“Configuration of an open source laboratory information management system (Senaite-LIMS) for a sequencing laboratory”

Estudiante: Carlos Fernández Medina
Plan de Estudios: Máster Universitario en Bioinformática y Bioestadística
Área del trabajo final: TFM-Bioinformática y Bioestadística Área 4 aula 1

Consultora: Elisabeth Ortega Carrasco
Profesor responsable: David Merino Arranz

Fecha de entrega: 05-01-2021

FICHA DEL TRABAJO FINAL

Título del trabajo:	Configuration of an open source laboratory information management system (Senaite-LIMS) for a sequencing laboratory.
Nombre del autor:	Carlos Fernández Medina
Nombre del consultor/a:	Elisabeth Ortega Carrasco
Nombre del PRA:	David Merino Arranz
Fecha de entrega (mm/aaaa):	01/2020
Titulación::	Máster Universitario en Bioinformática y Bioestadística
Área del Trabajo Final:	TFM-Bioinformática y Bioestadística Área 4 aula 1
Idioma del trabajo:	Inglés
Palabras clave	LIMS, Senaite, sequencing, agile, data management

Resumen del Trabajo (máximo 250 palabras):

El desarrollo de las tecnologías de secuenciación ha supuesto un cambio de paradigma en el campo de la genómica, ya que presentan múltiples utilidades que van desde el estudio de distintos tipos de cáncer o enfermedades raras como el estudio la filogenia de distintas especies, entre ellas del ser humano. No obstante, junto con el desarrollo de estas técnicas se ha hecho patente la necesidad de gestionar una gran cantidad de datos. Es por ello por lo que se hace necesario disponer de herramientas que permitan almacenar datos de manera ordenada y que sean amigables para los usuarios, con el objetivo de que exista una trazabilidad y un control de la información. Por esta razón, los sistemas de gestión de la información de los laboratorios (LIMS) son una muy buena solución para poder gestionar tal cantidad de datos generados en un laboratorio de secuenciación. En este trabajo, configuramos una herramienta LIMS tipo Open Source (Senaite-LIMS), adaptándola a las necesidades de los procesos que se llevan a cabo durante los procesos secuenciación, tales como control e identificación de muestras, manejo y visualización de resultados y ejecución de informes. Así mismo, llevamos a cabo un control del proyecto mediante una metodología agile de desarrollo de software, con el fin de describir los recursos utilizados, los desafíos, los factores clave de éxito y los resultados obtenidos en cada fase del proyecto.

Abstract (in English, 250 words or less):

Sequencing technologies development has represented a paradigm shift in the field of genomics, since they have multiple uses, from the study of different types of cancer or rare diseases to the study of the phylogeny of different species, including humans. However, the need to manage a large amount of data has become apparent along with the development of these techniques. Because of that, it is necessary to have tools that allow data to be stored in an orderly manner and in such a way that it is user-friendly,

with traceability and control of the information as the principal aims. For this reason, laboratory information management systems (LIMS) are a very good solution to be able to manage such an amount of data generated by sequencing laboratories. In this work, we configure an Open Source LIMS application (Senaite-LIMS) adapted to the needs of the processes that are carried out during sequencing process, such as control and identification of samples, result management and visualization and report printing. Likewise, we carry out a project control using an agile software development methodology, in order to describe the resources used, the challenges, the key success factors and the results obtained in each phase of the project.

INDEX

1. INTRODUCTION

1.1. CONTEXT AND JUSTIFICATION OF THE PROJECT

1.1.1. A brief history of sequencing

The finding of the three-dimensional (3-D) structure of Deoxyribonucleic Acid (DNA) composed of four bases (A, T, C, G) in 1953 by James Watson, Francis Crik, Maurice Wilkins and Rosalind Franklin was one of the most important events in the history of molecular biology to which contributed to a conceptual framework for both DNA replication and encoding proteins in nucleic acids (1,2). This important discovery has led to the decoding of genomic sequences and know DNA composition of organisms. The DNA sequencing is the discovery that uses the DNA composition to understand and decrypt the code to all biological life on earth as well as to understand and treat genetic diseases (1).

1.1.1.1. First-generation sequencing

Initial efforts focused on sequencing the most readily available populations of Ribonucleic Acid (RNA) species to produce fully and partially degraded RNA fragments(2). In 1965 Robert Holley and colleagues produced the first whole nucleic acid sequence of alanine transfer RNA (tRNA) from *Saccharomyces cerevisiae*. In parallel, Fred Sanger and colleagues developed a related technique based on the detection of radiolabelled partial-digestion fragments after two-dimensional (2-D) fractionation. It was also by using this 2-

D fractionation method that Walter Fiers' laboratory started to produce the first complete protein coding gene sequence in 1972 followed four years later by its complete genome.

It was around this time that researchers began to adapt their methods in order to sequence DNA, aided by the recent purification of bacteriophages with DNA genomes. Ray Wu and Dale Kaiser used DNA polymerase to fill the ends in with radioactive nucleotides, supplying each nucleotide one at a time and measuring incorporation to deduce sequence. This principle was generalized through the use of specific oligonucleotides to prime the DNA polymerase. Incorporation of radioactive nucleotides started to be used to infer the order of nucleotides anywhere.

The next practical change to make a large impact was the replacement of 2-D fractionation (which often consisted of both electrophoresis and chromatography) with a single separation by polynucleotide length via electrophoresis through polyacrylamide gels, which provided much greater resolving power (2). This technique was used in two complex protocols from the mid-1970s: Alan Coulson and Sanger's “plus and minus” system in 1975 and Allan Maxam and Walter Gilbert's chemical cleavage technique (1). These protocols used DNA polymerase to synthesize from a primer, incorporating radiolabelled nucleotides, before performing two second polymerisation reactions. By running the products on a polyacrylamide gel and comparing between the eight lanes, one is able to infer the position of nucleotides at each position in the covered sequence. Using this technique Sanger and colleagues sequenced the first DNA genome of bacteriophage φX174. This was the birth of first-generation DNA sequencing and opened the door to study the genetic code of living beings and to the development of faster and efficient sequencing technology (1,2).

