FEDERAL UNIVERSITY OF MINAS GERAIS
BIOLOGICAL SCIENCES INSTITUTE
INTERUNIT PROGRAM OF POST-GRADUATION IN BIOINFORMATICS

Master's Dissertation

LabControl – A software for microbial information management

STUDENT: **Mariana Teixeira Dornelles Parise**
SUPERVISOR: **Prof. Dr. Vasco Ariston de Carvalho Azevedo**
CO-SUPERVISOR: **Prof. Dr. Gabriel Fernandes**

BELO HORIZONTE
November – 2016

Mariana Teixeira Dornelles Parise

LabControl – A software for bacterial information management

Dissertation presented as partial requirement for obtaining the Master's Degree by Post-Graduation Department of Bioinformatics, Biological Sciences Institute, Federal University of Minas Gerais.

Supervisor: Prof. Dr. Vasco Ariston de Carvalho Azevedo

Co-supervisor: Prof. Dr. Gabriel Fernandes

BELO HORIZONTE

November – 2016

Parise, Mariana Teixeira Dornelles
043 LabControl: a software for bacterial information management [manuscrito] / Mariana Teixeira Dornelles Parise. – 2016.

71 f. : il. ; 29,5 cm.

Orientador: Prof. Dr. Vasco Ariston de Carvalho Azevedo. Coorientador: Prof. Dr. Gabriel Fernandes.
Dissertação (mestrado) – Universidade Federal de Minas Gerais, Instituto de Ciências Biológicas.

1. Laboratório - Biologia. 2. Sistemas de informação em laboratório clínico. 3. Automação laboratorial. 4. Bioinformática - Teses. I. Azevedo, Vasco Ariston de Carvalho. II. Fernandes, Gabriel. III. Universidade Federal de Minas Gerais. Instituto de Ciências Biológicas. IV. Título.

CDU: 573:004

Universidade Federal de Minas Gerais
Instituto de Ciências Biológicas
Programa Interunidades de Pós-Graduação em Bioinformática da UFMG

ATA DA DEFESA DE DISSERTAÇÃO

27/2016
entrada
1º/2015
CPF:
026.576.900-09

Mariana Teixeira Dornelles Parise

Às dez horas do dia **17 de novembro de 2016**, reuniu-se, no Instituto de Ciências Biológicas da UFMG, a Comissão Examinadora de Dissertação, indicada pelo Colegiado do Programa, para julgar, em exame final, o trabalho intitulado: **"LabControl - A software for microbial information management"**, requisito para obtenção do grau de Mestre em **Bioinformática.** Abrindo a sessão, o Presidente da Comissão, **Dr. Vasco Ariston de Carvalho Azevedo**, após dar a conhecer aos presentes o teor das Normas Regulamentares do Trabalho Final, passou a palavra à candidata, para apresentação de seu trabalho. Seguiu-se a arguição pelos Examinadores, com a respectiva defesa da candidata. Logo após, a Comissão se reuniu, sem a presença da candidata e do público, para julgamento e expedição de resultado final. Foram atribuídas as seguintes indicações:

Prof./Pesq.	Instituição	CPF	Indicação
Dr. Vasco Ariston de Carvalho Azevedo	UFMG	283.171.225-68	APROVADA
Dr. Gabriel da Rocha Fernandes	FIOCRUZ	052.706.216-22	APROVADA
Dr. Jan Baumbach	University of Southern Denmark		APROVADO
Dr. Richard Röttger	University of Southern Denmark	CH1H1XKZZ	APROVADA
Dr. Aristóteles Góes Neto	UFMG	944.348.825-20	APROVADO
Dr. Siomar de Castro Soares	UFTM	056.951.826-11	APROVADA

Pelas indicações, a candidata foi considerada: APROVADA
O resultado final foi comunicado publicamente à candidata pelo Presidente da Comissão. Nada mais havendo a tratar, o Presidente encerrou a reunião e lavrou a presente ATA, que será assinada por todos os membros participantes da Comissão Examinadora.

Belo Horizonte, 17 de novembro de 2016.

Dr. Vasco Ariston de Carvalho Azevedo - Orientador _____

Dr. Gabriel da Rocha Fernandes - Co-Orientador _____

Dr. Jan Baumbach p/p _____

Dr. Richard Röttger _____

Dr. Aristóteles Góes Neto _____

Dr. Siomar de Castro Soares _____

“Os ideais que iluminaram o meu caminho são a Bondade, a Beleza e a Verdade.”

“The ideals that have lighted my way have been Kindness, Beauty, and Truth.”

Albert Einstein

ACKNOWLEDGEMENTS

I would like to thank, in first place, God, for the opportunity of being alive and starting my scientific life. Also, I am really thankful to Professor Vasco Azevedo for accepting me as a student and guiding me on the science road, and to Professor Gabriel Fernandes for co-orienting me. I would also like to thank Professor Aristóteles Góes-Neto; his advice was essential to the development of this work. In addition, I am really thankful to Professors Jan Baumbach and Richard Röttger for accepting being members of the board of this dissertation.

A special thanks to Joarley dos Santos, who is my programming partner in this work. He has showed me a new way to develop web systems and taught me a lot.

I am very grateful to Sheila, Natália, Stefane and Fernanda who lovely helped me with all the bureaucracy during this time.

Also, I would like to thank my family for supporting me and always believing that I can achieve any goal, especially my Mom, Dad and grandparents. To my Dad, a huge thanks for showing me this amazing field of study.

I thank my life partner and best friend Doglas for all the support and patience in this stage of my life, for all the talks about my dissertation, all the endless times he read parts of this dissertation and gave his valuable opinion. Thank you for all the support and mostly by holding my hand and making me give one more step when I thought I wouldn't be able to.

I am really thankful to all LGCM team, especially to Thiago Sousa, Marcus, Raquel, Jamal, Sandeep and Anne who filled with joy my days in the laboratory. I am as well grateful to Nilson, Cassiana, Daniela, Arun, Thiago Castro, Núbia, Marcela, Kátia, Izabella, Yasmim and Flávia Alburjaile.

Additionally, I would like to thank Diego Mariano, Edgar Lacerda and Flávia Rocha for their contributions in the early stages of this work.

A huge thanks to all the people who on and on evaluated my data model, and answered my endless questions about the techniques, microorganisms, procedures and everything else.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ATCC	American Type Culture Collection
BSM	Brazilian Society of Microbiology
BRC	Biological Resource Centers
CABRI	Common Access to Biological Resource and Information
CBAS	Collection of Bacteria from Environment and Health
CBMAI	Brazilian Collection of Environmental and Industrial Microorganisms
CCMB	Microbial Culture Collection of Bahia
CIP	Collection of Institute Pasteur
CMC-UFMG	Collection of Microorganisms and Cells of UFMG
CSS	Cascading Style Sheets
DI	Dependency Injection
DNA	Deoxyribonucleic Acid
DSMZ	German Collection of Microorganisms and Cell Cultures
DTO	Data Transfer Object
FIOCRUZ	Osvaldo Cruz Foundation
GOLD	Genomes OnLine Database
GSC	Genomic Standards Consortium
HTML	HyperText Markup Language
INPI	National Institute of Industrial Property of Brazil
IoC	Inversion of Control
JSON	JavaScript Object Notation
LIS	Laboratory Information Systems
LIMS	Laboratory Information Management Systems
LGCM	Laboratory of Cellular and Molecular Genetics
MIGS	Minimum Information about a Genome Sequence
MIxS	Minimum Information of any (x) Sequence
MIMARKS	Minimum Information about a Marker Gene Sequence
MIMS	Minimum Information about a Metagenome Sequence
MVC	Model-View-Controller
NCBI	National Center for Biotechnology Information
NGS	Next Generation Sequencing

OEDC	Organization for Economic Co-operation and Development
OMMS	Omics Metadata Management Software
pH	Hydrogen ion concentration
REST	REpresentational State Transfer
RNA	Ribonucleic Acid
RNA-seq	RNA sequencing
SBM	Brazilian Society of Microbiology
SOP	Standard Operation Procedures
UFMG	Federal University of Minas Gerais
WFCC	World Federation for Culture Collection

ABSTRACT

During the past years, next-generation sequencing techniques have allowed large-scale sequencing and generated an enormous and constantly increasing quantity of genomic data. This scenery, together with the development of cheap, faster and modern microbiological techniques, has allowed scientists to produce more data with higher quality. However, this advance has created management issues to microbiological laboratories which deal with NGS and biological collections.

Considering those management issues, the main goal of this study is to develop a Laboratory Information Management System (LIMS) to support these laboratories in the management of the generated data. LabControl was implemented as a web-based system, using the programming language Java. Its functionalities and data model were designed according to literature, data patterns and researchers' opinions.

As a result, a reliable and validated data model was specified and the functionalities of the system were designed based on that model. To address the established needs, the architecture of the system was designed and the technologies to be used were chosen. The system has been built with the usage of modern and reliable technologies, and based on good practices of scientific computing and software development.

LabControl is presented as a comprehensive open-source LIMS specialized in managing NGS and biologic collections microbiological data. Nevertheless, the system can be used by any microbiological laboratory due to its capability to handle any type of information from *in silico*, *in vivo* and *in vitro* experiments. Finally, LabControl presents an easy-to-use interface and can be easily expanded to handle non-addressed features.

STRUCTURE OF THE DISSERTATION

This dissertation is divided into six chapters. Chapter one presents an introduction to the discussed topics. The motivation and aims of this work are presented in chapters two and three, respectively. Chapter four shows the methodology used in this dissertation. The results are presented in chapter five, and the discussion is presented in chapter six. In chapter seven, the conclusion and perspectives of this work are shown.

1. INTRODUCTION

1.1. Omics Sciences

According to Weinstein (1998), omics can be comprehended as the study of entities such as DNA and RNA as a whole (WEINSTEIN, 1998). Currently, the use of high-throughput methods in molecular biology and their applications is comprehended as omics (BARH *et al.*, 2013) and it is used to study organisms in terms of their molecules roles, relationships and actions. According to Barh *et al*. (2012), the four major fields of study in this area are Genomics, Transcriptomics, Proteomics and Metabolomics (BARH *et al.*, 2012).

The field of genomics can be comprehended as the study of the genome, which is the whole genetic complement of an organism. To this aim, techniques such as DNA sequencing are employed and the generated data is analyzed to explore genomes' structure and function (MCKUSICK & RUDDLE, 1987; MADIGAN, 2015).

Transcriptomics is the study the transcriptome, which is the complete set of RNA produced in an organism, using methods such as microarray or RNA-seq to understand which transcripts are produced under a specific condition or in a specific cell. (VELCULESCU *et al*., 1997; MCGETTIGAN, 2013; MADIGAN, 2015).

Proteomics is considered the study of the proteome, which is the complete set of proteins of an organism. This aims to explore proteins' structure, function and activity in an organism (WASINGER *et al*., 1995; KELLNER, 2000; MADIGAN, 2015).

Metabolomics refers to the study of the metabolome, the complete set of metabolite intermediates and other small molecules of an organism, through their systematic identification and quantification (FIEHN, 2002; MADIGAN, 2015).

This work will concentrate on the genomic field of study.

1.1.1. Genomics

Genomic studies have allowed a better comprehension of DNA sequences since the Sanger method development (SANGER *et al.*, 1977), and this field of study was officially stated in 1987 (MCKUSICK & RUDDLE, 1987). Next Generation Sequencing (NGS) technologies have allowed large-scale sequencing and

generated an enormous and constantly increasing quantity of genomic data (FIELD *et al*., 2008; MOREIRA *et al*., 2015). These improvements of the sequencing methods have drastically increased the number of genome projects and fostered knowledge generation. Public databases such as NCBI and GOLD have grown due to this phenomenon (as it is shown in Figure 1) and it has allowed the exploration of these genomes aiming to extract new information and knowledge (GARRITY *et al*., 2015).

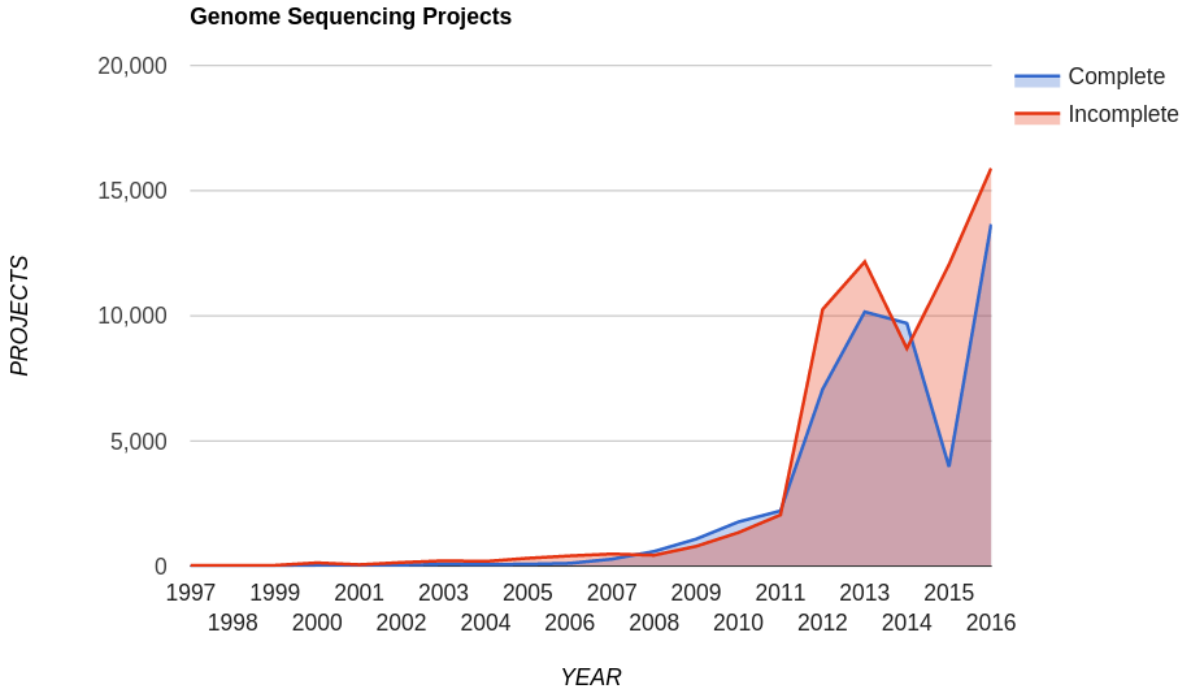

Figure 1 – Genome Projects growth from GOLD database.

Despite the possibilities brought by NGS technology, some issues appeared with this data explosion such as management and standardization problems. This huge amount of data and the information about this data (metadata) can be processed, compared, interpreted, shared and reused by researchers (PEREZ-ARRIAGA *et al.*, 2015). These tasks may be facilitated or complicated by the data completeness and standardization as well as the adopted management strategy, considering that good management strategies prevent data to be lost and provide reliable and available information.

In Bioinformatics, the most common genomic processes that generate data and metadata are sequencing, assembly and annotation; a general vision of these processes is shown in Figure 2.

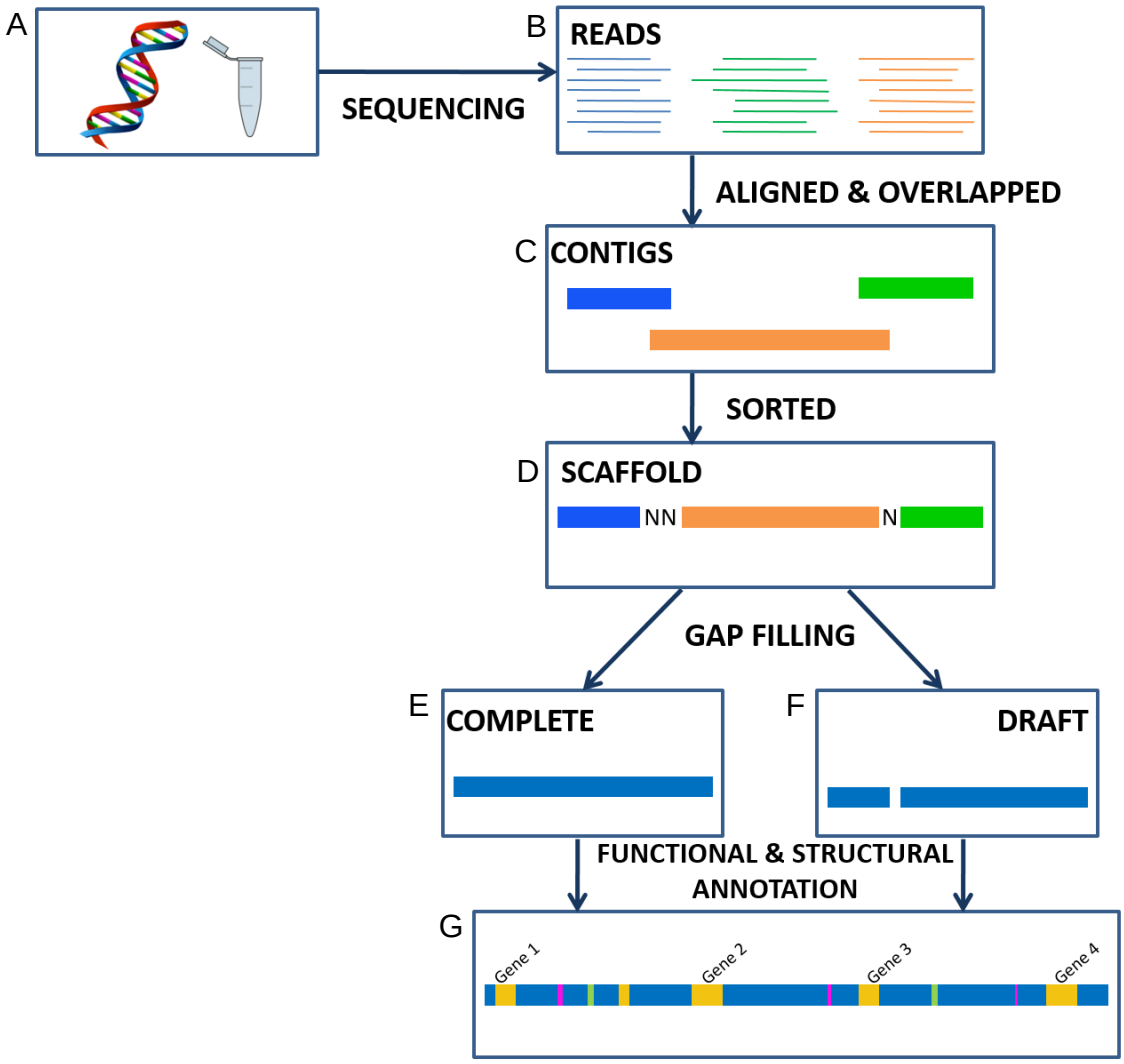

Figure 2 - Most common genomic processes that generate data and metadata.

