and Travassós with 321 emigrants. Among other parishes in Fafe, we have Golães, Revelhe, Queimadela, Estorãos, Monte, Varzea Cova, Fornelos and Serafão. This information is graphically represented in the Figure 35.

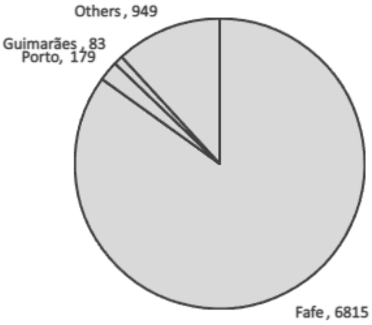


Figure 35: Common birthplaces

[1] CEPESE: A Emigração de Portugal para o Brasil: http://www.remessas.cepese.pt/remessas/mod/itsdatabase/view.php?n=1&v=

[2] Museu das Migrações e das Comunidades - CM Fafe: http://www.cm-fafe.pt/conteudo?item=31299

What is the most common destination location?

In the ontology, 4425 unique transits are registered. The most common destination for emigrants, taking into account the available data, are: Rio de Janeiro with 2536 transits (57.3%), Brazil (without any mention of the exact city) with 790 transits (17.9%), Pará with 710 transits (16.0%) and Spain with 80 transits (1.8%). The remaining 309 transits (7.0%) have destinations less popular among emigrants, including France, Manaus, Pernambuco, Santos, Porto Alegre, Argentina and Minas Gerais. This information is graphically represented in the Figure 36.

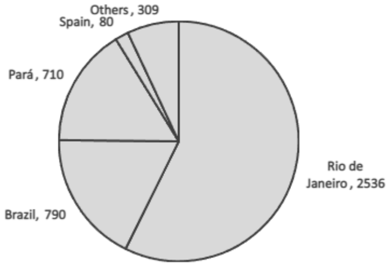


Figure 36: Most common destinations

What are the most common professions?

From a universe of more than 8,000 registrations, 3,440 emigrants have a designated profession. The most common occupation among all the emigrants in the ontology is Agricultor (in English, "Farmer") with 618 emigrants (18.0%), followed by Proprietário ("Owner") with 542 emigrants (15.8%), Jornaleiro ("Newsman") with 332 emigrants (9.7%), Trabalhador ("Worker") with 211 emigrants (6.1%), Lavrador ("Farmer") with 209 emigrants (6.1%), Capitalista ("Capitalist") with 209 emigrants (6.0%), Pedreiro ("Bricklayer") with 160 emigrants (4.7%) and Carpinteiro ("Carpenter") with 153 emigrants (4.4%). This information is graphically represented in the Figure 37.

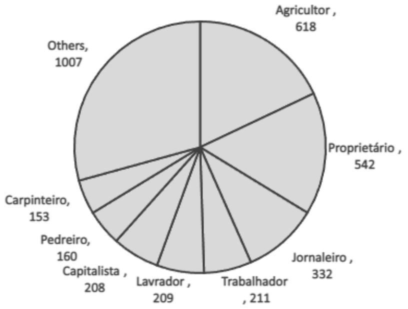


Figure 37: Most common occupations

How many emigrants have made more than one transit?

The maximum number of transits that a single emigrant made was 4, counting 6 different emigrants who made this total number of trips. Afterwards, a total of 26 emigrants made 3 different transits each. Finally, there are 150 emigrants in the ontology who embarked on 2 transits. Like this, it is possible to state that, in total, 182 single emigrants made more than one trip in their lifetime, based on the available data, corresponding to only 2.2% of the total ontology. This information is graphically represented in the Figure 38.