However, the major breakthrough that altered the progress of DNA sequencing technology came in 1977 with the development of Sanger's “chain-termination” or dideoxy technique (2,3). The chain termination technique makes use of Dideoxynucleotides (ddNTPs). Mixing radiolabelled ddNTPs into a DNA extension reaction at a fraction of the concentration of standard Deoxynucleotides (dNTPs) results in DNA

strands of each possible length being produced, as the ddNTPs get randomly incorporated as the strand extends, halting further progression. By performing four parallel reactions containing each individual ddNTP base and running the results on four lanes of a polyacrylamide gel we can infer what the nucleotide sequence in the original template was, as there will a radioactive band in the corresponding lane at that position of the gel. While working on the same principle as other techniques, the accuracy, robustness and ease of use led to the dideoxy chain-termination method (or Sanger sequencing) to become the most common technology used to sequence DNA for years to come (2).

1.1.1.2. Second-generation or Next-Generation Sequencing (NGS)

Sanger and Maxam-Gilbert sequencing technologies were the most common sequencing technologies used by biologists until the emergence of a new era of sequencing technologies opening new perspectives for genomes exploration and analysis (1). A number of improvements were made in the following years and contributed to the development of increasingly automated DNA sequencing machines and subsequently the first crop of commercial DNA sequencing machines which were used to sequence the genomes of increasingly complex species. These DNA sequencing machines produce reads slightly less than one kilobase (kb) in length. The development of techniques such as polymerase chain reaction (PCR) and recombinant DNA technologies further aided the genomics revolution by providing means of generating the high concentrations of pure DNA species required for sequencing. Also, more sequenced genomes and tools for genetic manipulation provided the resources to find polymerases that were better at accommodating the additional chemical moeities of the increasingly modified dNTPs used for sequencing. Eventually, newer dideoxy sequencers came to be used in the Human Genome Project.

Concurrent with the development of large-scale dideoxy sequencing efforts, another technique appeared that set the stage for the first wave of DNA sequencers. Researchers utilized a recently discovered luminescent method for measuring pyrophosphate

synthesis. This approach was used to infer sequence by measuring pyrophosphate production as each nucleotide is washed through the system in turn over the template DNA affixed to a solid phase. This technique could be performed using natural nucleotides (instead of the heavily-modified dNTPs used in the chaintermination protocols), and observed in real time (instead of requiring lengthy electrophoreses). Pyrosequencing was later licensed to 454 Life Sciences (later purchased by Roche), where it evolved into the first major successful commercial NGS technology (1,2). These sequencing machines were a paradigm shift in that they allowed the mass parallelisation of sequencing reactions, greatly increasing the amount of DNA that can be sequenced in any one run.

Few years after that, a number of parallel sequencing techniques sprung up following the success of 454 (2). Sequencing by Oligo Ligation Detection (SOLiD) was purchased by Applied Biosystems (ABI) in 2006. The sequencer adopts the technology of two-base sequencing based on ligationsequencing and produced short reads with length 35 base pair (bp) and output of 3 Gb/run and continued to improve their sequencing which increased the length of reads to 75 bp with an output up to 30 Gb/run. The strength of ABI/SOLiD platform is high accuracy because each base is read twice while the drawback is the relatively short reads and long run times. but the errors of sequencing in this technology is due to noise during the ligation cycle which causes error identification of bases (1,4).

During the same year, Solexa released the Genome Analyzer (GA) and in 2007 the company was purchased by Illumina. The sequencer adopts the technology of sequencing by synthesis (SBS) which the library with fixed adaptors is denatured to single strands and grafted to the flowcell, followed by bridge amplification to form clusters which contains clonal DNA fragments(4). The first sequencers Illumina/Solexa GA has been able to produce very short reads around 35 bp and the output data of the last Illumina sequencers is currently higher than 600 giga base pair (Gbp) and lengths of short reads are about 125 bp. One of the main drawbacks of the Illumina/Solexa platform is the high requirement for sample loading control because overloading can result in overlapping clusters and poor sequencing quality (1).

In 2010, Life Technologies commercialized the Ion Torrent semiconductor sequencing technology in 2010, which is similar to 454 pyrosequencing technology, but based on the detection of the hydrogen ion released during the sequencing process (5) The Ion Torrent sequencers are capable of producing reads lengths of 200 bp, 400 bp and 600 bp with throughput that can reach 10 Gbp for ion proton sequencer. The ion torrent sequencing advantages are are focused on read lengths which are longer to other sequencers and fast sequencing time between 2 and 8 hours. The major disadvantage is the difficulty of interpreting the homopolymer sequences (1).

The development of all these technologies have made the NGS revolutionize genomic and genetic research.

1.1.1.3. Sequencing data management

The study of genomics increasingly is becoming a field that is dominated by the growth in the size of data and the responses by the broader scientific community to effectively use and manage the resulting derived information (6). Genomics technologies have moved from gene to genome sequencing and are now capable of sequencing whole environments of microorganisms. The production of this metagenomic data, however, generates another challenge for data management, because sequences cannot always be associated with specific species as has been traditional for gene and genome sequence data management (7).

Also, the management of the big amount of data generated is an important challenge of the use of sequencing, specially NGS approach in clinical diagnostic. Indeed generation, analysis and also storage of sequencing data require sophisticated bioinformatics infrastructure to manage and analyse the data, and so both computing infrastructure and manpower impact on costs of applications in clinical diagnostics (8).

1.1.2. Laboratory Information Management Systems (LIMS)

A LIMS is a specific software that allows the acquisition and management of the information generated in a laboratory. Until 1970's management of laboratory samples and their associated analysis and reporting was a time-consuming manual processes so some organizations custom their own solutions for individual laboratories, while some enterprising entities at the same time sought to develop a more commercial reporting solution in the form of special instrument-based systems (9).

The first generation LIMS products appeared in the 1980's as the same time of the development of automatization of the laboratories. They were introduced in the form of a single centralized minicomputer, which offered laboratories the first opportunity to utilize automated reporting tools.

A second generation of LIMS were tapping into relational databases to expand LIMS into more application-specific territory. In the early 1990s personal computers became more powerful and prominent and third generation of LIMS emerged. These new LIMS took advantage of the developing client/server architecture, allowing laboratories to implement better data processing and exchanges and processing of data anywhere on the network. Web-enabled LIMS were introduced the following years, enabling researchers to extend operations outside the confines of the laboratory.

By the early 2010s, some LIMS added additional characteristics that continued to shape how a LIMS was defined. Examples include the addition of clinical functionality and electronic laboratory notebook (ELN) functionality, as well a rise in the cloud-based software as a service (SaaS) distribution model (10).

Nowadays, their use has been expanded not only to sample and results management, but also to interface different systems and instruments. In addition, the importance of having a LIMS has increased due to quality regulations like ISO 17025 (11). Because of that, a certified quality laboratory must have a software to audit trail implementation, authentication protocols, configuration management, data backup, data integrity,

electronic signature implementation, encryption, information privacy and network security (12). LIMS can contribute to it.