The sequencing process can be comprehended as the identification of the nucleotides of a certain DNA fragment such as a chromosome, a plasmid or a gene (Figure 2A). The nucleotides compose the DNA molecule as subunits, which are formed by a sugar, a phosphate and a nitrogenous base (SANGER *et al.*, 1977; PIERCE, 2011; MARIANO, 2015). As a result of this technique, the identified fragments of DNA called reads are obtained (Figure 2B), and the application of

bioinformatics techniques in these reads can generate a fragment or a whole genome sequence (MOREIRA *et al*., 2015).

The analysis of the obtained reads to reconstruct the DNA fragment or molecule is called assembly. Those reads are aligned and overlapped to generate *contigs* (Figure 2C), which can be sorted to form a scaffold (Figure 2D), and the remaining gaps can be filled. This process can result in a complete (Figure 2E) or incomplete (Figure 2F) DNA fragment, *e.g.*: the first can be a complete genome and the second can be a draft, which can be further investigated to extract knowledge and enrich our genetic understanding (LANDER *et al*., 2001; MARIANO, 2015; MOREIRA *et al*., 2015; ABURJAILLE *et al*., 2015).

With the bases identified and correctly sorted, the resulting file may be investigated to discover which genes are present in the fragment. This approach is called structural and functional annotation (Figure 2G) and can be done automatically and/or manually. Generally, this is performed first by software and then carefully curated by a human who analyzes the software predictions, correcting them when it is necessary. As a result of this step, a complete genome with identified genes can be achieved (MOREIRA *et al*., 2015; ABURJAILLE *et al*., 2015).

1.1.1.1. Genomic data patterns

According to Brooksbank and Quackennbush (2006), available data is useless unless it is presented in a way that analysis and interpretation are possible. With the genomic data explosion, standardization was needed to share, recover, compare and analyze the produced datasets as well as its related data in the most meaningful way. Those metadata are important to guide the analysis and comparison providing genome features referring to genome's source, isolation, preservation, taxonomic characteristics, among others (FIELD AND SANSONE, 2006; FIELD *et al*., 2008; MIXS PROJECT, 2016). In addition, this type of data can be essential to some studies such as epidemiological, genetics and evolutionary studies,

Since the late 90s, various research communities have been developing standards on omics data, which have influenced software developers, guided the development of public databases and facilitated analyses of this data (THE C.

ELEGANS SEQUENCING CONSORTIUM, 1998; ASHBURNER *et al*., 2000; BROOKSBANK AND QUACKENNBUSH, 2006).

As a result of this effort, international genomic data patterns arose; the Genomic Standards Consortium (GSC) created the Minimum Information about a Genome Sequence (MIGS) in 2008 (BROOKSBANK & QUACKENNBUSH, 2006; FIELD *et al.*, 2008). More recently, this consortium has created the Minimum Information about any (x) Sequence (MIxS) project, which mandates the minimal information to be kept about a genome, metagenome or marker gene sequence (REDDY *et. al.*, 2014; MIXS PROJECT, 2016). This project comprehends three specifications: the already presented MIGS, the Minimum Information about a Metagenome Sequence (MIMS) and the Minimum Information about a Marker Gene Sequence (MIMARKS).

MIGS specification, which is used in this work, is based on the idea that a minimum amount of information is necessary to guide comparative studies, data integration and knowledge generation. Such specification provides a detailed minimum information checklist about genome sequences, standardizing the information to be kept about those sequences. The use of this approach can increase the utility, accessibility and quality of genome sequences data (MIXS PROJECT, 2016).

1.2. Biologic Collections

The Osvaldo Cruz Foundation (FIOCRUZ) defines biological collections as sets of organisms or their parts about which there are available information concerning origin, collection and identification of each specimen (PORTAL FIOCRUZ, 2016). There are many possible biological materials to be kept in a collection; among them we can highlight microorganisms. Those are characterized as organisms which can only be seen with a microscope (PORTER, 1976; SETTE *et al*., 2007). Culture collections of microorganisms keep them as viable cultures which are commonly freeze-dried or cryopreserved.

Microorganism collections catalog, store and provide such specimens, preserving natural diversity as well as maintaining the purity of the strains for future studies. Besides allowing further research, keeping the strains as pure as possible

enables researchers to reproduce and check experiments that have already been done (TORTORA *et al*., 2012; MADIGAN *et al*., 2010).

According to Hewitt and Watson (2013), a facility to collect, preserve, store and supply biological samples and their associated data, following standardized operating procedures and providing those samples to scientific and clinical use, is called a biobank. Besides this definition, the term biobank is widely used in the literature to refer to biological resource centers (BRCs) (VAUGHT, 2016), which can contain: collections of culturable organisms; replicable parts of those; viable but not culturable cells and tissues; databases containing relevant information about the collections; and related bioinformatics systems (OEDC, 2007).

Biological collections, biobanks and BRCs can provide raw material to genetic and omics experiments, which turns them into an important resource tool to genomics studies as well as to every study that uses samples or their associated metadata (VAUGHT, 2016).

1.2.1. Biological collections data guidelines

Biological samples management requires information systems to be successful, according to Casaregola (2016) information management is the key to operate and use culture collections. In this process, not only samples and metadata must be handled, but also standard operation procedures, experiments results and other related processes have to be recorded and made available to researchers (VAUGHT, 2016).

Due to the importance of the management process and its impact on the operation of the laboratory and in the produced results, national and international agencies and consortiums have created data guidelines. Among those, we can highlight the World Federation for Culture Collections (WFCC), the Brazilian Society of Microbiology (BSM), the Organization for Economic Co-operation and Development (OEDC) and the Common Access to Biological Resource and Information (CABRI).

Those provide minimum sets of information to be kept about the samples and standard operational procedures in preservation, authentication, services provided, collaboration, among others (SETTE *et al*, 2007; OEDC, 2007; WFCC, 2010; CABRI,

2016B). The adoption of these procedures ensures the quality of the services and information provided as well as the correct operation of the laboratory that handles the collection. Besides that, it allows the researchers to easily integrate and compare samples information, facilitating knowledge extraction (VITT, 1992; BROOKSBANK & QUACKENBUSH, 2006; CABRI, 2016B).

1.2.2. Management of Biological Collections

Biological collections generate large amounts of data and metadata about the samples and the techniques applied to those, the management of this information is traditionally made by spreadsheets. Currently, researchers are performing larger studies with lots of samples and creating more samples due to modern, cheap and fast techniques. This scenery complicates the use of spreadsheets to retrieve, share, integrate, compare and query the generated data and metadata. Those tasks can demand from the researchers a great amount of effort, or even be impossible to be performed. Besides, this approach can cause data loss or inconsistency, leading laboratories to suffer financial losses, have tired and inefficient researchers that spend too much time organizing data and looking for samples or, worse, find wrong results due to outdated or inconsistent information (LIST *et. al.*, 2014; BLAZEK *et. al.*, 2015; QUO *et. al.*, 2005; RHOADS *et. al.*, 2014).

This scenery caused information systems to emerge aiming to solve these issues and other issues related to laboratory practices, such as experiments results and omics data management. Those belong to the class of the Laboratory Information Systems (LIS), and can provide the information-processing needs of the laboratories (ÇAĞINDI et. al., 2004; QUO *et. al.*, 2005; HENRICKS, 2016).

Even with LIS advantages, academic laboratories are normally not able to afford the costs of commercial software and there is not so much free software to support all laboratories necessities (LIST *et. al.*, 2014). This situation reflects on the management situation of those laboratories as it can be seen on Table 1, which shows the management strategy of some featured collections in Brazil and worldwide.

The American Type Culture Collection (ATCC), Collection of Institute Pasteur (CIP) and German Collection of Microorganisms and Cell Cultures (DSMZ), which are international, use not available software to perform the management. The

Brazilian collections Microbial Culture Collection of Bahia (CCMB), Collection of Bacteria from Environment and Health (CBAS), Collection of Microorganisms and Cells of UFMG (CMC-UFMG), Brazilian Collection of Environmental and Industrial Microorganisms (CBMAI) and LGCM collection use either a free software or spreadsheets to manage the collection.

Table 1 – Management strategies used by some biological collections.

Information sources: 1 - (INSTITUTE PASTEUR, 2016); 2 – (ROMANO, 2005; CABRI, 2016); 3- Personal communications; 4 - (SICOL CBMAI, 2016) 5 - (SICOL CBAS, 2016)

Collection	Digitalization strategy	Availability	Source
ATCC	Not informed	-	2
CIP	CABRI	Not available	2
DSMZ	CABRI	Not available	2
CBAS	SiCol	Free	5
CMC-UFMG	Google drive Spreadsheets	Free	3
CCMB	Spreadsheets	Free	3
CBMAI	SiCol	Free	4
LGCM Collection	Spreadsheets	Free	3

1.2.3. Laboratory Information Systems

With the raise of laboratorial information due to the advances of microbiology as well as to the NGS explosion, information systems have become essential to successfully run a laboratory (SEPULVEDA & YOUNG, 2013; DUBEY *et al.*, 2012). Those can be called by different names such as Laboratory Information Systems (LIS), Laboratory Information Management Systems (LIMS), microbiology LIS and other variations according to the purpose of the system (e.g. systems designed to clinical laboratories can be called clinical LIS). Through bibliographic research, it is possible to understand LIS as any software designed and intended to be used by a laboratory and LIMS as a software designed to be used by a laboratory with the intention to manage the whole laboratory operation or featured parts of it (RHOADS *et. al.*, 2014; SEPULVEDA & YOUNG, 2013; LIST *et. al.*, 2014; ÇAĞINDI et. al., 2004; PRASAD & BODHE, 2012).

Information systems regarding laboratory data have appeared in the early 1980s as centralized systems in the pharmaceutical industry. Those evolved along with the technology going to web-enabled (1996) and finally to fully web oriented

systems (2004) (PRASAD & BODHE, 2012). This change allowed information to be shared among researchers and easily recovered via the usage of a web browser. The purpose of LIMS also evolved; it started for results and financial management and today it is focused on providing a user-friendly and integrated solution to manage the information generated in the laboratory (ÇAĞINDI et. al., 2004; PRASAD & BODHE, 2012; SEPULVEDA & YOUNG, 2013).

With the use of information systems in the laboratory environment, it has been proven that those tools can improve microbiology laboratories' efficiency, accuracy, precision and rapidity (RHOADS *et. al.*, 2014). Significant amounts of time can be saved by the use of a LIMS considering that information is available, easily accessible and interpretable. It allows the researchers to concentrate on experiments and results generation instead of data or samples management problems (TAGGER, 2011, LIST *et. al.*, 2014). The usage of LIMS can drastically contribute to the quality, efficiency and competitiveness of a laboratory (ÇAĞINDI et. al., 2004).

A great amount of commercial software exists to this purpose, but those are not affordable to academic laboratories, which normally use open-source systems. According to List *et. al.* (2014) open-source LIMS are often customized to some specific activity or data as it is shown in Table 2. Among the possible focuses of those systems, this work concentrates on the management of the laboratory in general, microbial samples, genomic data and the related metadata.

Table 2 – Examples of LIMS software

Software name	Main purpose	Reference
BikaLIMS[1]	Clinical studies	-
LabIs	Substance synthesis studies	(BLAZEK *et al.*, 2015)
Omics metadata management software	Next-generation sequencing projects	(PEREZ-ARRAIGA *et al.*, 2015)
OpenLabFramework	Vector constructs and cell line library sample tracking	(LIST *et al.*, 2014)
openBIS ELN-LIMS	Academic scientific project management	(BARILLARI *et al.*, 2016)
MendelLIMS	Clinical genome sequencing	(GRIMMES & HANLEE, 2014)
SAVANAH	Management of High-Throughput Screening Libraries	(LIST *et al.*, 2016)

Based on the literature, some important features of this kind of LIMS have been stated as shown on Table 3.

[1] Available in: <https://www.bikalims.org/>

Table 3 – Important features regarding LIMS for the management of laboratory in general, microbial samples, genomic data and the related metadata that are designed for academic laboratories.

Feature	Justification/Description	Reference
Open-source code	Open-source code allows academic laboratories to use the software without license/maintenance costs as well as to add new functionalities to those	(LIST *et. al.*, 2014)
Documentation	A good documentation is essential to allow software easy maintenance and functionalities addition, and support users to learn how to use the software	(LIST *et. al.*, 2014; WILSON *et. al.*, 2014; SOMMERVILE, 2011)
Standard operation procedures support	Standard Operation Procedures (SOP) support, improve, or even guarantee (when the procedures are strictly followed) the quality of the laboratory's service and results	(HEWITT & WATSON, 2013; SEPULVEDA & YOUNG, 2013)
Sample management and tracking	The management of the samples and the related metadata are almost impossible to be done safely and efficiently without information systems. In addition, tracking the physical localization of the sample facilitates the organization and management of those.	(WFCC, 2010; LIST *et. al.*, 2014; ÇAĞINDI et. al., 2004)
Sample receipt and sending (flow) support	The description of WFCC records about the samples and the description of an "ideal LIS" state that the samples flows must be recorded; also, recording information to huge amounts of samples is difficult without a software	(WFCC, 2010; SEPULVEDA & YOUNG, 2013; KAMMERGRUBER *et. al.*, 2014)
Sample collections international guidelines	Following international guidelines to keep data allows the generated data to be useful to future research through facilitated comparison against other standardized data	(SETTE *et al*, 2007; OEDC, 2007; WFCC, 2010; CABRI, 2016B)
Bibliography support	The recovery of bibliography without digging into directories or public databases accelerates discoveries and saves researchers' effort	(TAGGER, 2011)
Web-based	Easy access to data is essential to LIMS software. A web-based system can provide this functionality and facilitate data sharing	(TAGGER, 2011)
Genomic data support	Due to exponential increase of quantity and quality of genomic data, efficient approaches to manage this data are required	(GRIMES & HANLEE, 2014; KYRPIDES *et. al.*, 2014)
Genomic data patterns	The adoption of international data patterns allows researchers to efficiently compare and share information	(FIELD *et. al.*,2008; FIELD *et. al.*,2011)
Reports support	Reports are useful to biologists by helping in the knowledge extraction and management of the laboratory activities	(SEPULVEDA & YOUNG, 2013; LIST *et. al.*, 2014)
Security	User login and different roles are needed to elevate the security of the system	(SEPULVEDA & YOUNG, 2013; LIST *et. al.*, 2014)
Audit-logging	The security of the LIMS can be enhanced by the storage of information insertions, modifications and deletions performed in the software as well as the user who did that.	(LIST *et. al.*, 2014; GRIMES & HANLEE, 2014)

.

During the bibliographic research done to this work, it was not found a unique LIMS to manage the laboratory in general, microbial samples, genomic data and the related metadata. Consequently, it was not found systems to meet the previously presented important features. This situation emphasizes the necessity of the development of a LIMS to support the microbiological laboratories that deal with biological collections and sequencing data.

1.3. Software Engineering

According to Sommerville (2011), Software Engineering is the discipline that focuses on all aspects of the software production, from the early stages of discovering the software purpose to the maintenance phase. The set of activities that leads to the production of a software product is called software process, and it can be generally divided into four fundamental activities:

(a) Software specification: In this stage, functionalities and constraints on the software's operation are defined.
(b) Design and implementation: Ways to meet the designed specifications are elaborated and implemented in this activity.
(c) Validation: In this activity, it is checked whether the software meets the users' needs.
(d) Evolution: Over time, users' needs may change. This activity consists of the product evolvement to meet the new necessities.

Software process models are representations of the software processes, which serve as guidelines to the development. Those can be categorized as plan-driven and agile processes. The first states that all activities must be planned in advance and the success is measured against the plain. The second treats planning as an incremental activity that can change according to the user's requirements changes.

The previously described activities can be differently organized according to the chosen software process model, sometimes being performed in sequence, interleaved or organized in different manners. Among those activities, specification

(also called requirements engineering), is a critical stage of the software process because of its impact in the resulting product. Bad specifications can lead software to hamper users' life through an unreal or distorted representation of the automated processes, or even be useless (SOMMERVILLE, 2011; PRESSMAN, 2011). The software requirements process can be subdivided into four activities as it can be seen in Figure 3.