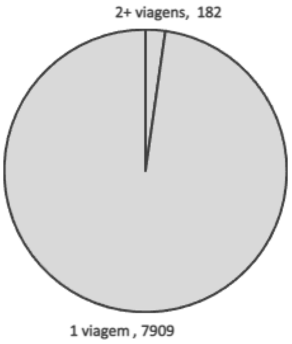


Figure 38: Distribution of number of transits per emigrant

6

CONCLUSION

Throughout this document, all the logic behind the development of the system entitled Emigration Tales was presented. This system can be visited at any time at `http://epl.di.uminho.pt:50502`.

The main objective that motivated the creation of this system was the generation of texts in natural language which describe the information about emigrants in the form of life stories. The information about emigrants is picked from an ontological knowledge base, referring to emigrants who traveled from Portugal to other parts of the globe. This text generation is based on rules of transforming and applying the information coming from the ontology into grammatical structures, thus forming complete sentences that, as a whole, form a life story with relevant information about the emigrant.

Like that, an ontology was constructed from scratch, customized to solve this project's requirements, and then populated with information from two different sources, to which the availability of this data is greatly appreciated: CEPESE [1] and Fafe [2]. This data was then extracted, processed and organized into the final ontology.

From here, with real data already in possession, a Web application was developed that would not only generate life stories for these emigrants, but also function as a complete system for research, consultation and insertion of data into the already built knowledge base. As it was possible to verify throughout this document, the Emigration Tales system offers several features alongside the generation of stories, such as the search for emigrants, insertion of new knowledge entries into the ontology, individual profiles, interactive maps, portraits calculated according to the person's age and gender, and search for birthplaces and transits made by one or more passengers.

[1] CEPESE: A Emigração de Portugal para o Brasil: http://www.remessas.cepese.pt/remessas/mod/itsdatabase/view.php?n=1&v=

[2] Museu das Migrações e das Comunidades - CM Fafe: http://www.cm-fafe.pt/conteudo?item=31299

Regarding the research hypothesis presented in the introduction to this document [Section 1.4], it was achieved by creating a natural language generator adjacent to the Emigration Tales application. This generator receives, from the ontology, information about one or more emigrants, which will be part of the final story, and, from there, infers the best pre-defined phrases to use, and then completes them with the data retrieved earlier. For this, it was defined a series of pre-defined phrases, with blank spaces strategically placed in order to be completed, at a later stage, with the emigrant's data. The choice of these sentences is not random, as it depends on certain factors such as the existence (or not) of that information to complete the blank space, and the personal and social context behind that information.

In order to add greater interaction, not only with the system user, but also between the emigrants themselves, the option of grouping a multiple number of people (maximum 3) in the same story was added. The user has the ability to choose the number of additional emigrants to add to the final story, also defining the connection point between them: emigrants will only be added to the story if they are directly related to the main emigrant thought the chosen connection point; otherwise, if there is no record of migrants who meet this requirement, only the main emigrant's life story is generated.

6.1 FUTURE WORK

About future work, the main focus is on adding more pre-defined sentences to the generator, in order to increase the textual variety of life stories. Currently, the generator already has a wide variety of phrases/expressions that can be used in the text, but adding more content is always positive when generating texts in natural language. It is also necessary to take into account that the domain of the ontology is based mostly on emigrants from the 19th and 20th centuries, so that the pre-defined phrases must stay coherent with this temporal space. Another limitation regarding the addition of new phrases is the amount of attributes that describe the ontology's Person class, which is not a very high number, but also the fact that there is a limited amount of information available from the data sources about each emigrant, limiting here the size of the life stories.

As mentioned in the initial Objectives [Section 1.2], the final result of this project could be integrated in the online pages of the museums that interested in this approach, in order to enrich their virtual visits provided to the users. Thus, contacts could be initiated regarding this proposal, with several national museums, in order to interconnect the systems.

Another, more secondary, improvement that could be made in the future is at the application interface level, where it is possible to add more animations to images and texts, more appealing colors and backgrounds, better layout of contents, among other visual uplifts.

Not being the main focus of this project, the graphical interface was designed primarily to be simple and minimally attractive to the user, so there is some room for improvement on this subject.