1.1.2.1. Existing LIMS

We can find different LIMS vendors and they can be classified according to the way we can use them:

- Commercial LIMS brands: some specialized in automatization brands offer their own LIMS. That is the example of Illumina, which has developed specific LIMS for manage sequencing data.
- LIMS specialized software companies: there are several companies which develop LIMS and offer different services to adapt the system for laboratory necessities. A licence membership is mandatory to use this kind of software. We can find examples in different companies, such as LabWare, Thermo Fischer Scientific, Abbot Informatics STARLIMS, etc.
- Open source LIMS: there are some LIMS open to all the community interested in implementing a software for managing laboratory data. This kind of LIMS are free and customizable, so is a low-cost alternative. Senaite/Bika LIMS, Free LIMS, Open LIMS are some of the examples.

1.1.2.2. Senaite-LIMS

Senaite-LIMS is a web-based Open Source LIMS for enterprise environments, especially focused to behave with excellent performance and stability (13). It was launched on GitHub in October 2017 and deployed and maintained by two start-up companies, Naralabs (Barcelona, Spain) and Rinding Bytes (Nuremberg, Germany).

The software is a derivative work of another open source LIMS (Bika LIMS), built on top of Plone Content Management System (CMS) with Python as its main programming

language. The principle architectural changes with respect to its predecessor Bika LIMS is that they have developed Senaite as a system of independent add-ons. This, combined with quality code standards, makes the application much easier to maintain and to contribute to and allows it to stay level with emerging laboratory requirements. To help in achieving so, Senaite is developed under the paradigm of continuous integration and continuous delivery (13).

The main examples of add-ons for Senaite are:

- *senaite.core:* provides the core functionalities and entities used by Senaite.
- *senaite.health*: to make Senaite suitable for health care laboratories, with additional capabilities for the management of patients, doctors, clinical cases, symptoms, and a lot more.
- *senaite.impress*: rendering engine for HTML documents to PDF in order to publish results.
- *senaite.jsonapi:* allows to create, read and update operations through http GET/POST requests.

The principal features of Senaite are (14):

- No licensing costs.
- Robust and secure
- Interoperability with business software.
- Improves the reporting speed, sample tracking and process efficiency.
- Traceability in the laboratory processes reducing the error correction time.
- Total Quality Management
- Powerful live search engine.
- Clean User Interface that works on any computer or smart device.

- Enterprise driven LIMS.
- Built on Plone CMS.
- Fully Open Source.
- Integrated RESTful JSON API.
- Customizable reporting machinery and data mining.
- Modular architecture.
- Web based and responsive.
- Low maintenance and installation costs.
- System developed in Python.
- Provide a robust user interface which follows industry standards.

The schema of Senaite is shown on figure 1 (15).

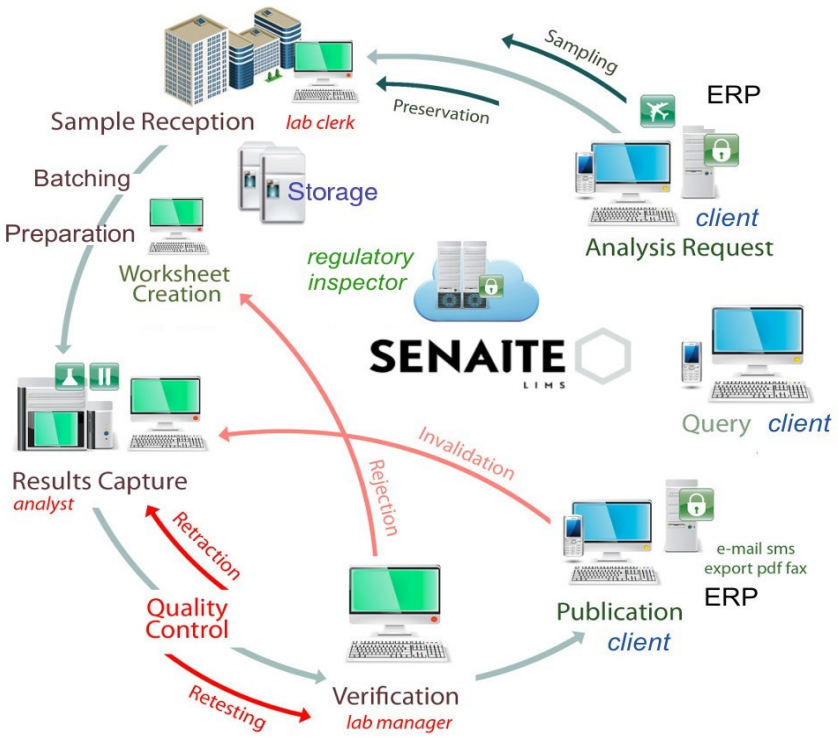


Figure 1: *General schema of Senaite-Bika LIMS, with different sample and result status and what roles privileges are. Modified from Bika web page (15).*

1.2. OBJECTIVES

1. Configure an Open Source LIMS system to manage sequencing data.
    I. Select a LIMS system.
    II. Find real sequencing data.
    III. Install the LIMS selected.
    IV. Configure LIMS.
    V. Configure result reports.
    VI. Create and develop interfaces.
    VII. Test the system.
    VIII. Put LIMS into operation.

2. Manage the project using agile tools.
    I. Select agile tools to manage the project.
    II. Write the final report and do the presentation.

1.3. APPROACH AND METHODS

There are some open source LIMS available, so we decided to find information about them and configure Senaite LIMS, adapting it to the requirements. This strategy is the most suitable because we can use an existed software and modified and configure it,

which can reduce costs and time. To manage the project, we use the principles of an agile software development methodology.

We focused in the requirements showing in a MoSCoW method (table 1).

REQUIREMENTS	MuSCoW
Open agile tools	M
Open Access and customizable LIMS.	M
Safety and backup.	M
Trustworthy.	M
Allow sample tracking.	M
Identify who is using the application and what changes have done.	M
Allow to print reports.	S
Easy to use.	S
Connected to sequencing equipment (instrument interfaces).	C
Sample automation.	W

Table 1: *MoSCoW method of prioritization of requirements. M: must have; S: should have; C: could have; W: would not have.*

1.4. PLANNING

1.4.1. Tasks and schedule

We identified 31 tasks form each objective and we have classified them according to the priority (table 2).