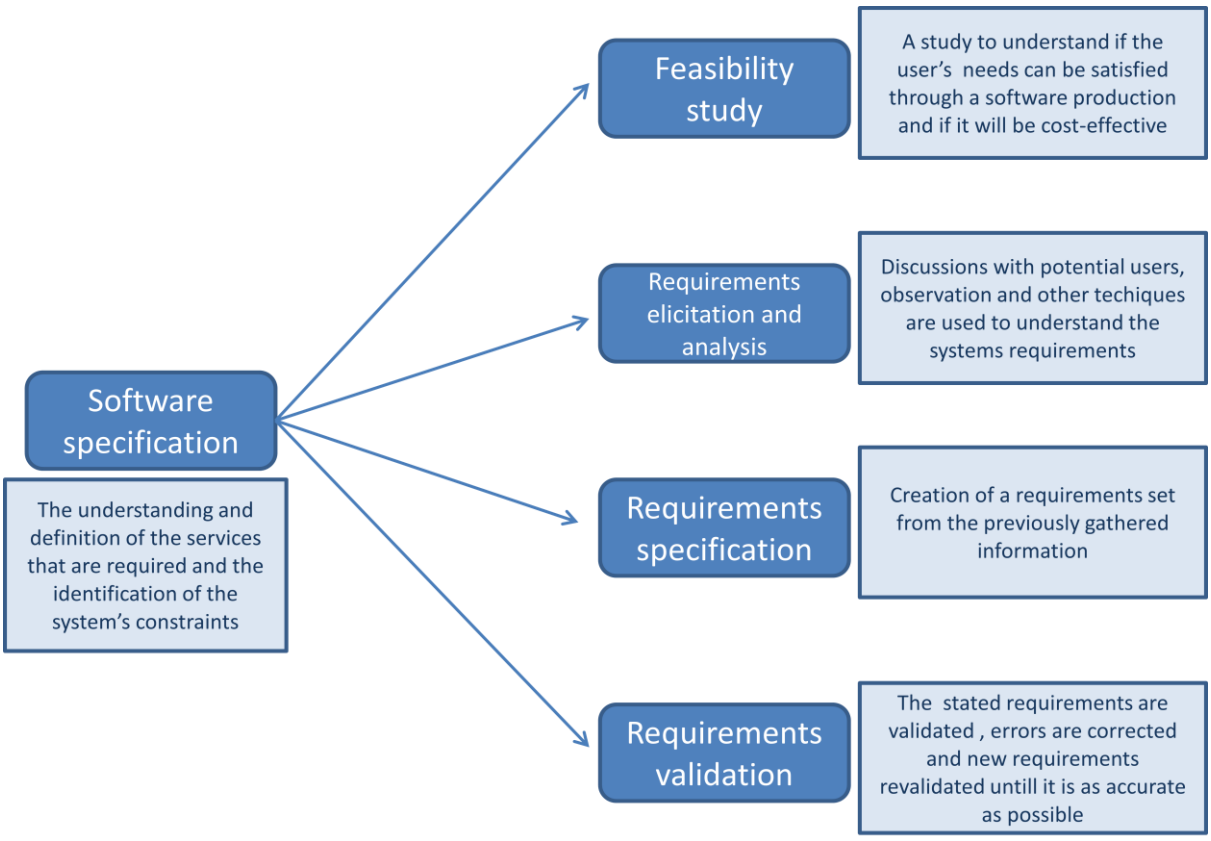

Figure 3 – Scheme of software specification general sub-activities based on Sommerville (2011)

The execution of those sub-activities with attention and care can lead to good specifications. It facilitates successful software construction, meeting the users' needs and really enhancing the automated processes (SOMMERVILLE, 2011). To LIMS software, as to any kinds of software, requirements engineering is crucial. A proper study of the laboratory routine and functionalities is essential to a really helpful LIMS (SOMMERVILLE, 2011; PRASAD & BODHE, 2012).

According to Casaregola (2016), in the construction of information systems to culture collections, the opinion of the users is rarely taken into account. This can

affect the level of comprehension of the developers of the software requirements and, as a consequence, negatively affect the impact of the software in the researcher's life (CASAREGOLA *et al.*, 2016; SOMMERVILLE, 2011).

To ensure that the specified needs are addressed in the system, tests are performed. Those are essential to evaluate the usefulness of the software and an important part of the software process. Aller and Salazar (2016) highlighted the necessity of submitting the system prototype to a real laboratory (SOMMERVILLE, 2011; ALLER & SALAZAR, 2016).

2. MOTIVATION

In the past decade, the NGS revolution has generated genomic data in an exponential rate and allowed more comprehensive analysis of this data, achieving better results and scientific conclusions. In addition to the possibilities of research, such genomic explosion brought challenges on data management and standardization, which have not been completely solved until the present day (FIELD *et al*., 2011; FIELD *et al*., 2008). Along with this explosion, data regarding laboratory experiments have also increased in quality and quantity due to modern, fast and cheap techniques, creating more challenges for researchers to manage and extract good results from the generated data (DUBEY *et al.,* 2012).

According to Krypides *et al.* (2014), DNA sequencing information is essential to collections, bringing this NGS data challenge to the collections reality (GRIMES & HANLEE, 2014). Sample collections data and metadata influence in other studies and may ruin them if the information is wrong or outdated (LIST *et. al.*, 2014; BLAZEK *et. al.*, 2015; QUO *et. al.*, 2005; RHOADS *et. al.*, 2014). Currently, biobank facilities are required to provide the related metadata with the samples to the users, turning an effective management even more critical (HEWITT & WATSON, 2013).

Larger studies have been allowed through the previously presented scenarios, comparing thousands of samples and/or genomes and achieving amazing results. However, the process of comparison can be laborious or even impossible due to non-standardized data, taking unnecessary time and effort from researches and turning standardization essential to store and publish data (BROOKSBANK & QUACKENNBUSH, 2006; FIELD *et al.*, 2011; GRIMES & HANLEE, 2014).

To solve these problems, laboratories have been using spreadsheets or LIMS to manage huge amounts of laboratorial data as well as to manage sample collections (TAGGER, 2011). As stated before, the spreadsheets approach can take a great amount of effort to the researchers and be costly, besides the fact that this approach is more likely to present errors due to human-guided data manipulation. (DUBEY *et. al.*, 2012; LIST *et. al.*, 2014; RHOADS *et. al.*, 2014)

LIMS is the ideal tool to improve the management scenario and drastically contribute to the quality of the generated results as well as to the efficiency and competitiveness of a laboratory (ÇAĞINDI et. al., 2004; RHOADS *et. al.*, 2014). In

the bibliographic research done to this dissertation, it was not found a unique LIMS to support the management of the laboratory in general, microbial samples, genomic data, and the related metadata. It highlights the necessity of the development of a new LIMS tool that embraces the previously presented needs, providing a helpful and complete bioinformatics tool to manage genomic data, microbial samples and the laboratory in general.

3. AIMS

3.1. General aim

Develop a LIMS software to support the management of genomic data, microbial samples and the laboratory in general, handling the generated data regarding *in vivo*, *in silico* and *in vitro* techniques.

3.2. Specific Aims

- Understand researchers' needs concerning biologic samples, genomic data and laboratorial data in general.
- Identify data patterns concerning microbiological genomic and samples data.
- Design, develop and make available the LIMS created according to the previous findings.

4. METHODOLOGY

4.1. Software development methodology

LabControl's development was inspired on agile software process models according to Somerville (2011) and Pressman (2011). In this methodology, the previously explained activities of the software process (specification, design and implementation, and validation) are done in cycles which are repeated until the end of the software production, as shown in Figure 4. When maintenance (software evolution) is needed, the same process is done.

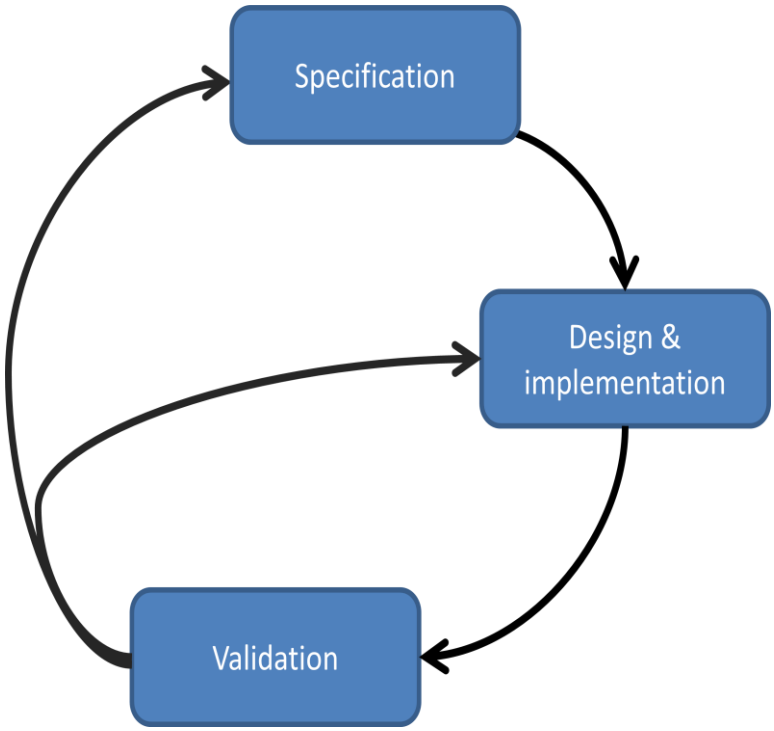

Figure 4 – Software process cycle used in LabControl's production

In the very beginning of the software production, more time was spent in specification to state the first set of requirements, which was designed and validated with potential users. Then, database design was performed based on the stated requirements and followed by implementation and validation of the designed database with potential users. Afterwards, the software itself was designed and the implementation was performed to small parts of the product followed by the validation of the coded parts. When some problem appeared in the validation stage,

the coded part was re-designed and re-implemented, and if necessary, the specifications were refined or updated. This cycle continued until the end of the software's implementation and testing, considering that the more parts of the software were ready, the more comprehensive tests were made. A more detailed explanation of the activities performed in each stage can be seen as follows.

4.1.1. Specification

Before the proposal of this dissertation, a feasibility study was performed to understand if a LIMS would properly support the management of the laboratorial data and if there was a necessity to produce a new LIMS, considering that the laboratory that supported this work (Laboratory of Cellular and Molecular Genetics, LGCM) has suffered because of inadequate management.

The requirements elicitation and analysis process was performed in three steps, as follows.

(a) Identification of stakeholders: people who can be benefited from this system directly or not were identified in this step.

(b) Interviews: it was performed interviews with bioinformaticians that work with NGS microbial data analysis, wet-lab researchers that use microbial strains in their research, researchers that work both in *in silico* analysis and wet-lab experiments (hybrids). The number of researchers interviewed by area is summarized in Table 3, 16 people were interviewed in total.

Table 4 - Number of researchers interviewed by area

Researchers	Number of researchers
Bioinformaticians	4
Wet-lab researchers	6
Hybrid researchers	6
Total	16

(c) Comprehension of the automated processes: the modeled (automated) processes were studied through bibliographic research, observation of researchers' routine of information management and usage, and analysis of the management technique used in LGCM (spreadsheets).

Step (a) was performed at first, and then steps (b) and (c) were performed in parallel, considering that (a) was performed according to Pressman (2011) and (b) according to Sommerville (2011). After this, the requirements were stated (requirements specification) and validated with the stakeholders. This statement and validation was repeated until the requirements met the users' needs. The requirements were divided into two main categories: user requirements and system requirements. The first one refers to an abstract description using natural language of the services provided by the system as well as its restrictions. The second one exposes what is going to be implemented and it is divided into functional and non-functional system requirements. Functional system requirements represent the system functions, and non-functional system requirements represent demands such as security, availability and usability. All requirements were classified in essential, important or desirable, depending on their role in the system (SOMMERVILLE, 2011).

4.1.2. Design and implementation

At first, the architecture of the software was designed and the used technologies were chosen. The architecture was designed based on the concepts shown in Silveira *et al.* (2012). Afterwards, the database was projected using Astah[2] tool and the implementation of both software and database started. Software implementation was done using NetBeans[3] Integrated Development Environment (IDE) and Java programming language; database implementation was done using PostgreSQL[4] database. The framework Hibernate[5] was used to the object-relational mapping between PostgreSQL and Java, the framework Spring[6] was used to Dependency Injection (DI), Inversion of Control (IoC) and to REST service implementation; it was configured by using Spring Boot[7]. The framework Spring

[2] Available in: <http://astah.net/download>
[3] Available in: <https://netbeans.org/downloads/>
[4] Available in: <https://www.postgresql.org/download/>
[5] Available in: <http://hibernate.org/orm/
[6] Available in: <http://projects.spring.io/spring-framework/>
[7] Available in: <http://projects.spring.io/spring-boot/>

Security[8] was used to support the security demand. To interface implementation, it was used HTML5, CSS3, Bootstrap[9] and AngularJS[10].

4.1.3. Validation

LabControl has been validated according to Sommerville (2011) and it can be understood in three different processes: development testing, system testing and acceptance testing. Development testing is the process of independently testing every constructed piece of software in order to ensure that this component or function is properly working. System testing consists of checking if the constructed and already tested components can work together without failures; both development testing and system testing have been performed by the developers. Acceptance testing can be comprehended as the final stage of software testing, where the entire system is tested by the developer and the users or potential users of the software (SOMMERVILLE, 2011).

[8] Available in: <http://projects.spring.io/spring-security/>
[9] Available in: <http://getbootstrap.com/>
[10] Available in: <https://angularjs.org/>

5. RESULTS

5.1. Requirements

5.1.1. User requirements

LabControl shall store, manage and make data regarding microbiology laboratories with focus on NGS and samples available through a web interface. All kinds of handled information can be added, changed and visualized; some information can be excluded or deactivated. All actions performed in the system must be stored as a history, and exclusions, activations or deactivations have to be associated with a reason. Those peculiar actions can be only executed by administrators.

In a general view, LabControl models the data according to the workflow showed in Figure 5. With a given sample it is possible to associate it with a biological collection, register the usage of SOPs and register the related bibliographic production. Besides, it is possible to register related sequencings, assemblies, annotations and submissions to public databases.

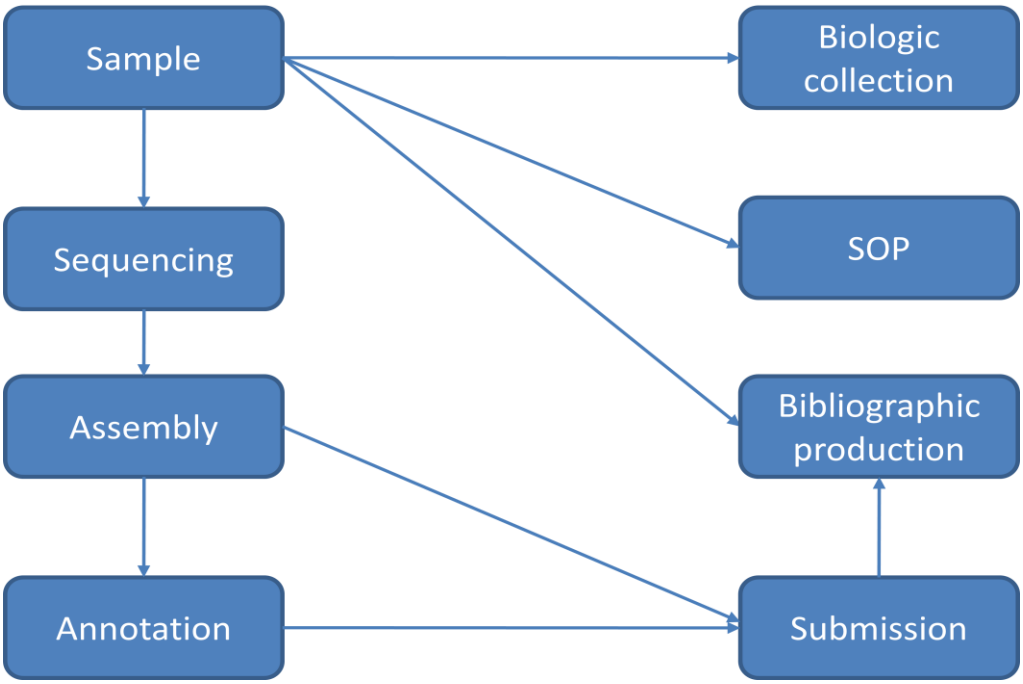

Figure 5 – LabControl's general workflow view

A detailed description of the system consisting of the handled information and its restrictions is organized by the different handled topics and presented as follows. In this description, the method or methods used to elicit the stated requirements are presented in checkboxes at the end of each section (see 4.4.1 Specification to a detailed explanation of those methods).

5.1.1.1. History

A history concerning all modifications performed on the stored data has to be maintained. Any addition, modification or deletion has to be kept along with the date and the user that performed the action; in case of deletion or deactivation (a kind of modification), a reason must be stored. Such modifications have to be stored in chronological order.

- [x] Interviews and observation of researchers' routine
- [x] Bibliographic research
- [ ] Spreadsheets analysis

Used references: (ÇAĞINDI *et. al*., 2004; SEPULVEDA & YOUNG, 2013)

5.1.1.2. Collaborators

In the software, collaborators are people or institutions that participate in any stage in the processes modeled by LabControl. Concerning those, the system has to store the name, the related laboratory and an email; the latter must be unique to each collaborator. Those can only be excluded if they are not recorded in the history and if there is no user associated to them. Collaborators can be activated or deactivated, and can only be excluded if there is no mention about them in the history.

- [x] Interviews and observation of researchers' routine
- [ ] Bibliographic research
- [ ] Spreadsheets analysis

5.1.1.3. Users

A registered user is required to use the system. In addition to a related collaborator, it has to be kept a username in “name.surname” format, a password consisting of a minimum of eight characters, the type of the user and the status (activated or deactivated). A user can only be excluded if there is no related record in the history, otherwise it can be just activated or deactivated. The registration of new users can only be done by administrators, and each user can edit their own user.