6.2 WORK PROGRESS

The development cycle of this project had an estimated duration of one year and was be divided in 6 phases, according to the following schedule, illustrated in the Figure 39:

1ST TO 2RD MONTH Bibliographic search about the Portuguese emigration and database search for information that can be valuable to populate the system.

3RD MONTH Research for already-to-use automatic text generators. Elaboration of the general system architecture. Planning and writing the Pre-Thesis Document.

4TH TO 5TH MONTH Design, build and revise the ontology.

6TH TO 7TH MONTH Extract, filter and store the information in the ontology. Tweak the ontology structure, if necessary.

8TH TO 10TH MONTH Plan and develop the automatic text generator of life stories.

11TH TO 12TH MONTH Run several tests on the system and write Master's Thesis document.

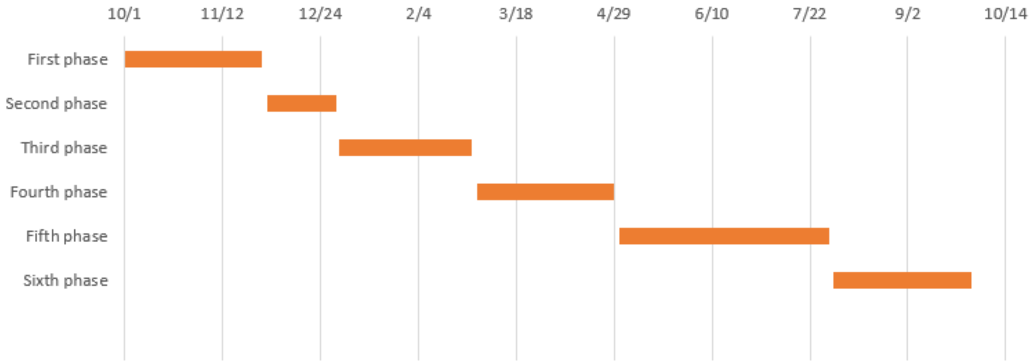


Figure 39: Gantt chart

Regarding this work plan, it can be said that, at an initial stage, it was accomplished without any considerable delays, and respecting the deadlines presented on the chart. During the following phases, was used a more flexible approach, with less rigid deadlines, but always maintaining a high work rate, and always complying with the established end dates.

BIBLIOGRAPHY

Prof. Dr. Jorge Carvalho Arroteia. Aspectos da emigração portuguesa. *Revista Electrónica de Geografía y Ciencias Sociales*, 2001.

Diego Moussallem, Thiago Castro Ferreira, Marcos Zampieri, Maria Claudia Cavalcanti, Geraldo Xexeo, Mariana Neves, and Axel-Cyrille Ngonga Ngomo. Rdf2pt: Generating brazilian portuguese texts from rdf data. 2018.

Ehud Reiter and Robert Dale. Building natural language generation systems. 2000.

Hugo Liu and Push Singh. Makebelieve: Using commonsense knowledge to generate stories. *MIT Media Laboratory*, 2002.

Ion Androutsopoulos, Gerasimos Lampouras, and Dimitrios Galanis,. Generating natural language descriptions from owl ontologies: the naturalowl system. *Journal of Artificial Intelligence Research*, 2013.

Li Boyang, Stephen Lee-Urban, George Johnston, and Mark Riedl. Story generation with crowdsourced plot graphs. 2013.

Philipp Cimiano, Janna Luker, David Nagel, and Christina Unger. Exploiting ontology lexica for generating natural language texts from rdf data. *Cognitive Interaction Technology – Center of Excellence, Bielefeld University*, 2013.

Reid Swanson and Andrew S. Gordon. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. 2012.

Seiji Isotani and Ig Ibert Bittencourt. Dados abertos conectados. 2015.

Telmo Móia. Construção de textos: Gramaticalidade, coesão e coerência. *Centro de Estudos Judiciários*, 2014.