OBJECTIVES	TASKS	PRIORITY
1: SELECT A LIMS	Search different LIMS documentation	
	Search different LIMS bibliopgraphy	
	Search sequencing bibliography	
2: SELECT AGILE TOOLS	Create JIRA and GitHub accounts	
	Create GitHub account and configure the programme	
3: FIND REAL DATA	Search real sequencing data	
4: INSTALL LIMS	Prepare operating system	
	Install data base	
	Install LIMS development environment	
5: CONFIGURE LIMS	Create and configure users, roles and clients	
	Create and configure analysis	
	Create and configure samples	
	Create methods specifications	
	Create and configure instruments	
6: CREATE REPORTS	Select open source reports tool	
	Configure report templates	
	Create and configure roles and clients	
7: CREATE INTERFACES WITH INSTRUMENTS	Prepare sequencing data files	
	Configure parsings to import data	
	Configure and code reports	
8: WRITE FINAL REPORT AND PRESENTATION	Prepare objects	
	Create instalation manuals	
	Create admin and user manuals	
	Write the final project report	
	Do presentation	

Tabla 2: *Tasks and priority. Red, the most priority tasks; yellow, the least priority; blue, optional tasks.*

We have determined the planning in a Gannt chart, where we have defined the tasks to complete each secondary objectives (see Anex I). We have also transcript the Gannt diagram to a Jira repository (figure 2), which is a software development tool used by agile teams to plan, track and release software projects (16):

- https://fdezmedina-carlos.atlassian.net/jira/software/projects/TFM/boards/1/roadmap?selectedIssue=TFM-2

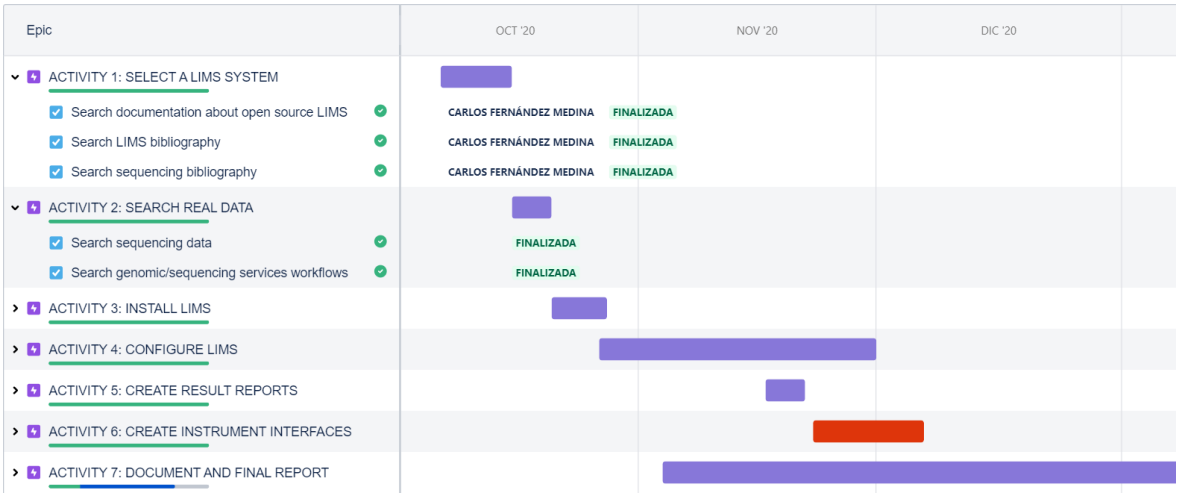


Figure 2: *Detail of the Jira planner. We can control the deadlines of each activity or sub-activity and change the state of each one to "to do", "in progress" or "complete".*

1.4.3. Milestones

The project has been divided in three phases (table 3) agree to the last day of each master programme evaluation activity (PEC). We also have established the deliverables for each phase.

PHASE	DEADLINE	DELIVERABLES
Phase 0	13rd October 2020	- GitHub repository. - Jira planner. - PEC0 document.
Phase 1	16th November 2020	- First version of LIMS configured. - Installation manual. - Admin manual. - First version of final report. - PEC1 document.
Phase 2	14th December 2020	- Final version of LIMS.

		- User manuals. - Advanced version of final report. - PEC2 document.
Phase 3	5th January 2021	- Final version of final report.
Phase 4	10th January 2021	- Presentation.

Table 3: *Milestones and deliverables.*

All of the deliverables were attached to GitHub repository: https://github.com/cferna0256/TFM_LIMS-Bioinf

1.4.4. Risks

Some risks were identified and classified in a risk matrix (table 4).

RISKS	PROBABILITY	IMPACT
Lack of technical knowledge	3	4
Lack of technical documentation	3	4
Lack of bibliography	2	4
Too ambitious project	3	5
Bad estimation of times	3	4
Lack of communication with tutor	1	3
Lack of free sequencing data	2	2
The selected system does not fit with the requirements	2	5
Sick leave	3	4
Technical problems with operating system or PC	3	5

Table 4: *Risk matrix. From 1 to 5 (very low, low, medium, high and very high) attending to its probability and its impact.*

1.5. SUMMARY OF OBTAINED PRODUCTS

During this project we have obtained some products which have been attached to the next GitHub repository: https://github.com/cferna0256/TFM_LIMS-Bioinf

- Description about the project done: *README.md, LICENSE.*
- An url to the Oracle Virtual Machine with our configured Senaite-LIMS system with a desktop shortcut to access to the system (*Ubuntu-TFM-Senaite-1.0.ova*).
- An url to JIRA repository.
- A file with an explanation of the main configurations that we have made into Senaite: C*onfigurations.md*.
- A file with explanations of the installation: *Installation.md.*
- A user manual with the main Senaite functionalities: *UserManual.md.*
- The Final Report, explaining the different parts of the project: *FernandezMedina_Carlos_TFM_FinalReport.pdf*
- Other modified configuration objects: *senaite.impress, senaite.core, buildout.cfg, requirements.txt.*
- Desktop shortcut files: s*enaite_logo.png* and *senaitelims.desktop.*
- Script bash file to execute the Senaite-LIMS instance: *senaite_lims_start.sh*

1.6. SUMMARY OF OTHER CHAPTERS

In this report we have included the next chapters:

- Methods: we explain the methodologies and materials used to the development of the LIMS.
- Results: we explain the results obtained.

- Discussion: in this chapter we explain the obtained results, the advantages and disadvantages of Senaite LIMS and the problems that occurred during the development of the project.