In the system, the users can take three roles: administrator, researcher or guest. Administrators have access to all functionalities of the software, researchers are not allowed to delete and activate/deactivate, and guests are only allowed to visualize the stored data, except for history data.

☒ Interviews and observation of researchers’ routine
☐ Bibliographic research
☐ Spreadsheets analysis
Used references: (OEDC, 2007; SEPULVEDA & YOUNG, 2013; LIST et. al., 2014)

5.1.1.4. Strains

In LabControl, strains can only be excluded if there is no register related to it, and that can only be done by an administrator, with a given reason. The stored information about the strains shall consist of a current strain code, a customized strain code, an old strain code, ploidy, propagation, number of replicons, trophic level, identifier, depositor and the information regarding to growth condition, isolation, preservation, collection, taxonomic characteristics and additional characteristics.

Old strain code is used in the case of current strain code be changed, to facilitate the transition and maintain the reference to the old code. A situation that requires this change is the genetic modification of a strain, for example *Lactococcus lactis* strain NZ9000, which is a genetic modification of *L. lactis* strain MG1363 (LINARES *et al.*, 2010).

The customized code is composed by the acronym of the collection or the owner laboratory plus the current strain code; this information is used to define the name of the strain in the managed collection. For example, if it was registered a strain with current strain code “CP1002” from the laboratory “LGCM”, the customized code should be LGCM_CP1002, the underline should be added to facilitate reading.

The remaining information is presented with details as follows.

☒ Interviews and observation of researchers’ routine ☐ Bibliographic research ☐ Spreadsheets analysis Used references: (SETTE *et al*, 2007; FIELD *et al*., 2008; WFCC, 2010)

5.1.1.4.1. Collection

Data concerning how a sample was collected should contain collection date, geographic localization (city, state, country, geographic coordinates and depth), the biome in which the sample was collected and the environment description. The latter can comprehend humidity conditions, dust conditions, oxygen level and other conditions of the environment where the strain was collected.

Additionally, the host or substrate is stored as well as the localization of the strain in the host, the host’s health state, the biotic relationship and the used SOP.

☒ Interviews and observation of researchers’ routine ☐ Bibliographic research ☐ Spreadsheets analysis Used references: (OEDC, 2007; WFCC, 2010; VAUGHT, 2016)

5.1.1.4.2. Isolation and preservation

Concerning the methodology used to isolate the sample, it is relevant to keep the isolation date, the collaborator who isolated the strain, the used SOP, and any additional information shall be kept as comments. About the preservation

methodology, it is necessary to store the used SOP and additional information if necessary; as in the preservation, additional information shall be kept as comments.

☒ Interviews and observation of researchers' routine ☐ Bibliographic research ☐ Spreadsheets analysis Used references: (OEDC, 2007; SETTE *et al*., 2007; WFCC, 2010; VAUGHT, 2016)

5.1.1.4.3. Growth conditions

Considering strain growth conditions, LabControl shall store the growth medium, relevant temperature or temperatures to the growth (optimal, minimum, maximum and others), pH, environmental conditions, humidity conditions, CO_2 condition, and other specific conditions.

☒ Interviews and observation of researchers' routine ☐ Bibliographic research ☐ Spreadsheets analysis Used references: (WFCC, 2010; SETTE *et al*., 2007; VAUGHT, 2016)

5.1.1.4.4. Taxonomic characteristics

The basic taxonomic characteristics to be kept in the system are domain, kingdom, phylum, class, order, family, gender and species (TORTORA *et al.,* 2012). Besides, other taxonomic classifications can exist as biovar concerning *Corynebacterium pseudotuberculosis* (GUEDES *et al*., 2015) or serovar concerning *Leptospira interrogans* (MARSHALL *et al.*, 1981); those must be stored as well.

☒ Interviews and observation of researchers' routine
☐ Bibliographic research
☐ Spreadsheets analysis
Used references: (SETTE *et al*., 2007; WFCC, 2010; RHOADS *et al.*, 2014)

5.1.1.4.5. Extrachromosomal elements

Extrachromosomal elements can be associated to each strain. Concerning those, it shall be stored a name, the extrachromosomal element type (e.g. plasmids and viruses) (MCGEOCH & BELL, 2008), and any additional information shall be kept as comments.

☒ Interviews and observation of researchers' routine
☐ Bibliographic research
☐ Spreadsheets analysis
Used references: (FIELD et. al., 2008)

5.1.1.4.6. Other characteristics

Besides the already presented strains' metadata, those can present different features such as biochemical characteristics, for example the ability of nitrate production and the color found in Gram's method (GUEDES *et al*., 2015). Features as security level or other regulatory conditions can be stored as other characteristics as well. It must be possible to add as many features as necessary.

☒ Interviews and observation of researchers' routine
☐ Bibliographic research
☐ Spreadsheets analysis
Used references: (WFCC, 2010; VAUGHT, 2016)

5.1.1.5. Strains' sending and receiving flow

Strains's sending and receiving flow has to be registered. About both flows, it has to be kept the purpose, sending responsible, receiving responsible, laboratory of origin and receiving laboratory, sending date, receiving date, the sent/received strain, sending or receiving date and a protocol number.

Protocol number:
-for sending flow: S + sending lab abbreviation + receiving lab abbreviation + id
-for receiving flow: R + receiving lab abbreviation + sending lab abbreviation + id

☒ Interviews and observation of researchers' routine
☐ Bibliographic research
☐ Spreadsheets analysis
Used references: (WFCC, 2010; SEPULVEDA & YOUNG, 2013; KAMMERGRUBER et. al., 2014)

5.1.1.6. Freezer

About the freezer/freezers that house the managed collection, it must be kept the number of floors, shelves, horizontal and vertical drawers and the maximum dimension of the box that keeps the strains. The freezer structure and maximum box dimension can be seen in Figures 6 and 7, respectively.

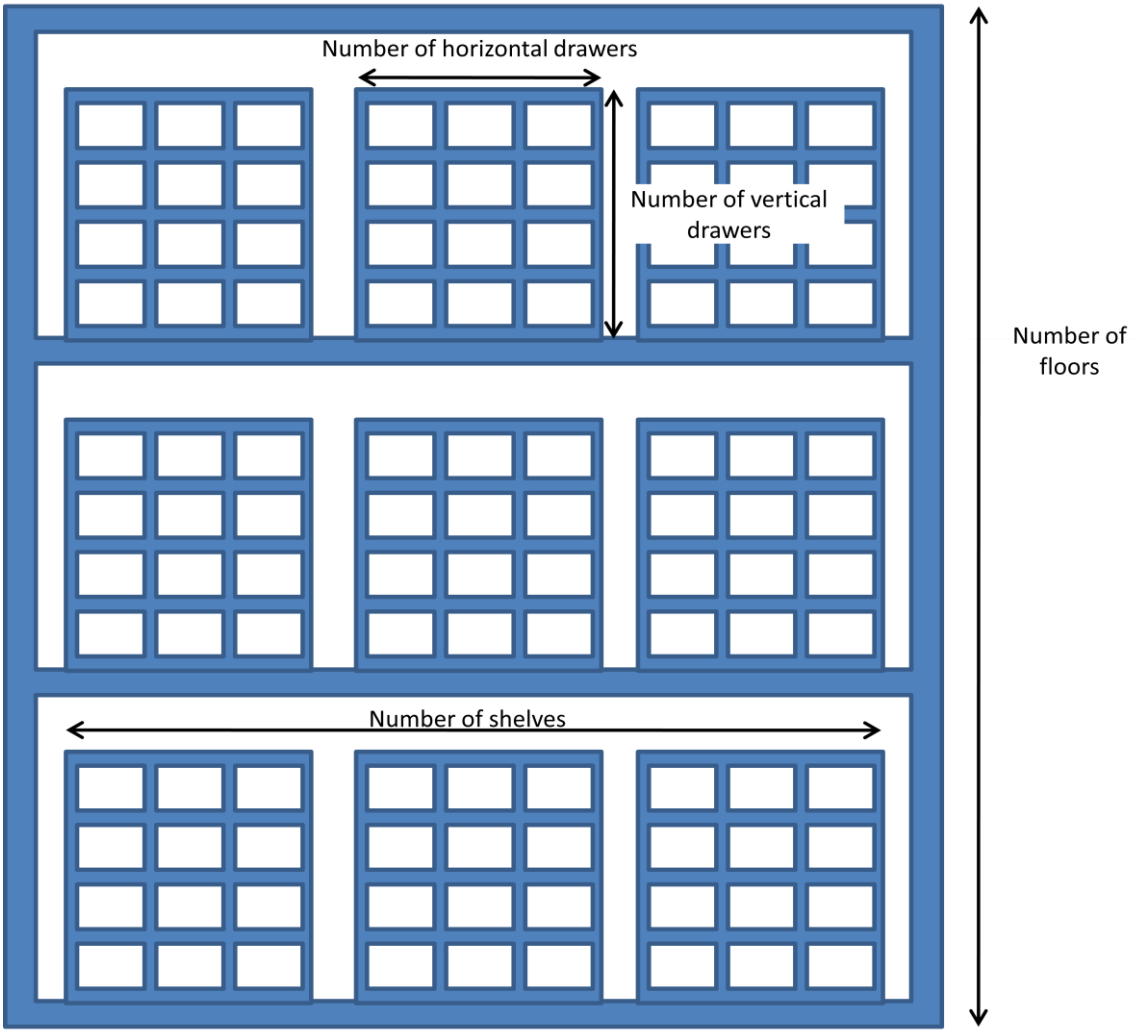

Figure 6 – Freezer that stores the biological collection

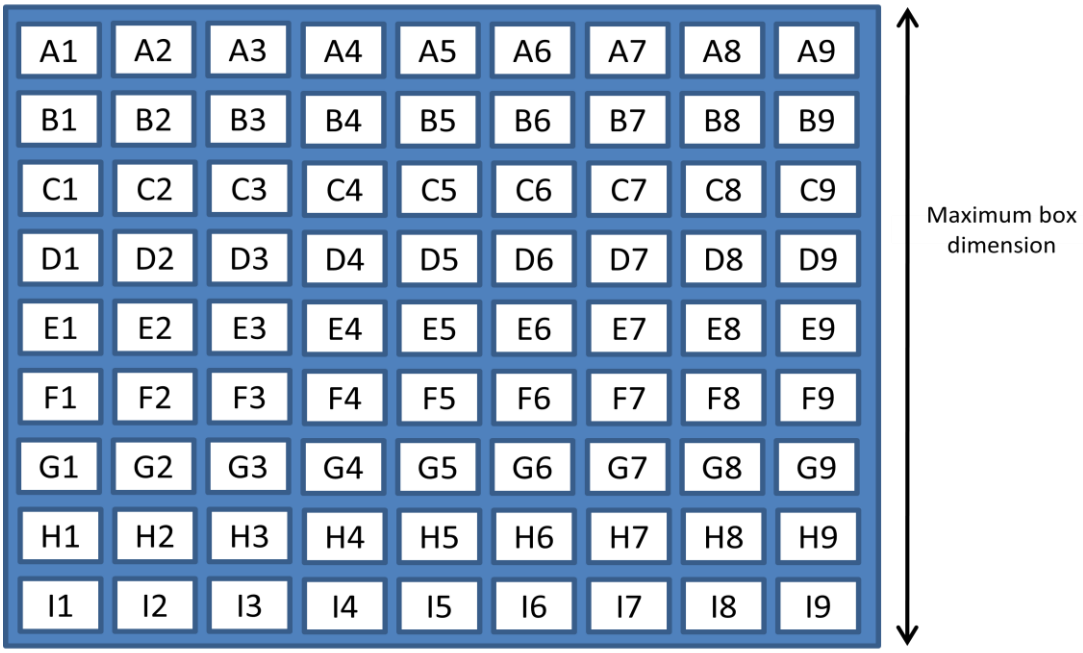

Figure 7 – Box where the strains are in

To track the strains in the biological collection, a tracking number is used. This number is created according to Figure 8, and represents the exact localization of a strain.

freezer number . floor number . shelf number . drawer number . box localization

Figure 8 – Template of the tracking number

For example, to register that a strain in the freezer number one, second floor, third shelf, tenth drawer and twenty-second place in the box, it should be used the number 1.2.3.10.22. A strain can be located in more than one place in the biologic collection.

- [x] Interviews and observation of researchers' routine
- [ ] Bibliographic research
- [ ] Spreadsheets analysis

Used references: (KAMMERGRUBER et. al., 2014; LIST *et al*., 2014; VAUGHT, 2016)

5.1.1.7. Standard operating procedure

The SOPs applied to the strains or the data concerning those shall be stored on LabControl in file format. A SOP is comprehended as a document that determines instructions to perform specific or common operations in a laboratory (CORREIA, 2005), which can be a technique or several techniques to be performed *in vivo*, *in vitro* or *in silico*. A SOP, as well as multiple SOPs, can be applied to one or more strains.

When the SOPs are applied, it shall be stored the purpose of the usage, the used strains, the collaborators who participate and the responsible one, the laboratory where it was done, the result (in file format); additional information can be stored as comments.

☒ Interviews and observation of researchers' routine ☐ Bibliographic research ☐ Spreadsheets analysis Used references: (VITT, 1992; HEWITT & WATSON, 2013; SEPULVEDA & YOUNG, 2013)

5.1.1.8. Sequencing

The system shall manage sequencing data; about that, it should be stored the sequencing technology and platform, the used library and kit, the pair distance, the expected coverage and size, depth of coverage, the quantity of reads, the SOPs used to the sequencing and genetic material extraction, the date, the status, the strain source of the genetic material, a name and the collaborators who worked in this process as well as the responsible one.

☒ Interviews and observation of researchers' routine ☐ Bibliographic research ☐ Spreadsheets analysis Used references: (FIELD *et al.*, 2008; KRYPEDES *et al.*, 2014)

5.1.1.9. Assembly

Data regarding assembly shall be stored in LabControl; about that, it should be stored the assembly type, which can be *ab initio* or by reference; in case of reference assemblies, it is important to store which genome was used as reference. Data concerning the quality of the assembly such as N50, genome size, initial and final number of contigs, number of gaps, biggest and smallest contigs size must be stored as well. The programs used in the assembly are kept, as well as their purpose; additional information about the software usage, such as the used parameters can be stored as comments. If optical mapping information is used to help in the assembly process it should be kept, as well as the used software.

The used SOP, the collaborators who participated in the assembly and the responsible one should be stored together with the date, the status, the number of the project in SIMBA, a given name, the purpose of the assembly, the referred sequencing, the generated data and additional information.

☒ Interviews and observation of researchers' routine
☐ Bibliographic research
☐ Spreadsheets analysis
Used references: (FIELD *et al.*, 2008; EDWARDS & HOLT, 2013; KRYPEDES *et al.*, 2014; SIMS *et al.*, 2014; MARIANO, 2015)

5.1.1.10. Annotation

Considering the annotation, it shall be stored the used SOP, the situation, the date, the responsible collaborator and the ones who participate in the annotation, and the referred sequence. The used programs and databases should be kept as well as the purpose of the usage. Additional information shall be kept as comments.

☒ Interviews and observation of researchers' routine
☐ Bibliographic research
☐ Spreadsheets analysis
Used references: (FIELD *et al.*, 2008; EDWARDS & HOLT, 2013; KRYPEDES *et al.*, 2014)

5.1.1.11. Submission

The data to be kept about the submission of a genome to a public database is the type of the submission, the code and name of the forms that are submitted with the sequence, the responsible collaborator, the authors, the status (in progress, accepted, etc.), the sending and acceptance date, the associated SOP, a given name, and additional information.

The final sequence must be kept as well. About that, it is important to store the associated assembly and annotation, the content of each nucleotide in the sequence, the locus tag, the genbank id, the date of finishing, a given name and the sequence itself.

☒ Interviews and observation of researchers' routine
☐ Bibliographic research
☐ Spreadsheets analysis
Used references: (VARMUS, 2002; BARRET, 2012)

5.1.1.12. Bibliographic production

Bibliographic production concerning the stored samples shall be kept in LabControl. In the system, there is the bibliographic production in general and the articles published by the laboratory that uses the system, which are a type of production. Concerning the general one, it must be kept the authors, the title, the DOI, the date of publication and the file or link of the production. About the second one, it is kept the collaborator responsible for the publication, the journal, the situation (*e.g.* in progress, sent, accepted, under correction, not accepted), the

sending date, the acceptance date; additional information shall be stored as comments.

☒ Interviews and observation of researchers' routine ☐ Bibliographic research ☐ Spreadsheets analysis Used references: (PRASAD & BODHE, 2012; CASAREGOLA *et al.*, 2016)

5.1.1.13. Software and databases

About the software and databases used in the processes modeled by the system, it shall be stored the name, the version, and a reference link or document (article, manual, etc.).

☒ Interviews and observation of researchers' routine ☐ Bibliographic research ☐ Spreadsheets analysis Used references:

5.1.2. System requirements

The system requirements were designed according to the user requirements, and are divided into functional and non-functional ones, as it was stated in the section 4.1.1 Specification of this work. The functional requirements represent the functions that the system should offer, and those are classified into desirable, important and essential ones. All the 102 functional requirements can be seen in the appendix I. The non-functional requirements are constrains in the system's services or functions, as shown bellow.

The system should:

- be available online,
- be secured against non-authorized login,

- have an easy and user-friendly interface,
- be in English but have support to other languages as well,
- be well documented.

5.2. Database

The database used in this system was designed based on the user and system requirements (refer to 5.1). The database was designed as an entity-relationship diagram (Figure 9) and then implemented using PostgreSQL.

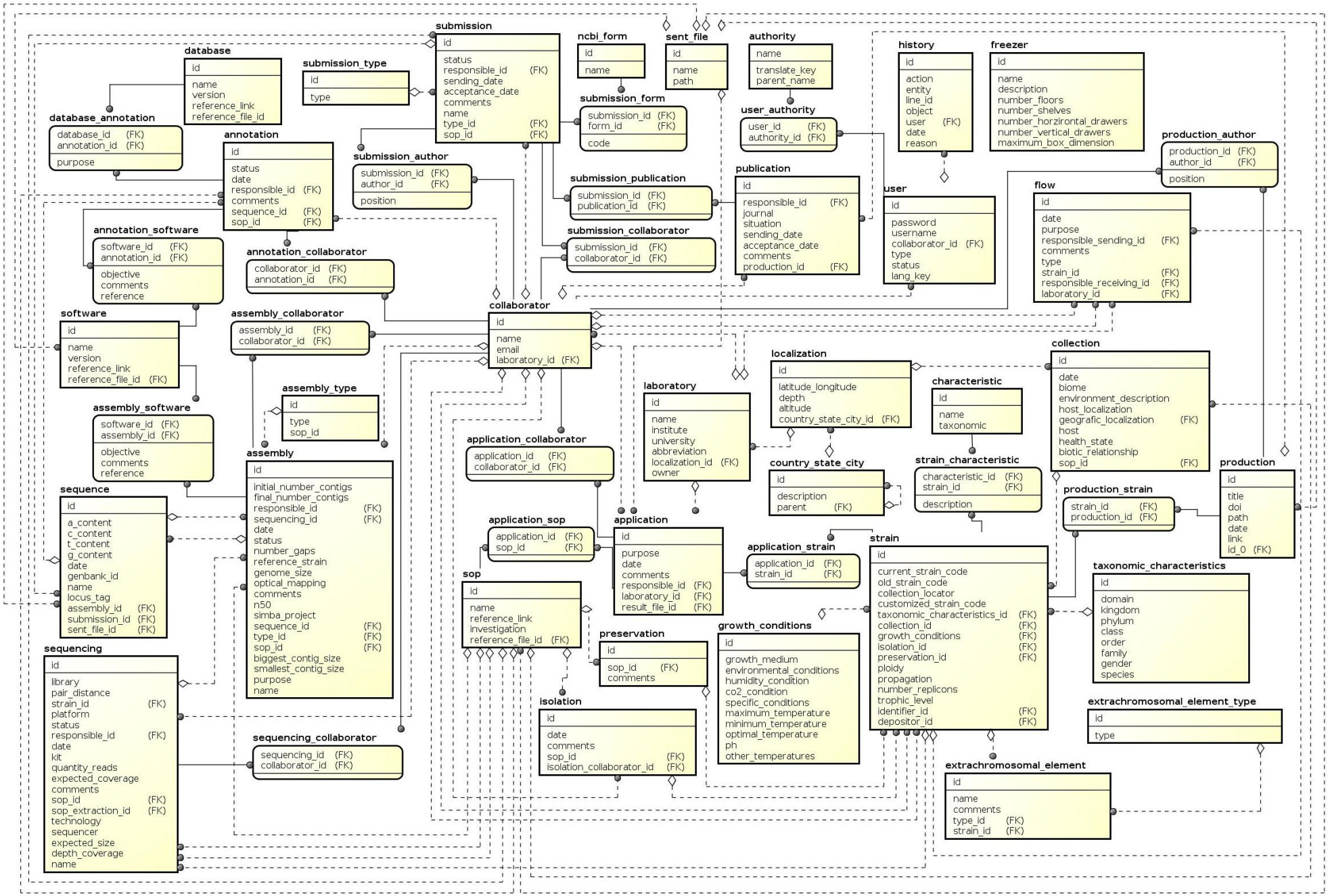

Figure 9 – Entity-relationship diagram

5.3. Software architecture and implementation

LabControl's architecture is based on Model-View-Controller (MVC) pattern (KRASNER & POPE, 1988; LUCKOW & MELO, 2010); a representation of that and the used technologies is shown in Figure 10. This architectural pattern divides the system into three layers of code: model, view and controller. The view is responsible for the visual part, which is the one that interacts with the user through the browser. The model is responsible for the operations related to the data handled by the system, such as database access and restrictions handling. The controller is responsible for linking the other two layers through delivering of data from one to another.

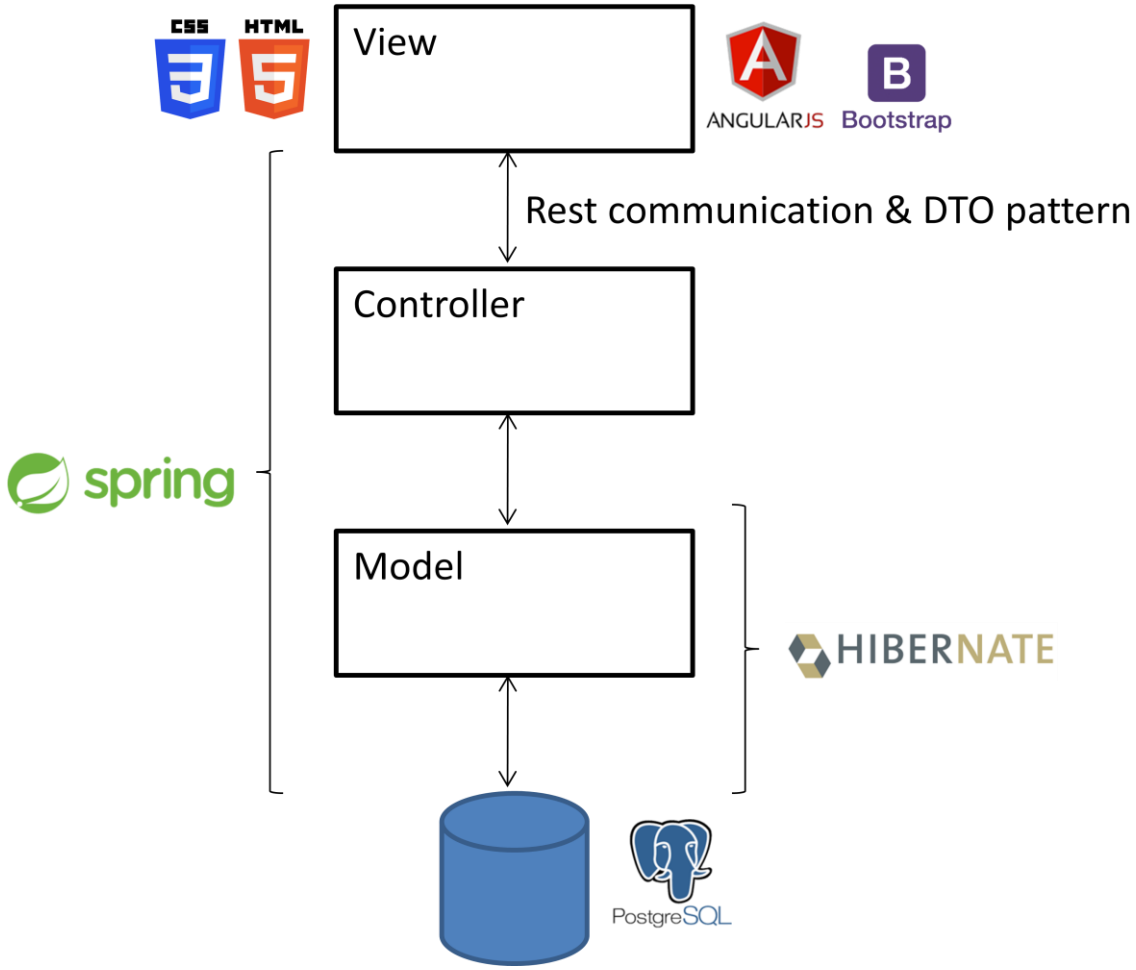

Figure 10 – LabControl's architecture

To implement this architecture, it was used HTML5, CSS3, Bootstrap and AngularJS to create the view. This layer receives and sends information to the

controller through a RESTful web service using Data Transfer Objects (DTOs) in JavaScript Object Notation (JSON) format. The controller receives the information from the model and passes it to the view by the RESTful web service, built using Spring Framework. The model implements database access through Hibernate's object-relational mapping; it handles the mapped entities together with its associations, restrictions and behavior. In the whole system Spring Framework is used for dependency injection and inversion of control; Spring's configuration is done by using Spring Boot, and the management of the project dependencies is done by Maven.

It can be considered that parts of code regarding the functionalities of the system are spread in the three layers. The functionalities can be generally divided into three categories: general, NGS and biologic collection functionalities. The general one refers to user management and security; NGS refers to all the functionalities regarding NGS data management; and biological collection addresses all the functionalities related to the collection. A visual representation of those in the architecture can be seen in Figure 11.

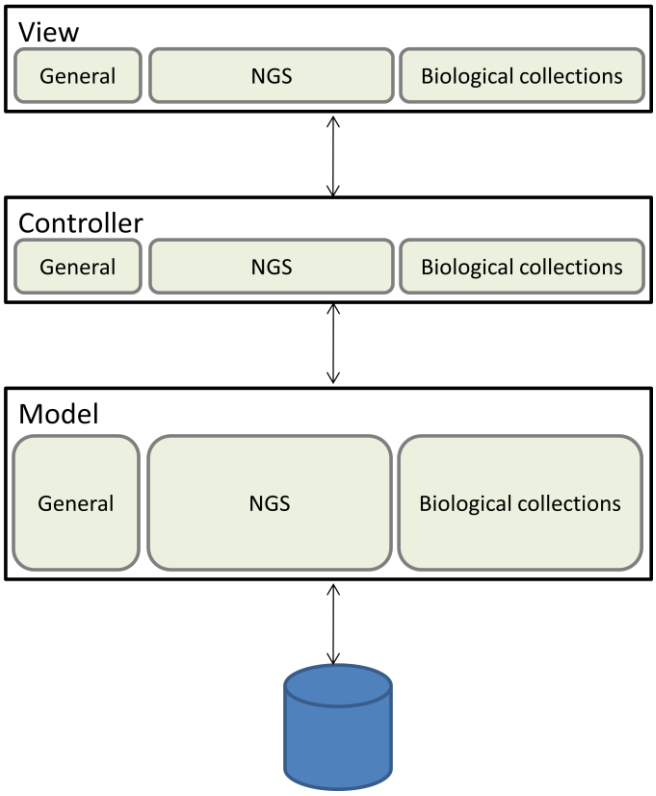

Figure 11 – A visualization of LabControl's functionalities in the layers of the system

All the general and biological collections functionalities (except by reports) are implemented. The code regarding NGS functionalities is fully implemented in the model layer and in the database. LabControl implementation and documentation will be publically available on author's GitHub (https://github.com/marianaparise/labcontrol).

As results of interface implementation, some images showing the interface can be seen in Figures 12, 13 and 14. Figure 12 shows the login interface, while Figure 13 presents the screen used to register a new strain. In Figure 14, a vision of a freezer is shown.

Figure 12 – LabControl login interface

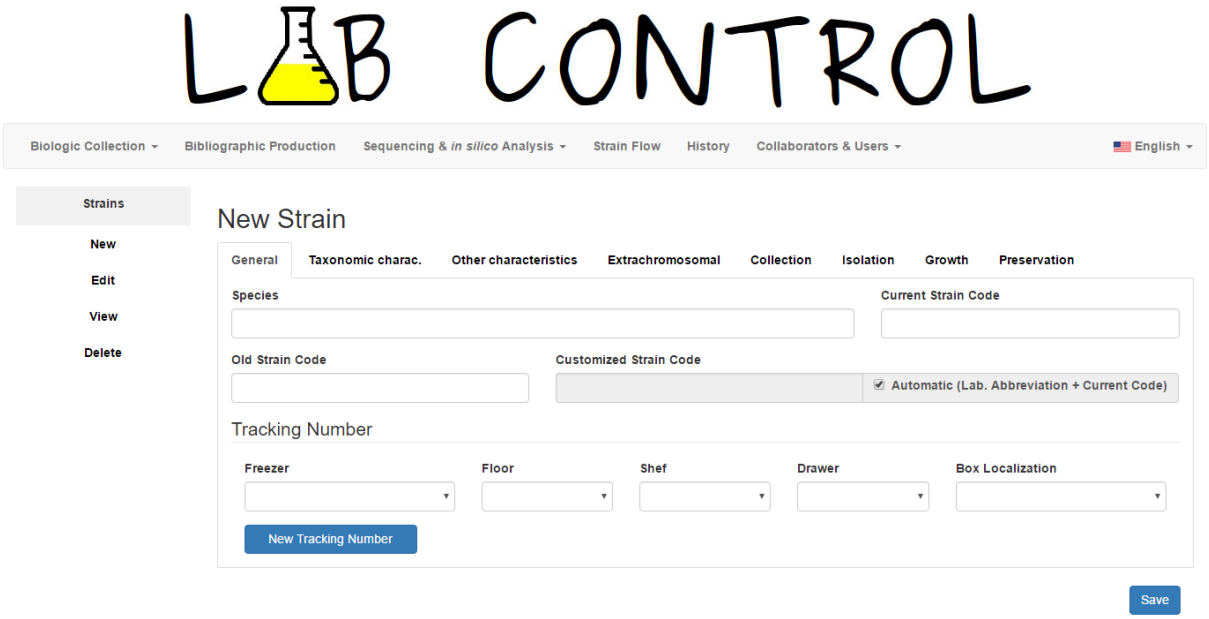

Figure 13 – Screen used to register a strain in LabControl showing strain's general information

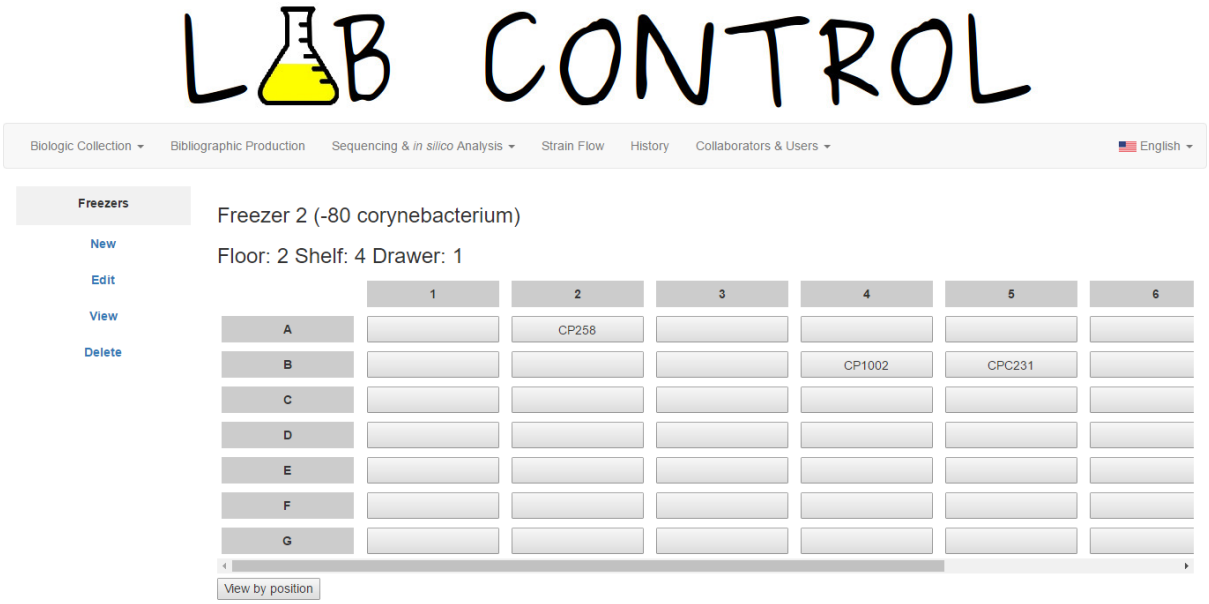

Figure 14 – LabControl's visualization of the freezer

Additional figures showing the interface of the system are shown in the A APPENDIX II.

5.4. Software registration

LabControl software was registered in the National Institute of Industrial Property (INPI) of Brazil. The certificate of registration was granted to the authors on June 7th, 2016 as shown in the APPENDIX III.

6. Discussion

6.1. Data model evaluation

The data model presented in the previous section was created based on the literature, interviews, observations of researchers' routine, and analysis of the methodology used by those researchers to manage laboratory information (spreadsheets). This approach confers to LabControl more proximity to the reality of the laboratories, considering that the data model was designed and re-designed 15 times until it was well accepted by the researchers, and considered comprehensive enough as well as in accordance with the literature. The interviews and observation approach allowed needs to be understood in a practical way and the software to be designed to facilitate not only management, but also to improve researchers' everyday life.

As it was stated by Aller and Salazar (2016), systems based only on theoretical specifications do not meet microbiology laboratories' needs. Modeling a system according to the users' needs and receiving constant evaluation of them is part of the agile methodologies of software engineering. It contributes to build a more reliable and useful system, which will need less modifications to thoroughly fit the scientists' management necessities.

Besides the effort done with the interviews and observation, the resulting data model is based on the literature and international data patterns; it makes the software more useful and reliable. All the biological information to be stored through this model was confirmed or designed according to the literature, what ensures that no biologically useless information is kept. In addition to that, the model was designed according to WFCC, SBM and GSC guidelines. This foments international standardization into the laboratories and facilitates future comparisons both to internal and external data.