2. METHODS

2.1. PROJECT PLAN

We followed the development of the system by agile methodology. Agile methodologies allow adapting the way of working to the conditions of the project, achieving flexibility and immediacy in the response to adapt the project and its development to the specific circumstances of the environment (17)

A requirements matrix was made by a MoSCoW chart. Furthermore, a Gantt diagram was created in Excel to set project schedule, millstones and deliverables and transferred to a Jira repository. We divided the project in three phases. In addition, a risks matrix was made to determine possible risks and their impact. All of these was transfer to JIRA software.

Finally, we created a GitHub repository to control versions of the objects and deliverables.

2.2. LIMS SELECTION

We selected Open Source Senaite-LIMS based on the next must-be criteria:

- Free and open source software.
- Customizable.
- Safety and back up.

- Trustworthy.
- Allow sample tracking.
- Identify who is using the application and what changes have done.

A search information about the different open source LIMS was done using the web page *LIMS Wiki*, which is a WIKI where we could find detailed information about laboratory systems (12).

2.3. INSTALLATION

First, we install an Ubuntu 20.04.1 64-bit operative system (OS) in Oracle Virtual Machine 6.1.14. Then, we installed Senaite 1.3.4 following the instructions of Senaite web page (18) and using the virtual Python environment Miniconda.

As requirement of the software we use Plone 4.3.19, which is an open source and free content management system build on top of the Zope application server (19). Zope is a web application storage server written in Python language. The database management system used is ZODB (Zope Database) which is a database native to Python and makes it very fast for query or read operations because the language is not separate from it. In addition, ZODB provides a JSON-API for integrate Senaite and other systems. We use Python 2.7.18 and Zope 2.13.29.

In addition, we have installed some available add-ons according to our necessities. These add-ons were added on *buildout.cfg* file. This file is necessary for install Senaite in our OS.

2.4. LIMS CONFIGURATION

After the installation we run the server instance and we configure the non-coding parts of the software. Senaite allows *admin* user to make this configurations through *setup* menu (figure 3). In this menu we can find all the items to configure the software for laboratory necessities. The configuration was made following the user manuals of GitHub Senaite repository and the information of Senaite web site.

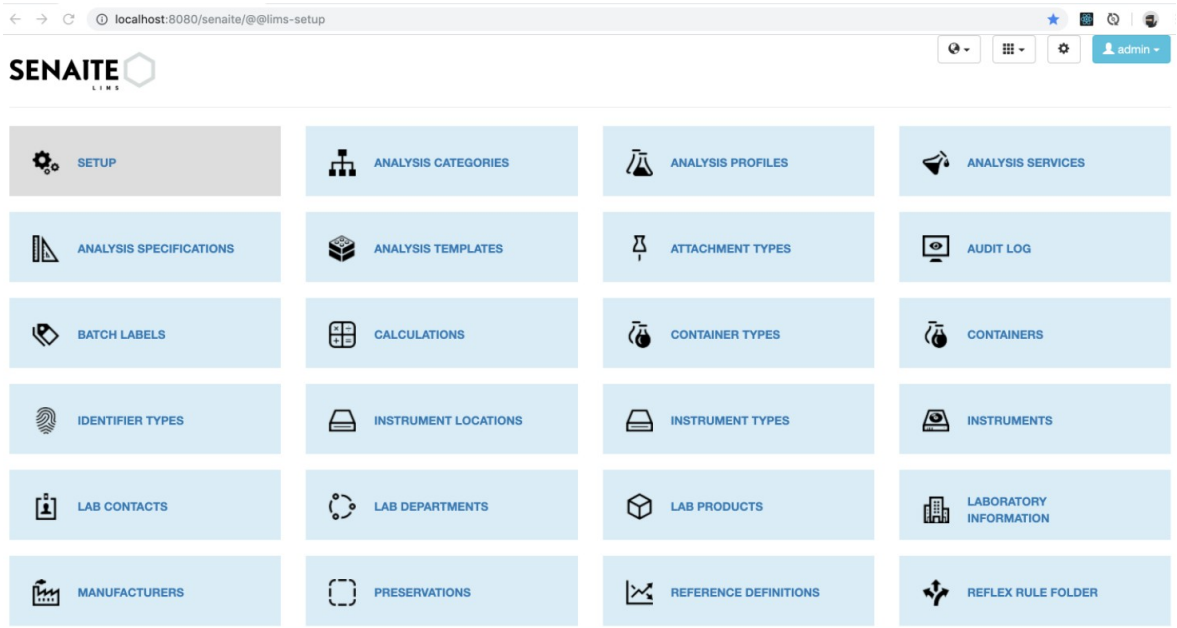


Figure 3: *Senaite Set Up menu.*

We adapt the software to the necessities and workflows of genomic laboratory services, based on some real genomic laboratory services workflows and equipment of *Servicio de Genómica de la Universidad Complutense de Madrid* (20), *Laboratorio de Secuenciación del Instituto de Investigación Hospital Universitario La Paz* (21), *Servicio de Secuenciación*

ADN, genómica y Proteómica de la Universidad de Salamanca (22) and *Sanger Sequencing Service of Macrogen* (23).

We divided the configurations in six groups:

1. *Users, roles and clients.*
2. *Analysis.*
3. *Samples.*
4. *Specifications and methods.*
5. *Instruments.*
6. *General configurations (SetUp menu).*
7. *Reports.*

2.5. LIMS MODIFICATIONS

As we detailed on the *Introduction* section, Senaite LIMS administrator is allowed to modify its code through *senaite.core*. So, we make some changes using this add-on to add some fields or change others of the analysis request template, in order to adapt sample registration to our necessities.

On the other hand, Senaite LIMS has an add-on called *senaite.impress* through which we could modify and set the reports generated. So, we modified the report template called *Default.pt* to adapt the reports to our necessities.

2.6. FUNCTIONAL TESTS AND VALIDATION

In order to test the software functionalities and its configurations we made technical documentation to control changes and user manuals to understand the workflows. All of these documents were copied and submitted on the GitHub repository. Also, we have added to the Annexes of this report.

3. RESULTS

3.1. PROJECT PROGRESS

We show the project progress in table 5. We have complete the main programmed activities, except “Tests scripts” and “Configure Instrument Interfaces”. So, the total development percentage of the project is 81,25%.

ACTIVITY	PERCENTAGE
Select a LIMS system	100%
Find real Data	50%
Install LIMS	100%
Configure LIMS	100%
Configure Result Reports	100%
Configure Instrument Interfaces	50%
Test scripts	50%
Document and write final report	100%
Total	**81.25%**

Tabla 5: *Project progress.*

3.2. LIMS SELECTION

Senaite LIMS was the chosen system, not only because satisfies the criteria detailed before, but also because there are extensive documentation and a big support network behind this brand. In addition, it has some additional characteristics that allows the developers and users to adapt the system to their necessities.

Additionally, we have a deep reading of the documentation and characteristics of this LIMS and we found that its standard characteristics could fit with the requirements raised.

Especially, we found interesting the standard tables, the sample tracking and the possibility of configure reports.

Besides that, Senaite is an open source software and have a great and live community sharing their configurations and experience in different forums, which it is a good point to ask questions and find answers to our technical doubts or even free developments in GitHub repositories.

3.3. LIMS INSTALLATION

We followed the steps detailed on Senaite web site to install LIMS in our OS . The installation steps have been summarize into Anex II.

We also installed some available add-ons for Senaite which must will be used later. These add-ons must be indicated into *bindout.fg* file (see Anex II).

- *senaite.core.*

- *senaite.jsonapi.*

- *senaite.impress.*

After the installation, we have created a desktop shortcut (*senaitelims.desktop*) to open the url in Mozilla Firefox navigator (figures 4 and 5). The file created have been stored in *"homesenaite/.local/share/applications"* directory and take the information of other files stored in "*homesenaite/senaitelims/desktop/"* directory *(senaite_logo.png* and *senaite_lims_start.sh).*

We have create a bash script called *senaite_lims_start.sh* that contains the commands to initiate Senaite service (figure 6). We created a *crontab* file to execute the script everytime the system boots (figure 7).

```
[Desktop Entry]
Version = 1.0
Name = Senaite LIMS
Comment = Senaite LIMS for sequencing Labs by CFM
Exec = firefox localhost:8080/senaite/login/
Icon= /home/senaite/senaitelims/desktop/senaite_logo.png
Terminal = false
Type = Application
NoDisplay = false
```

Figure 4: *senaitelims.desktop file to open Senaite LIMS on Firefox.*

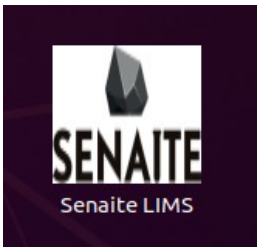


Figure 5: *Senaite shortcut into "Activities" menu of Ubuntu..*

```
#!/bin/bash
Open Senaite LIMS
echo "Opening Senaite LIMS"
conda activate senaite
cd /home/senaite/senaitelims
bin/instance fg
```

Figura 6: *senaite_lims_start.sh with a script to initiate Senaite service.*

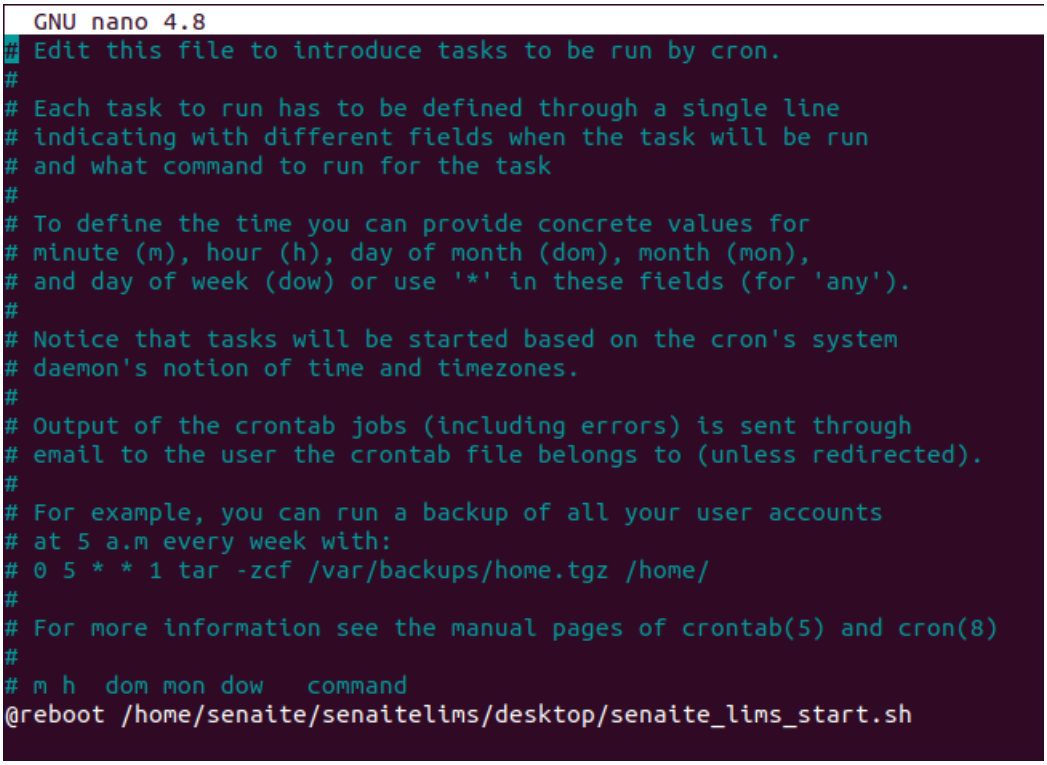


Figure 7: *Crontab file to execute senaite_lims_start.sh script when system boots.*

3.4. LIMS CONFIGURATION

For the LIMS configuration we followed up the configuration manuals and the recommendations of the web site.

3.4.1. Users, roles and clients

To access into the system, a number of five users have been configured (in addition to the admin role), all of them belong to the same laboratory: *Sequencing Service.*

- *Laboratory Manager*: this user can manage the LIMS from a functional perspective and have rights to modify analysis, instruments and also to verify and publish results and samples (user/password: *labman*).
- *Laboratory Clerk*: this user can manage basic setup such as clients and it has rights to register samples and verify results (user/password: *labclerk*).
- *Laboratory Analyst* and *Laboratory Analyst 2:* they have no rights to manage, only to enter analysis results (user/password: *labanalyst - labanalyst2*).

The permissions of each user are shown in Users Overview of Plone SetUp (*localhost:8080/senaite/@@usergroup-userprefs?set_language=en*, figure 8).

User name	Roles																
	Analyst	Client	Contributor	Editor	LabClerk	LabManager	Member	Preserver	Publisher	Reader	RegulatoryInspector	Reviewer	Sampler	SamplingCoordinator	Site Administrator	Verifier	Manager
labanalyst (labanalyst)	⊗	☐	☐	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
labanalyst2 (labanalyst2)	☑	☐	☐	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
labclerk (labclerk)	☐	☐	☐	☐	⊗	☐	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	☐
labman (labman)	☐	☐	☐	☐	☐	⊗	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Figure 8: *Users and permissions.*

Also, we created two fictitious clients in order to identify the samples:

- North Hospital (NH).