As it was explained, this data model can be considered reliable, close to the reality and its usage can facilitate both researchers' everyday life and the creation of new articles, thesis or dissertations due to an organized and standardized source of information.

6.2. LabControl evaluation

In this section, the produced software will be evaluated according to the features considered important to microbiological LIMS that deals with NGS and samples, which were presented in Table 2 (section 1.2.3), and compared with other systems as well. The discussed features can be seen as follows:

(a) Open-source code and documentation: Open-source code is considered indispensable to a LIMS designed to academic laboratories, and allows modifications from other laboratories to meet their specific needs (LIST *et. al.*, 2014). Considering that, LabControl was developed as an open-source LIMS to support microbiology academic laboratories without maintenance or license costs, and with the possibility of changing the software according to additional desired functionalities. To allow developers to understand the code, make the code reusable and lower maintenance effort and cost, LabControl's documentation will be available together with the code on GitHub[11]. Making documentation available is a good practice of scientific computing (WILSON *et al.*, 2014) and is considered essential do LIMS software (LIST et. al., 2014).

(b) Standard operation procedures support: As stated by Vitt (1992), following SOPs is an essential good laboratory practice and it also increases the competitiveness of the laboratory (BLAZEK *et al.*, 2015). Taking it into account, this feature is supported by LabControl. The system allows users to register SOPs and classify those as *in vitro*, *in vivo* and *in silico* experiments. Furthermore, SOPs can be related to the collection, isolation, growth and preservation of the samples. This approach allows studies to be reproducible, which is essential to science, as well as laboratories to standardize procedures and make them available in the system. Having the SOPs available in a centralized database lowers duplicated SOPs, and helps researchers to find the used procedures without asking or digging into files. Moreover, LabControl stores the applied techniques and experiments (SOP usage), which helps in the general management of the laboratory preventing

[11] https://github.com/marianaparise/labcontrol

experiments to be repeated without need (spending unnecessary money) and in the design of new experiments, as it was found in the interviews. Additionally, the storage of experiments results is extremely important to the laboratories; this keeps them available to be found by the researchers and avoids results loss. Rhoads (2014) discussed that results are really difficult to be kept in a data structure, considering that those can be numerical, texts, images or even have all of them together (RHOADS *et al*., 2014). To solve this issue, in LabControl results are stored as files. This approach allows laboratories to standardize their files and use a results model to each kind of experiment, considering that different experiments require different ways to present the findings. It provides freedom to the laboratories together with a way to standardize and have all the results and SOPs in an easily accessible repository.

(c) Sample management: LabControl provides a comprehensive samples metadata storage, which embraces collection, isolation, growth, preservation, taxonomic characteristics and other characteristics. During interviews and observation steps, it was found that the possibility of adding other characteristics to the strains is extremely important, considering the myriad of organisms and their specific features. Besides, characteristics found by techniques appliance or regulatory conditions can be stored without limitations or need of modifications in the software. It gives freedom to the researchers to add any important feature and recover it without effort. Another advantage is that once it is found, the source can be registered in the description of the feature and, also, attached to the related bibliography. Another important issue about samples management is concerning their names in other databases or collections. Romano *et al*. (2005) shows that problems related to strain names can arise when integrating databases or comparing the same strain in different collections. To provide an efficient identification of the strains, LabControl creates a unique identifier to each strain using the collection abbreviation plus a numeric identifier and enables users to store other collections' names. It solves the problem of strain names to integrate databases by allowing samples to be associated with multiple collections'

names. Additionally, it facilitates future comparisons among collections and saves researchers efforts to that purpose. In addition to that, multi-derivate tracking (RHOADS *et al*., 2014) is supported by this system. It allows researchers to link one sample to another, for example, when one derivates from the other such as in mutation studies. When there is just one derivation, it can be recorded in the "old strain code" feature, and when the sample comes from more derivations it is possible to store this as an additional characteristic of the sample. Considering taxonomic nomenclature, LabControl suggests the already used names to help in the nomenclature standardization. This can be a big problem if each user registers *Corynebacterium pseudotuberculosis* in a different way (*C. pseudotuberculosis*, *Coryne pseudotuberculosis*, *C. pseudo*, etc.) because the system will not be able to recognize all of them as the same organism. This problem was clearly noticed when observing researchers' routine and in the interviews, and It was also reported by RHOADS *et al*. (2014). To solve that, LabControl always suggests the already stored possibilities and allows the user to create a new option if it is necessary. The presence of biological collection metadata together with sequence data and metadata, as it is presented by LabControl, allows more comprehensive analysis to be done (ROMANO *et. al*., 2005; KRYPEDES *et al*., 2014; CASAREGOLA *et al*., 2016). Features such as pathogenicity and the health conditions of the host can contribute to the sequence analysis, and the techniques used in assembly and annotation can explain some bias in the final DNA sequence.

(d) Sample tracking: LabControl presents sample tracking by a tracking number and a visualization of the used freezers. Those can be registered with the internal characteristics (number of floors, shelves, drawers and boxes), which gives a flexibility to store information about different models of freezers without changing the software. It is known that biological collections can be physically stored in places other than freezers; the implementation of this possibility can be hereafter done. This approach was chosen to be firstly implemented (based on the interviews) because it is a really common approach and it was the necessity of the host laboratory. Nevertheless, any

structure that presents floors, shelves, drawers and boxes can be represented by the system in a general way. The presence of a sample tracking number shows the localization of a sample by number and facilitates to find samples in the freezer. Tracking samples is an essential feature of a LIMS designed to deal with biological collections (GRIMES & HANLEE, 2014; LIST *et al.*, 2014), and helps the researchers to organize and find the samples, saving time and effort.

(e) Sample receipt and sending (flow) support: LabControl manages samples coming in or out of the laboratory, which is considered essential (WFCC, 2010; SEPULVEDA & YOUNG, 2013). If it is not controlled, samples can arrive in the laboratory and not be analyzed or, worse, they can be lost; when going to other labs, the same scenery can occur, and it can cause damage to both laboratories involved (*e.g.* financial losses or the loss of a unique sample which cannot be recovered).

(f) Sample collections international guidelines and genomic data patterns: Considering the bibliographic research done to this work, no open-source microbiology LIMS focused on NGS and samples uses data patterns. In LabControl's requirements elicitation phase, WFCC, SBM and GSC were considered to create a data model according to the patterns. The accordance with these patterns increases the usability of the generated data, facilitating further studies and reducing effort to deposit data on public databases (BROOKSBANK and QUACKENNBUSH, 2006; FIELD *et al*., 2008). The idea of patterns in the LIMS turns the laboratories into standardized ones and supports the construction of a standardized genomic community. As an effort to the standardization, LabControl supports this idea through an easy and user-friendly way to store, manage and easily recover standardized information about microbial strains and NGS data.

(g) Bibliography support: LabControl meets the requirement of support bibliography related to the stored samples. Besides the importance stated in the literature (PRASAD & BODHE, 2012; TAGGER, 2011; CASAREGOLA *et*

al., 2016), during the interviews it was noticed that keeping the produced bibliography linked to the strains can help in the production of new works by easily recovering the produced or relevant bibliography about a strain of interest. This approach can save scientists great amounts of effort; those can, instead of digging into public repositories or asking other researchers to send their works, just search in the software and recover the bibliography about the strain.

(h) Web-based software: The evaluated software meets the requirement of being a web-enabled system, allowing information to be available in the click of a button, and relevant data to be readily shared with other scientists, facilitating information access and collaboration among researchers. It can hasten studies results, considering that the time of the researcher will be better employed than looking for information and trying to share it with other scientists. Saving efforts from researchers make them more productive and free to focus on more interesting things than how to share or carry information (QUO *et al.*, 2005; DUBEY *et al.*, 2012; LIST *et al.*, 2014). Also, it is well known and confirmed by the interviews that researchers study both in and out of the laboratory, so having information available with no effort can facilitate their lives.

(i) Genomic data support: As it was explained by Grimes & Hanlee (2014) and Perez-Arriaga (2015), genomic data can be difficult to manage without information systems. LabControl supports general genomic workflow (sequencing, assembly, annotation and submission to public databases) by a comprehensive storage and recover interface, and other *in silico* analysis can be stored as SOP usage. As it can be seen in the literature (EDWARDS & HOLT, 2013; YANDELL & ENCE, 2012) as well as in interviews and observation, annotation is a decisive, yet sometimes underestimated process to genome analysis. The metadata regarding this process is very important to further comparative genomic studies, which normally uses the annotated genes to compare genomes. The software used to annotation as well as the databases can influence in the results, and knowing what was used can solve

some issues or give the path to new annotations. Besides that, it is common bioinformaticians standardize the annotations before starting comparative genomic studies, as it was done by (COSTA, 2015). The knowledge of the annotation metadata allows the researchers to not lose the information about how a genome was annotated and avoid unnecessary harmonization, as well as guide future annotations and standardizations. The stored metadata about annotation can address all these issues. Besides, Field et al. (2005) states that the availability of genomics metadata allows researchers to analyze a richer set of information and take most comprehensive conclusions, considering that genomic features can be explained, associated or interpreted based on the referred metadata.

(j) Reports support: Through the interviews and observation, it was noticed that reports represent a way to extract information out of the system and, consequently, share it or use it in a different system as well as out of the computer; those are considered essential to LIMS (LIST *et al.*, 2014; ALLER & SALAZAR, 2016). LabControl meets the requirement of reports support by giving the possibility to extract the queries done in the system as PDFs or excel files. It gives freedom for the users to create their own reports based on the system's queries, but also limits the user to them. Even without total freedom to create reports, this approach is far better than just having some ready models that cannot be changed or adapted to the users' necessities. This feature can be easily enhanced if it is noticed that more customized reports are required.

(k) Security: Security is one of the major concerns when building information systems to be used in microbiological laboratories, as well as to any laboratory (SEPULVEDA *et al.*, 2013; RHOADS *et al.*, 2014; BLAZEK *et al.*, 2015). In accordance with this, LabControl was developed using Spring Security framework to guarantee the authentication of users and the authorized access to the system resources. This framework has been used by thousands of projects around the world, including governmental and military systems. The major advantage of using a well known and recognized security

framework is that it has been used for years in the most diverse projects. This is far more secure than building new security solutions that may be error-prone and would be tested in this software by the first time ever. In addition, a community of expert developers is involved in the solution of any problem that may appear in the framework (LUCKOW & MELO, 2010; SERRANO *et al.*, 2015). LabControl's security is based on three user types: administrator, researcher and guest. This approach allows data to be only deleted by administrators, who are supposed to be senior researches with a good notion of responsibility and comprehension of the research. To guest users, it is only allowed to visualize data. These roles are generic and can be applied to any microbiology laboratory; if there is need to different roles, it can be easily added in the system. The used approach meets the security requirements stated in the introduction of this work, and due to the usage of Spring Security. it is easily adaptable to new roles. Besides, Spring's access control lists can be used to a more customized role definition if it is hereafter required.

(l) Audit-logging: LabControl maintains a history about all modifications in the database associated with the user who did the modification, and in case of deletions, activations and deactivations the reason of the action is stored as well. This approach is important for information to not be lost and for mistakes to be found (LIST *et al.*, 2014; GRIMES & HANLEE, 2014). In the history, all the changes in the database can be seen; it allows researchers to track modifications in the data in case of wrongly changed information or, even, if there is a suspicion of data manipulation. This approach enhances the security in case of system invasion or malicious use of some username and password because the changes are going to be safe and it can be hereafter analyzed.

In addition to the stated features, another important feature of the system is the ability to store collaborators. Through the observation and interviews that were done, it was noticed that storing people who participate in the activities is also important. We noticed that if it is not stored, the one who applied the techniques can be easily forgotten or mistaken. It is a problem if the laboratory history is wrong, and

it also complicates when publishing a study (authors can be forgotten or mistaken). Additionally, when it comes to reproduction of the studies, it is important to easily solve questions about the experiment. To solve this issue, the system was implemented with collaborators registered in almost every modeled process (samples isolation, collection, identification, deposit, sequencing, assembly, annotation, submission and every applied *in silico*, *in vitro* and *in vivo* technique). Besides, it is possible to address the responsible one in each previously named process.

Considering what was showed in this section, LabControl has been developed according to the important features to a microbiological LIMS. A comparison with open-source LIMS systems that could be used by microbiological laboratories which handle samples and NGS data is shown in Table 5. Among the compared systems, LabControl is the unique software that supports all the features, contrasting the others. Thus, this software gathers all the features and facilitates researchers' life through a unique LIMS, which makes all information available in one place.

Table 5 – LIMS comparison.

References: BikaLIMS[12], openBIS ELN-LIMS (BARILLARI *et al*., 2016), Omics metadata management software (PEREZ-ARRAIGA *et al.*, 2015)

	BikaLIMS	LabControl	openBIS ELN-LIMS	Omics metadata management software
Open-source code	✔	✔	✔	✔
Documentation	✔	✔	✔	✘
Standard operation procedures support	✘	✔	✔	✘
Sample management	✔	✔	✔	✔
Sample tracking	✔	✔	✘	✔
Sample receipt and sending (flow) support	✘	✔	✘	✘
Sample collections international guidelines	✘	✔	✘	✘
Bibliography support	✔	✔	✔	✘
Web-based	✔	✔	✔	✔
Genomic data support	✘	✔	✔	✔
Genomic data patterns	✘	✔	✘	✘
Reports support	✔	✔	✔	✔
Security	✔	✔	✔	✔
Audit-logging	✔	✔	✔	✘

Symbols meanings:

✘ - not informed ✘ - not implemented ✔ - fully implemented

Besides the previously presented advantages, some features considered important by other authors are not addressed in this software, such as data analysis, compatibility with mobile devices, barcodes usage when tracking samples (LIST *et al.*, 2014), and financial management (SEPULVEDA & YOUNG, 2013). The lack of these features does not reduce LabControl's usefulness and good impact to the researchers that are going to use it. Indeed, the implementation of these features can improve the system and benefit the users; those can be hereafter added. The fact that the system is open-source, combined with good documentation and an easy-to-modify architecture allows those features or others to be easily added by any developer with enough knowledge in the used technologies.

[12] https://www.bikalims.org/

Another important feature to address in the system is an easy-to-use interface (PRASAD & BODHE, 2012; SEPULVEDA & YOUNG, 2013; RHOADS *et al.*, 2014). As it is shown in LabControl's interface to add new collaborators (Figure 15), the system has a simple and user-friendly interface; it has been designed to be easy and to not tire user's mind with too much information. This approach facilitates researchers' life, considering that using the system is not one more problem to be solved, but a solution to at least part of the management problems in the laboratory.

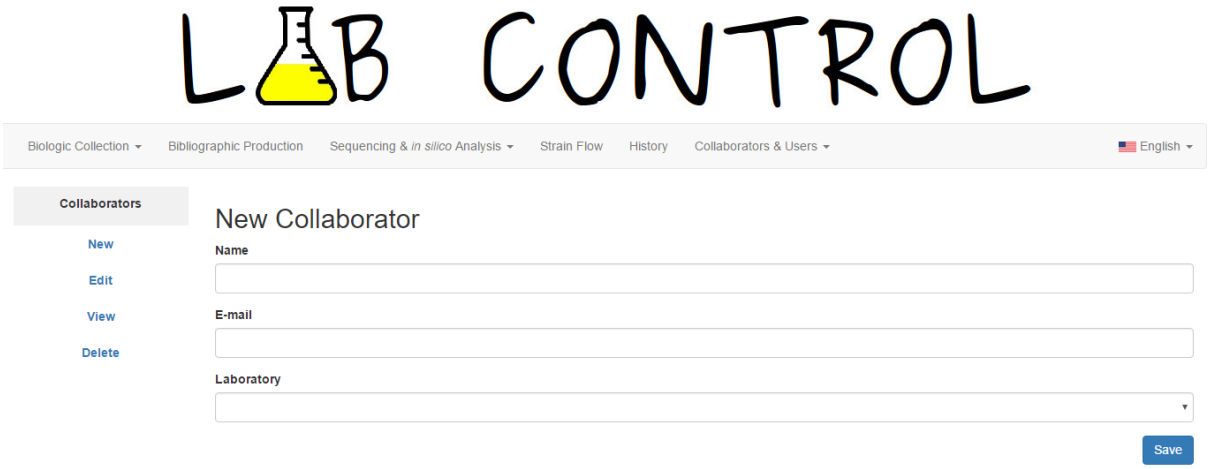

Figure 15 – LabControl's interface to register a new collaborator

6.3. Systems' architecture and implementation evaluation

LabControl was developed with modern and reliable technologies, which are developed and maintained by the experts in their area: Spring Framework, Spring Security, AngularJS, Hibernate, Boot Strap and Spring Boot (SERRANO *et al.*, 2015; RADFORD, 2015; WALLS, 2016; FISHER & MURPHY, 2016). Furthermore, it is based on MVC architectural pattern, which provides low coupling between the layers and is widely used in web development. Low coupling is considered a good programming practice that enhances code maintainability (LUCKOW & MELO, 2010; SOMMERVILLE, 2011), which is an important feature to LIMS. In these systems, requirements can easily change due to science advances, demanding software maintenance, and the system can be used by other laboratories which have different

needs and would add new functionalities to it. The possibility of easily changing the database or the way that the data is presented without ruining the entire system is highly desirable and implemented in LabControl; flexibility in database implementation is considered important by List *et al.* (2014).

The usage of RESTful communication between view and controller layers provides decoupling between those; this allows the implementation of one layer to be changed without changing the other layer. Additionally, the RESTful service allows other visualization forms to be easily added (SOMMERVILLE, 2011; SAUDATE, 2013), for example just creating a mobile or tablet app able to communicate with the service. Hibernate usage to object-relational mapping confers to the system the possibility of changing databases with just configuration adjustments in the model layer and no modification in the other layers, which is a great advantage considering easy maintenance and low coupling.