- Public Health Institute (PHI).

3.4.2. Analysis

We create two *Analysis Categories* to differentiate a total number of 10 *Analysis Services* that have been created. They were classified depending on when they are carrying out, before sequencing or during sequencing process itself (table 6).

Analysis Categories	Analysis Service	Units	Result option	Attachments
Pre-Sequencing	DNA – Extraction	nm ratio	Numeric	Permitted
	PCR	bp	Numeric	Permitted
Standard Sequencing	Forward – Chromatogram	NA	Boolean (OK/Not OK)	Mandatory
	Forward - GC content	%	String	Permitted
	Forward – Length	bp	Numeric	Permitted
	Forward – Sequence	NA	String	Permitted
	Reverse – Chromatogram	NA	Boolean (OK/Not OK)	Mandatory
	Reverse - GC content	%	String	Permitted
	Reverse – Length	bp	Numeric	Permitted
	Reverse – Sequence	NA	String	Permitted

Table 6: *Classification of Analysis services, according to their categorization. bp: base pair; NA: not applicable.*

Furthermore, *Analysis Templates* were created for services to appear by default when samples are requested:

- *Tissue – Sequencing:* to create samples with *DNA Extraction, PCR, Forward* and *Standard Sequence* services and *Tissue* as *Sample Type.*

- *DNA – Sequencing:* to create samples with *PCR, Forward* and *Standard Sequence* services and *DNA* as *Sample Type.*

- *PCR Product – Sequencing*: to create samples with *Standard Sequence* services and *PCR Product* as *Sample Type.*

3.4.3. Samples

Samples are the basic objects for LIMS, through which we will be able to obtain information on the results. In Senaite, they are called Analysis Request. To configure the samples, we must first configure the following tables:

- *Sample Type.*
- *Sample / Analysis Template.*
- *Container.*
- *Sub-Group:* we change the name to "*Organism*".
- *Sample Condition:* we change the name to "*Primers*".

The following *Sample Types* were created:

- *Tissue*: for samples that requires previous DNA extraction and amplification through PCR.
- *DNA*: samples that requires previous DNA amplification through PCR.
- *PCR Product*: PCR amplification samples, which are ready for sequencing.

We create the next *Containers:*

- *Tube.*
- *96 Well Plate.*

On the other hand, we create the next *Organisms:*

- *Microorganism (Bacteria).*
- *Microorganism (Eukarya).*
- *Plant.*
- *Animal.*
- *Human.*
- *Other.*

Finally, we create the *Primers* bellow:

- *User Primers*

For the registration of samples, the fields of the registration template are configured, keeping only the necessary ones. We also add two fields called "Organism" and "Primers".

In addition, we modified the sample identifier (ID) to show the date of registration of the sample and the client (see "*3.4.7. General configurations"*).

We also configure a *Supplier* called *Reference Samples* to use them to create *Reference Samples* when is needed.

3.4.4. Specifications

Analysis Specifications are configurated per *Sample Type*. They can be defined as the limits in which a result is OK or Not OK. To configure them we have to have the previously configured analyses and samples.

In the table 7 we summarize the analysis specifications created.

ANALYSIS SPECIFICATION	SAMPLE TYPE	ANALYSIS	LIMITS			
			Min war	Min	Max	Max warn
Tissue – Sequencing	Tissue	DNA Extraction (nm ratio)	NA	1.8	2.0	NA
		PCR (bp)	50	100	1000	2000
		Forward – Length (bp)	50	100	1000	2000
		Reverse – Length (bp)	50	100	1000	2000
DNA – Sequencing	DNA	PCR (bp)	50	100	1000	2000
		Forward – Length (bp)	50	100	1000	2000
		Reverse – Length (bp)	50	100	1000	2000
PCR Product – Sequencing	PCR Product	Forward – Length (bp)	50	100	1000	2000
		Reverse – Length (bp)	50	100	1000	2000

Table 7: *Analysis specifications and their limits. Min: minimum result allowed; Min warn: minimum result warned; Max warn: maximum result warned; Max: maximum result allowed; NA: not applicable.*

3.4.5. Instruments and methods

Senaite LIMS allows users to configure instruments to import and export results. We create the next instruments and configure them to export and import results in CSV format (but we do not configure any interface).

To register instruments we have to configure the following tables:

- *Instrument Types.*
- *Manufacturers.*
- *Methods.*
- *Instruments.*

The following *Instrument Types* were configured:

- *Thermo Cycler.*
- *Sequencing Instruments.*
- *DNA Extraction.*

We also create the next companies to configure the *Manufacturers*:

- *Agilent Technologies.*
- *Thermo Fischer Scientific – Applied Biosystems.*
- *Fluidigm.*

We configure the following *Methods*:

- *Spectrophotometry.*
- *Standard PCR.*
- *Standard Sanger Sequencing.*

After configure the above tables, we could configure the following *Instruments* (table 8):

INSTRUMENT	METHOD	INSTRUMENT TYPE	MANUFACTURER
2100 Bioanalyzer	Spectrophotometry	DNA Extraction	Agilent Technologies
3500 Genetic Analyzer	Standard Sanger Sequencing	Sequencing Intruments	Thermo Fischer Scientific – Applied Biosystems
Biomark HD System	Standard PCR	Thermo Cycler	Fluidigm

Table 8: *Instrument configuration.*

In addition, the system will warn if any instrument is out of calibration. So, we calibrate all the instruments.

3.4.6. Reports

The report configuration was mad thorugh the configuration menu of *senaite.impress add-on*. In this menu we set the following fields bellow (table 9).

Senaite Impress Settings	Function
Default template: “Default.pt”	Initially loaded report template.
Default orientation: “Landscape”	Initially loaded orientation.
Default paper format: “A4 210.0x297.0 mm”	Initially loaded paper format.
Store multi-report PDFs individually: “True“	Store reports individually if we are publishing different samples at the same time.