Furthermore, the usage of an architectural pattern facilitates the comprehension of the code (SOMMERVILLE, 2011), considering that it is organized according to a well known pattern and, if the developer is aware of the pattern, it is easy to understand the layers, their purpose and how they interact. Frameworks usage provides the same facility to developers, who understanding the framework would understand the software. To frameworks, documentation and examples of use are abundant in the internet and books.

Another relevant feature of LabControl's architecture is to be platform independent, which is considered important to LIMS by Prasad and Bodhe (2012). This feature confers more possibilities to the system to be used and attend the operation systems requirements of the laboratories; in academic environment, it is not feasible to change the server's operating system to support a given software.

Considering LabControl's implementation, a relevant feature is that it has been done incrementally, which is part of the agile methodologies of software process. According to Wilson *et al.* (2014), making incremental changes in the software is considered a good practice to scientific software development, which increases the productivity and facilitates error correction.

In a general vision, LabControl's architecture and implementation are based on software development, LIMS development and scientific computing good practices (SOMMERVILLE, 2011; PRASAD & BODHE, 2012; WILSON *et al.*, 2014;

LIST *et al.*, 2014; SERRANO *et al.*, 2015; RADFORD, 2015; WALLS, 2016; FISHER & MURPHY, 2016). Easy maintainability and extension of the system are provided by MVC and frameworks usage, which is considered appropriate to LIMS due to the fact that system's adaptation is a difficult and time consuming task that can be facilitated by the designed architecture. Furthermore, frameworks usage is a benefit to LIMS systems due to its reliability and developmental community support (SOMMERVILLE, 2011; WEISSMANN, 2014; LIST *et al*., 2014).

7. CONCLUSIONS AND PERSPECTIVES

Nowadays, management of NGS and biological collections data is an important concern due to the huge amounts of data that must be handled, causing financial and organizational damages to academic laboratories that are not able to afford a commercial solution to this issue. LabControl was created to support microbiological laboratories which deal with NGS and samples through a comprehensive and easy-to-use web system that was engineered based on researcher's opinions, literature and international data patterns. Besides, it has been built upon a modern and reliable architecture together with a good requirements engineering process. It conferred an important advantage to LabControl, considering that according to Prasad and Bodhe (2012) many LIMS fail due to an incorrect vision when implementing the system.

Although LabControl is specialized in microbiological laboratories which deal with samples and NGS data, it can be used by any laboratory due to its possibility to handle any type of *in silico*, *in vivo* and *in vitro* experiments. This fact expands LabControl's potential users and allows it to support more researchers or laboratories.

As a perspective, LabControl's implementation has to be finished. As it was explained in the section 5.2 of this work, the functionalities of the system can be divided into general, NGS and biologic collection functionalities; the general and biologic collection ones are implemented, except by reports, and the NGS ones are going to be implemented.

After implementation, tests are going to be run. LabControl will be tested by the host laboratory (LGCM) and released to be tested by other laboratories as well. This approach will ensure that the implemented features can completely support microbiology laboratories that deal with samples collections and NGS data. As a result of testing, new features can be added or old features can be modified in accordance with the researchers' feedback.

Considering that LabControl's maintenance was a concern when developing the software, it was designed to be easily modified (see 6.6 System's architecture evaluation). This facilitates modifications or adaptations, not only benefiting system

testing and adaptations, but also giving the possibility for other laboratories to easily modify the software to meet their specific needs.

8. BIBLIOGRAPHY

ABURJAILE, FF *et al*. Genomics. In: Zahoorullah S MD. (Org.). A Textbook of Biotechnology. 1ed.: **SM Online Publishers** LLC, 2015, v. , p. 1-19.

ALLER, Raymond D.; SALAZAR, Vincent. Microbiology Laboratory Information Systems. Manual of Commercial Methods in Clinical Microbiology: International Edition, Second Edition, p. 377-385, 2016.

ATCC. IN: < http://www.atcc.org > Accessed in: May 31, 2016.

ASHBURNER, Michael *et al*. Gene Ontology: tool for the unification of biology. **Nature genetics**, v. 25, n. 1, p. 25-29, 2000.

BARRETT, Tanya et al. NCBI GEO: mining tens of millions of expression profiles—database and tools update. **Nucleic acids research**, v. 35, n. suppl 1, p. D760-D765, 2007.

BARH, Debmalya; BLUM, Kenneth; MADIGAN, Margaret A. (Ed.). **OMICS: biomedical perspectives and applications.** CRC Press, 2012.

BARH, Debmalya; ZAMBARE, Vasudeo; AZEVEDO, Vasco (Ed.). **Omics: applications in biomedical, agricultural, and environmental sciences.** CRC Press, 2013.

BARILLARI, Caterina *et al*. openBIS ELN-LIMS: an open-source database for academic laboratories. **Bioinformatics**, v. 32, n. 4, p. 638-640, 2016.

BLAZEK, Pavel *et al*. Development of Information and Management System for Laboratory Based on Open Source Licensed Software. In: **Computational Collective Intelligence**. Springer International Publishing, 2015. p. 377-387.

BROOKSBANK, Cath; QUACKENBUSH, John. Data standards: a call to action. **Omics: a journal of integrative biology**, v. 10, n. 2, p. 94-99, 2006.

CABRI. In: <http://www.cabri.org/> Accessed in: July 27, 2016A.

CABRI. In: <http://www.cabri.org/guidelines.html> Accessed in: September 23, 2016B.

CASAREGOLA, Serge *et al*. An Information System for European culture collections: the way forward. **SpringerPlus**, v. 5, n. 1, p. 1-11, 2016.

CBAS. In: <http://cbas.fiocruz.br/index> Accessed in: September 12, 2016.

CORREIA, Angela de Fatima Kanesaki. **Implementação de um sistema de qualidade para laboratório de análise sensorial baseado no sistema de boas práticas.** Master's dissertation. Escola Superior de Agricultura "Luiz de Queiroz", Universidade de São Paulo, Piracicaba (SP), 2005.

COSTA, Daniela Arruda *et al*. Genômica comparativa de leveduras de interesse biotecnológico. Doctorate thesis. 2015.

ÇAĞINDI, Özlem; ÖTLEŞ, Semih. Importance of laboratory information management systems (LIMS) software for food processing factories. **Journal of food engineering,** v. 65, n. 4, p. 565-568, 2004.

DUBEY, Sunil K.; ANAND, Amit; JANGALA, Hemanth. Laboratory information and management system: A tool to increase laboratory productivity. **Clinical Research and Regulatory Affairs**, v. 29, n. 2, p. 46-56, 2012.

EDWARDS, David J.; HOLT, Kathryn E. Beginner's guide to comparative bacterial genome analysis using next-generation sequence data. **Microbial informatics and experimentation**, v. 3, n. 1, p. 1, 2013.

FIEHN, Oliver. Metabolomics–the link between genotypes and phenotypes. **Plant molecular biology**, v. 48, n. 1-2, p. 155-171, 2002.

FIELD, Dawn; FEIL, Edward J.; WILSON, Gareth A. Databases and software for the comparison of prokaryotic genomes. **Microbiology**, v. 151, n. 7, p. 2125-2132, 2005.

FIELD, Dawn *et al*. The minimum information about a genome sequence (MIGS) specification. **Nature biotechnology**, v. 26, n. 5, p. 541-547, 2008.

FIELD, Dawn *et al*. The genomic standards consortium. **PLoS Biol**, v. 9, n. 6, p. e1001088, 2011.

FIELD, Dawn; SANSONE, Susanna-Assunta. A special issue on data standards. **OMICS: a Journal of integrative Biology**, v. 10, n. 2, p. 84-93, 2006.

FISHER, Paul; MURPHY, Brian D. Architecting Your Application with Spring, Hibernate, and Patterns. In: **Spring Persistence with Hibernate**. Apress, 2016. p. 1-16.

FLEISCHMANN, Robert D. *et al*. Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. **Science**, v. 269, n. 5223, p. 496, 1995.

GARRITY, George *et al*. Standards in Genomic Sciences. **Standards in Genomic Sciences**. 2009;1(1):1-2. doi:10.4056/sigs.34251.

GERHARDT, Philipp. Methods for General and Molecular Bacteriology. 2ª edition. **Ed: American Society for Microbiology**, p. 143-147, 1994.

GRIMES, Susan M.; HANLEE, P Ji. MendeLIMS: a web-based laboratory information management system for clinical genome sequencing. **BMC bioinformatics**, v. 15, n. 1, p. 1, 2014.

GUEDES, Maria T. *et al*. Infecção por Corynebacterium pseudotuberculosis em equinos: aspectos microbiológicos, clínicos e preventivos. **Pesq. Vet. Bras**, v. 35, n. 8, p. 701-708, 2015.

HENRICKS, Walter H. Laboratory Information Systems. **Clinics in Laboratory Medicine**, Volume 36, Issue 1, Pages 1-11, 2016.

HEWITT, Robert; WATSON, Peter. Defining biobank. **Biopreservation and biobanking**, v. 11, n. 5, p. 309-315, 2013.

INSTITUTE PASTEUR. In: <https://research.pasteur.fr/en/team/biological-resources-center/ > Accessed in: May 19, 2016.

KAMMERGRUBER, Roland *et al*. The future of the laboratory information system–what are the requirements for a powerful system for a laboratory data management?. **Clinical Chemistry and Laboratory Medicine**, v. 52, n. 11, p. e225-e230, 2014.

KELLNER, Roland. Proteomics. Concepts and perspectives. **Fresenius' journal of analytical chemistry**, v. 366, n. 6-7, p. 517-524, 2000.

KRASNER, Glenn E.; POPE, Stephen T.; A description of the model-view-controller user interface paradigm in the smalltalk-80 system. Journal of object oriented programming, v. 1, n. 3, p. 26-49, 1988.

KYRPIDES, Nikos C. *et al*. Genomic encyclopedia of bacteria and archaea: sequencing a myriad of type strains. **PLoS Biol**, v. 12, n. 8, p. e1001920, 2014.

LANDER, Eric S. *et al*. Initial sequencing and analysis of the human genome. **Nature**, v. 409, n. 6822, p. 860-921, 2001.LINARES, Daniel M.; KOK, Jan; POOLMAN, Bert. Genome sequences of Lactococcus lactis MG1363 (revised) and NZ9000 and comparative physiological studies. **Journal of bacteriology**, v. 192, n. 21, p. 5806-5812, 2010.

LIST, Markus *et al*. Efficient Sample Tracking With OpenLabFramework. **Scientific reports**, v. 4, 2014.

LIST, Markus *et al*. Efficient Management of High-Throughput Screening Libraries with SAVANAH. **Journal of Biomolecular Screening**, p. 1087057116673607, 2016.

LUCKOW, Décio Heinzelmann; MELO, Alexandre Altair. Programação Java para a WEB. **Novatec Editora**, 2010.

MADIGAN, Michael T. *et al*. Microbiologia de brock. 12ª edição. **Artmed Editora**, 2010.

MADIGAN, Michael T. *et al*. Brock Biology of Microorganisms. 14ª edition. **Pearson Editora**, 2015.

MARIANO, Diego César Batista**. SIMBA: uma ferramenta Web para gerenciamento de montagens de genomas bacterianos.** Dissetação de mestrado. Programa Interunidades de Pós-Graduação em Bioinformática, Universidade Federal de Minas Gerais, Belo Horizonte (MG), 2015.

MARSHALL, R. B.; WILTON, B. E.; ROBINSON, A. J. Identification of Leptospira serovars by restriction-endonuclease analysis. **Journal of medical microbiology**, v. 14, n. 1, p. 163-166, 1981.

MCGEOCH, Adam T.; BELL, Stephen D. Extra-chromosomal elements and the evolution of cellular DNA replication machineries. **Nature Reviews Molecular Cell Biology**, v. 9, n. 7, p. 569-574, 2008.

MCGETTIGAN, Paul A. Transcriptomics in the RNA-seq era. **Current opinion in chemical biology**, v. 17, n. 1, p. 4-11, 2013.

MCKUSICK, Victor A.; RUDDLE, Frank H. A new discipline, a new name, a new journal. **Genomics**, v. 1, n. 1, p. 1-2, 1987.

MIXS PROJECT. In: <http://gensc.org/projects/mixs-gsc-project/> Accessed in: September 12, 2016.

MOREIRA, Leandro M. *et al*. Ciências genômicas: fundamentos e aplicações. 1ª edição. **Sociedade Brasileira de Genética**, 2015.

OEDC. OEDC Best Practices Guidelines for BRCs. 2007

PEREZ-ARRIAGA, Martha O. *et al*. Omics Metadata Management Software (OMMS). **Bioinformation**, v. 11, n. 4, p. 165, 2015.

PIERCE, Benjamin A. Genética: um enfoque conceitual. 3ª edição. **Guanabara Koogan**, 2011.

PORTAL FIOCRUZ. In: <http://portal.fiocruz.br/pt-br/content/cole%C3%A7%C3%B5es-biol%C3%B3gicas>. Accessed in: Maio 16, 2016.

PORTER, J. R. Antony van Leeuwenhoek: tercentenary of his discovery of bacteria. Bacteriological reviews, v. 40, n. 2, p. 260, 1976.

PRASAD, Poonam J.; BODHE, G. L. Trends in laboratory information management system. **Chemometrics and Intelligent Laboratory Systems**, v. 118, p. 187-192, 2012.

PRESSMAN, Roger S. Engenharia de software: uma abordagem profissional. 7ª edição. **Ed: McGraw Hill**, 2011.

QUO, Chang F.; WU, B.; WANG, May D. Development of a laboratory information system for cancer collaboration projects. In: **27th Annual International Conference of the Engineering in Medicine and Biology Societ.** 2005.

REDDY, T. B. K. *et al*. The Genomes OnLine Database (GOLD) v. 5: a metadata management system based on a four level (meta) genome project classification. **Nucleic acids research**, p. gku950, 2014.

RADFORD, Stephen. Learning Web Development with Bootstrap and AngularJS. **Packt Publishing Ltd**, 2015.

RHOADS, Daniel D. *et al*. Clinical microbiology informatics. **Clinical microbiology reviews**, v. 27, n. 4, p. 1025-1047, 2014.

ROMANO, Paolo *et al*. Improving interoperability between microbial information and sequence databases. **BMC bioinformatics**, v. 6, n. 4, p. 1, 2005.

SANGER, Frederick; NICKLEN, Steven; COULSON, Alan R. DNA sequencing with chain-terminating inhibitors. Proceedings of the National Academy of Sciences, v. 74, n. 12, p. 5463-5467, 1977.

SAUDATE, Alexandre. REST: Construa API's inteligentes de maneira simples. **Casa do Código**, 2013. 294p.

SEPULVEDA, Jorge L.; YOUNG, Donald S. The ideal laboratory information system. **Archives of Pathology and Laboratory Medicine**, v. 137, n. 8, p. 1129-1140, 2013.

SERRANO, Maurício; SERRANO, Milene; CAVALCANTE, André Cruz. Reference Software Architecture for Government Information Systems. In: **Proceedings of the annual conference on Brazilian Symposium on Information Systems: Information Systems: A Computer Socio-Technical Perspective-Volume 1**. Brazilian Computer Society, 2015. p. 11.

SETTE, Lara D. *et al*. Recomendações para operação e gerenciamento de coleções de culturas de microrganismos. **Microbiol. Foco.(Brazilian Society of Microbiology)**, v. 2, p. 49-55, 2007.

SICOL CBAS. In: <http://splink.cria.org.br/manager/detail?system=sicol&resource=CBMAI> Accessed in: September 14, 2016.

SICOL CBMAI. In: <http://splink.cria.org.br/manager/detail?system=sicol&resource=CBMAI> Accessed in: September 14, 2016.

SILVEIRA, Paulo *et al*. Introdução à arquitetura e design de software: uma visão sobre a plataforma Java. **Elsevier – Campus**. 2012.

SIMS, David *et al*. Sequencing depth and coverage: key considerations in genomic analyses. **Nature Reviews Genetics**, v. 15, n. 2, p. 121-132, 2014.

SOMMERVILLE, Ian. Software engineering, 9ª edition. **Pearson Addison-Wesley**, 2011.

TAGGER, Ben. An introduction and guide to successfully implementing a LIMS (Laboratory Information Management System). **Computer Science Department, University of Wales**, 2011.

THE C. ELEGANS SEQUENCING CONSORTIUM. Genome Sequence of the Nematode C. elegans: A Platform for Investigating Biology. **Science**. Vol. 282, Issue 5396, pp. 2012-2018, 1998.

TORTORA, Gerard J. *et al.* Microbiologia. 10ª edição. **Artmed Editora**, 2012.

VARMUS, Harold. Genomic empowerment: the importance of public databases. **Nature genetics**, v. 32, p. 3-3, 2002.

VAUGHT, Jim. Biobanking comes of age: the transition to biospecimen science. **Annual review of pharmacology and toxicology**, v. 56, p. 211-228, 2016.

VELCULESCU, Victor E. *et al*. Characterization of the yeast transcriptome. **Cell**, v. 88, n. 2, p. 243-251, 1997.

VITT, Dick Le. Technical note: Beyond Good Laboratory Practice. **Journal of Analytical Methods in Chemistry**, v. 14, n. 5, p. 189-191, 1992.

WALLS, Craig. Spring Boot in Action. **Manning Publications Company**, 2016.

WASINGER, Valerie C. *et al*. Progress with gene- product mapping of the Mollicutes: Mycoplasma genitalium. **Electrophoresis**, v. 16, n. 1, p. 1090-1094, 1995.

WEISSMANN, Henrique Lobo. Vire o jogo com Spring Framework. Editora **Casa do Código**, 2014.

WEINSTEIN, John N. Fishing expeditions. Science, v. 282, n. 5389, p. 627-627, 1998.

WFCC. Guidelines for the establishment and operation of collections of cultures of microorganisms. 3ª edition, 2010.

WILSON, Greg *et al*. Best practices for scientific computing. PLoS Biol, v. 12, n. 1, p. e1001745, 2014.

YANDELL, Mark; ENCE, Daniel. A beginner's guide to eukaryotic genome annotation. **Nature Reviews Genetics**, v. 13, n. 5, p. 329-342, 2012.

APPENDIX I – FUNCTIONAL REQUIREMENTS TABLE

Code	Requirement	Priority
RF001	Register collaborator	Essential
RF002	Modify collaborator	Essential
RF003	Deactivate collaborator	Essential
RF004	Visualize a collaborator	Essential
RF005	Visualize all collaborators	Essential
RF006	Generate reports about collaborators	Desirable
RF007	Register user	Essential
RF008	Modify user	Essential
RF009	Deactivate user	Essential
RF010	Visualize a user	Essential
RF011	Visualize all users	Essential
RF012	Generate reports about users	Desirable
RF013	Register laboratory	Essential
RF014	Modify laboratory	Essential
RF015	Deactivate laboratory	Essential
RF016	Visualize a laboratory	Essential
RF017	Visualize all laboratories	Essential
RF018	Generate reports about laboratories	Desirable
RF019	Register strain	Essential
RF020	Modify strain	Essential
RF021	Exclude strain	Essential
RF022	Visualize a strain	Essential
RF023	Visualize all strains	Essential
RF024	Generate reports about strains	Desirable
RF025	Register bibliographic production related to strain	Important
RF026	Edit bibliographic production related to strain	Important
RF027	Exclude bibliographic production related to strain	Important
RF028	Download bibliographic production related to strain	Important
RF029	Visualize a bibliographic production related to strain	Important
RF030	Visualize all bibliographic production related to strain	Important
RF031	Register the receipt of a strain	Essential
RF032	Modify the receipt of a strain	Essential
RF033	Exclude the receipt of a strain	Essential
RF034	Register the send of a strain	Essential
RF035	Modify the send of a strain	Essential
RF036	Exclude the send of a strain	Essential
RF037	Register technique	Essential
RF038	Modify technique	Essential
RF039	Exclude technique	Essential
RF040	Visualize a technique	Essential

RF041	Visualize all techniques	Essential
RF042	Register the application of a technique to a strain	Essential
RF043	Modify the application of a technique to a strain	Essential
RF044	Exclude the application of a technique to a strain	Essential
RF045	Visualize the techniques applied to a strain	Essential
RF046	Visualize the techniques applied during a time period	Desirable
RF047	Visualize the techniques applied to a strain during a time period	Important
RF048	Generate a report with the techniques applied to a strain	Desirable
RF049	Generate a report with the techniques applied during a time period	Desirable
RF050	Generate a report with the techniques applied to a strain during a time period	Desirable
RF051	Register a freezer	Essential
RF052	Modify a freezer	Essential
RF053	Exclude a freezer	Essential
RF054	Visualize a freezer	Essential
RF055	Visualize all freezers	Essential
RF056	Generate reports about freezers	Desirable
RF057	Register the sequencing of a strain	Essential
RF058	Modify the sequencing of a strain	Essential
RF059	Exclude the sequencing of a strain	Essential
RF060	Visualize the sequencing of a strain	Essential
RF061	Visualize the sequencings of a strain	Essential
RF062	Visualize the sequencings that occurred during a time period	Desirable
RF063	Generate reports about the sequencing of a strain	Desirable
RF064	Generate reports about the sequencings that occurred during a time period	Desirable
RF065	Register the assembly of a strain	Essential
RF066	Modify the assembly of a strain	Essential
RF067	Exclude the assembly of a strain	Essential
RF068	Visualize the assembly of a strain	Essential
RF069	Visualize all assemblies of a strain	Essential
RF070	Visualize the assemblies that occurred during a time period	Desirable
RF071	Generate reports about the assemblies of a strain	Desirable
RF072	Generate reports about the assemblies that occurred during a time period	Desirable
RF073	Register the final sequence of the genome	Essential
RF074	Modify the final sequence of the genome	Essential
RF075	Exclude the final sequence of the genome	Essential
RF076	Visualize the final sequence of a genome	Essential
RF077	Visualize the genomes that have a final sequence	Important
RF078	Generate reports about final sequences	Desirable
RF079	Register the annotation of a genome	Essential

RF080	Modify the annotation of a genome	Essential
RF081	Exclude the annotation of a genome	Essential
RF082	Visualize the annotation of a genome	Essential
RF083	Visualize all genome that have annotation	Important
RF084	Register a software to be used in the annotation or in the assembly of a genome	Essential
RF085	Modify a software to be used in the annotation or in the assembly of a genome	Essential
RF086	Exclude a software to be used in the annotation or in the assembly of a genome	Essential
RF087	Visualize a software	Essential
RF088	Visualize all software	Essential
RF089	Generate reports about software	Desirable
RF090	Register the submission/deposit of a genome to a public database	Essential
RF091	Modify the submission/deposit of a genome to a public database	Essential
RF092	Exclude the submission/deposit of a genome to a public database	Essential
RF093	Visualize a genome submitted or to be submitted to a public database	Essential
RF094	Visualize the genomes submitted or to be submitted to a public database	Essential
RF095	Generate reports about genomes submitted or to a public database	Desirable
RF096	Register a publication related to one or more genomes deposited in a public database	Essential
RF097	Modify a publication related to one or more genomes deposited in a public database	Essential
RF098	Exclude a publication related to one or more genomes deposited in a public database	Essential
RF099	Visualize a publication related to one or more genomes deposited in a public database	Essential
RF100	Visualize all publications related to one or more genomes deposited in a public database	Essential
RF101	Visualize the genomes and their respective status in the NGS analysis	Important
RF102	Visualize the history of actions of the LabControl	Essential

APPENDIX II – LABCONTROL'S INTERFACE ADDITIONAL FIGURES

Figure 16 -Screen used to register a strain in LabControl showing taxonomic information

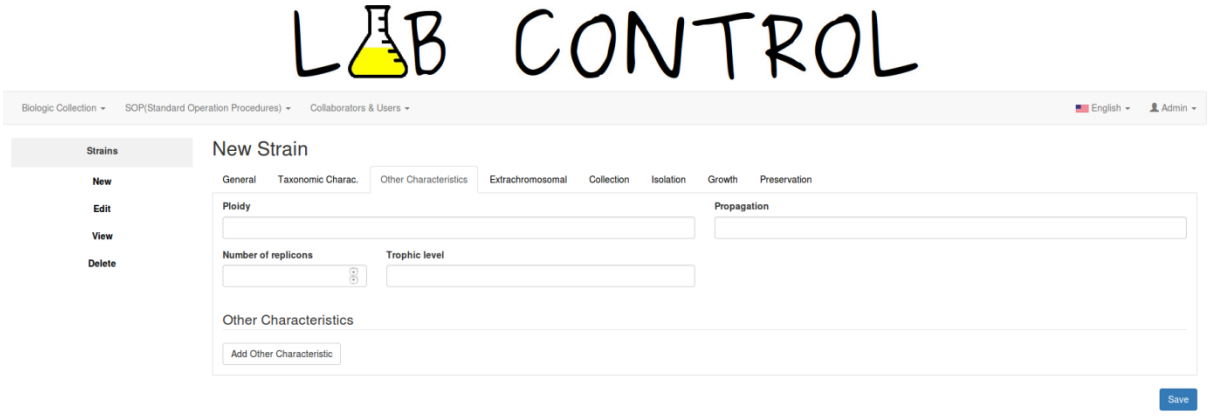

Figure 17 - Screen used to register a strain in LabControl showing information about other characteristics

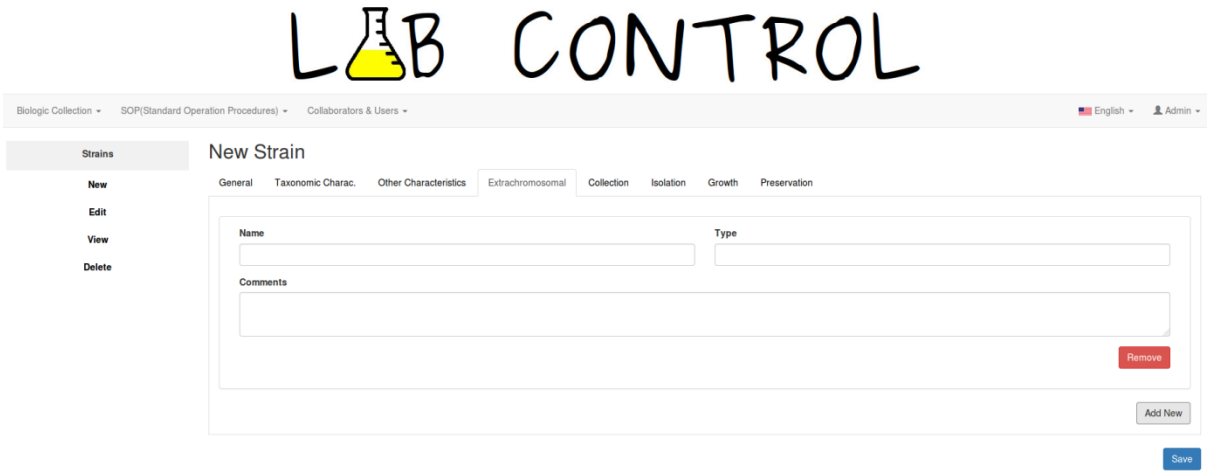

Figure 18- Screen used to register a strain in LabControl showing extrachromosomal information

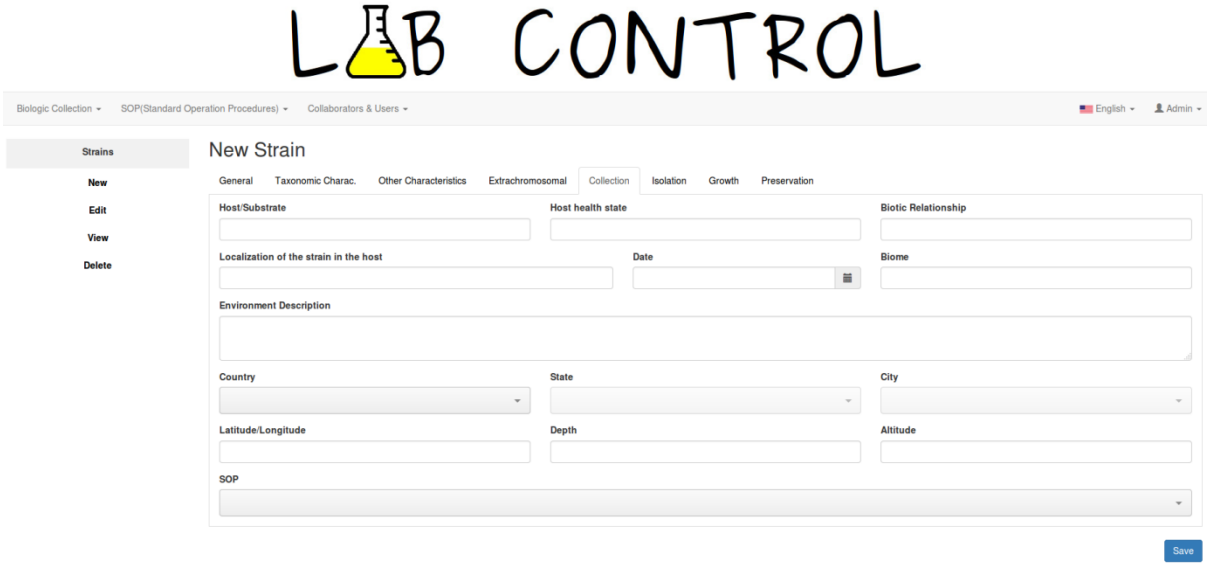

Figure 19 - Screen used to register a strain in LabControl showing collection information

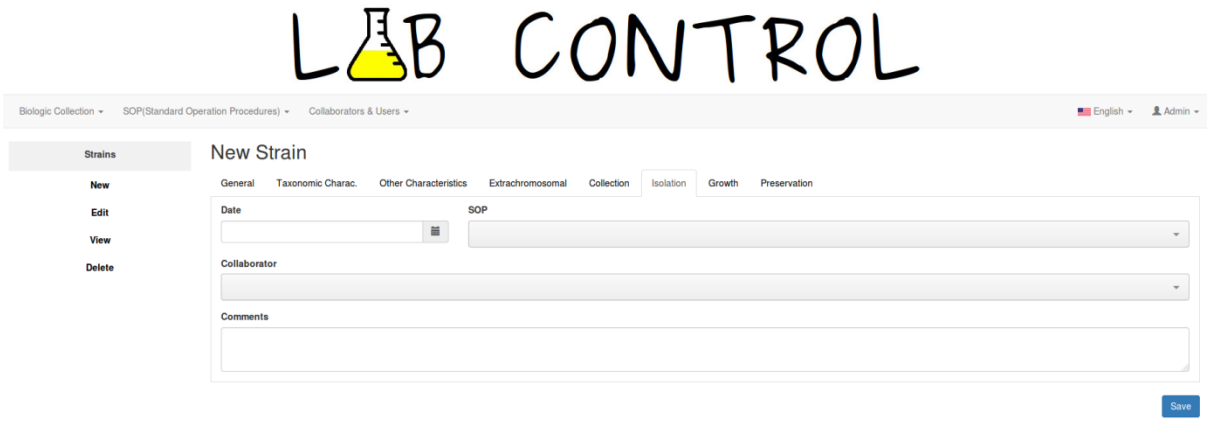

Figure 20 - Screen used to register a strain in LabControl showing isolation information

Figure 21 - Screen used to register a strain in LabControl showing growth information

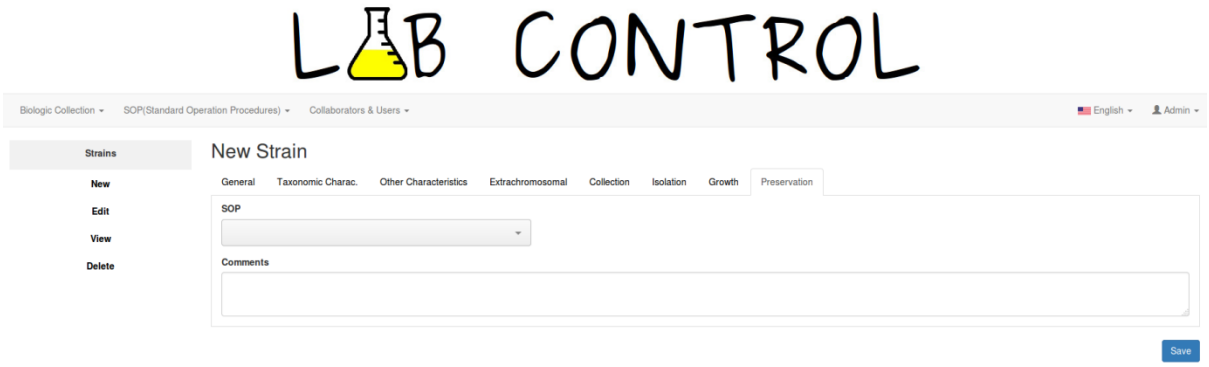

Figure 22 - Screen used to register a strain in LabControl showing preservation information

APPENDIX III – SOFTWARE REGISTRATION

REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DO DESENVOLVIMENTO, INDÚSTRIA E COMÉRCIO EXTERIOR
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE CONTRATOS, INDICAÇÕES GEOGRÁFICAS E REGISTROS

CERTIFICADO DE REGISTRO
DE PROGRAMA DE COMPUTADOR

Processo: BR 51 2015 001045-0

O INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL expede o presente Certificado de Registro de Programa de Computador, **válido por 50 anos** a partir de 1º de janeiro subsequente à data de criação indicada, em conformidade com o art. 3º da Lei Nº 9.609, de 19 de Fevereiro de 1998, e arts. 1º e 2º do Decreto 2.556 de 20 de Abril de 1998.

Título:	**LABCONTROL**
Criação:	31 de agosto de 2015
Titular(es):	UNIVERSIDADE FEDERAL DE MINAS GERAIS (17.217.985/0001-04)
Autor(es):	FLÁVIA DE SOUZA ROCHA ( ) MARIANA TEIXEIRA DORNELLES PARISE ( ) VASCO ARISTON DE CARVALHO AZEVEDO ( )
Linguagem:	JAVA
Aplicação:	AD-08, BL-02, BL-04
Tipo Prog.:	GI-01, GI-02, GI-04

DOCUMENTAÇÃO TÉCNICA EM DEPÓSITO SOB SIGILO ATÉ 18/09/2025.

Os Direitos Patrimoniais relativos ao programa de computador objeto do presente registro foram cedidos dos Criadores para o Titular, na data de 04 de setembro de 2015, conforme documentação

A exclusividade de comercialização deste programa de computador não tem a abrangência relativa à exclusividade de fornecimento estatuída pelo art.25, I, da Lei nº8.666, de 21 de Junho de 1993, para fins de inexigibilidade de licitação para compras pelo poder público.

Expedido em 07 de junho de 2016

Assinado digitalmente por:

Breno Bello de Almeida Neves
Diretor de Contratos, Indicações Geográficas e Registros