Table 9: *Senaite Impress settings.*

3.4.7. General configurations (*SetUp* menu)

Senaite LIMS allows admin user some general configurations through *Setup* menu. In our case, we made the modifications indicated in the table 10:

Set Up Tab	Modification
Security	Allow access to worksheets only to assigned analysts = False
	Only lab managers can create and manage worksheets = False

Accounting	Include a display pricing information = False
Samples	Auto-receive samples = True
Analysis	Categorise analysis services = True
	Default count of sample to add = 1
Id Server	AnalysisRequest = {clientId}{sampleType}{yymmdd}_{year}-{seq:04d}

Table 10: *Set up modifications.*

3.5. CODE MODIFICATIONS

Senaite have some add-ons which allow add some extra functionalities according to the necessities of the laboratory workflows. Specifically, *seniate.core* allows access for advanced users to the files that contain the code of Senaite. So, we have made the next modifications bellow.

3.5.1. *senaite.core* modifications

For adapting the software to the workflow requirements of a sequencing service, it is necessary to modify and add some of the fields of the sample record (*Analysis Request* window). To modify that, the standard add-on called *senaite.core* was modified, specifically, the file "*analysisrequest.py*" which can be found in the next directory: *"home/ senaite/buildout-cache/eggs/senaite.core-1.3.4-py2.7.egg/bika/lims/content/ analysisrequest.py".*

- Modifications in "*Sub-group"* reference field. We modify the label to "*Organism"* and the description to indicate the organism of the sample (figure 9).
- Modifications in "*Template*" reference field. We modify the field "required=True" to make mandatory this field (figure 10).

```
335     ReferenceField(
336         'SubGroup',
337         required=True,
338         allowed_types=('SubGroup',),
339         referenceClass=HoldingReference,
340         relationship='AnalysisRequestSubGroup',
341         mode="rw",
342         read_permission=View,
343         write_permission=FieldEditBatch,
344         widget=ReferenceWidget(
345             label=_("Organism"),
346             description=_("Type of organism in the sample"),
347             size=20,
348             render_own_label=True,
349             visible={
350                 'add': 'edit',
351             },
352             catalog_name='bika_setup_catalog',
353             colModel=[
354                 {'columnName': 'Title', 'width': '30',
355                  'label': _('Title'), 'align': 'left'},
356                 {'columnName': 'Description', 'width': '70',
357                  'label': _('Description'), 'align': 'left'},
358                 {'columnName': 'SortKey', 'hidden': True},
359                 {'columnName': 'UID', 'hidden': True},
360             ],
361             base_query={'is_active': True},
362             sidx='SortKey',
363             sord='asc',
364             showOn=True,
365         ),
366     ),
```

Figure 9: *"Sub-group" reference field modifed to show information about the organism.*

- Modifications in "*Sample Condition*" reference field. We modify the label to "*Primers"* and the description to indicate the primers of the sample (figure 11). Modifications in "Sample Condition" reference field. We modify the field "required=True" to make mandatory this field (figure 12).

```
368     ReferenceField(
369         'Template',
370         required = True,
371         allowed_types=('ARTemplate',),
372         referenceClass=HoldingReference,
373         relationship='AnalysisRequestARTemplate',
374         mode="rw",
375         read_permission=View,
376         write_permission=FieldEditTemplate,
377         widget=ReferenceWidget(
378             label=_("Sample Template"),
379             description=_("The predefined values of the Sample template are set "
380                           "in the request"),
381             size=20,
382             render_own_label=True,
383             visible={
384                 'add': 'edit',
385                 'secondary': 'disabled',
386             },
387             catalog_name='bika_setup_catalog',
388             base_query={"is_active": True,
389                         "sort_on": "sortable_title",
390                         "sort_order": "ascending"},
391             showOn=True,
392         ),
393     ),
```

Figure 10: *Modification in "Template" reference field to make it mandatory.*

```
845     UIDReferenceField(
846         'SampleCondition',
847         required=1,
848         #multiValued=1,
849         allowed_types='SampleCondition',
850         mode="rw",
851         read_permission=View,
852         write_permission=FieldEditSampleCondition,
853         widget=ReferenceWidget(
854             label=_("Primers"),
855             description=_("Primers to use"),
856             size=20,
857             render_own_label=True,
858             visible={
859                 'add': 'edit',
860                 'secondary': 'disabled',
861             },
862             catalog_name='bika_setup_catalog',
863             base_query={"is_active": True,
864                         "sort_on": "sortable_title",
865                         "sort_order": "ascending"},
866             showOn=True,
867         ),
868     ),
```

Figure 11: *"Sample Condition" reference field modifed to show information about the primers.*

```
698     ReferenceField(
699         'PublicationSpecification',
700         required=True,
701         allowed_types='AnalysisSpec',
702         relationship='AnalysisRequestPublicationSpec',
703         mode="rw",
704         read_permission=View,
705         write_permission=FieldEditPublicationSpecifications,
706         widget=ReferenceWidget(
707             label=_("Publication Specification"),
708             description=_(
709                 "Set the specification to be used before publishing a Sample."),
710             size=20,
711             render_own_label=True,
712             visible={
713                 "add": "invisible",
714                 'secondary': 'disabled',
715             },
716             catalog_name='bika_setup_catalog',
717             base_query={"is_active": True,
718                         "sort_on": "sortable_title",
719                         "sort_order": "ascending"},
720             showOn=True,
721         ),
722     ),
723
```

Figure 12: *Modification in "Specifications" reference field to make it mandatory.*

3.5.2. *senaite.impress* modifications

For adapting the results reports created in Senaite we modify the template "*Default.pt*". This file is set in the next directory: "*home/senaite/buildout-cache/eggs/senaite.impress-1.2.4-py2.7.egg/senaite/impress/templates/reports/Default.pt".*

- Change in "attachments per row = 1" to show only one attached file in the report by default on each row instead of two.
- We add a CSS style class called "break" so that the results appear on multiple lines. This is necessary for results in which the DNA sequence appears, where string can exceeds 300 characters (figure 13).
- We apply this class within the "Results" field of the report (figure 14).
- We modify the dimensions of the results table (figure 15).

- We add "Organism" and "Primers" fields to the "Summary" section of the report (figure 16).

```
108      }
109    </style>
110    <style type="text/css">
111      .break {
112        width: 30px;
113        word-wrap: break-word;
114        word-break:break-all;
115      }
116    </style>
117  </tal:css>
```

Figure 13: *"Break" CCS style class.*

```
437                </td>
438                <td class="text-right break">
439                  <span class="result" tal:content="structure python:model.get_formatted_result(analysis)">23</span>
440                </td>
```

Figure 14: *"Results" field with "break" CSS class.*

```
389            <!-- Analysis in POC and Category -->
390            <table class="table table-sm table-condensed">
391              <colgroup>
392                <!-- Category -->
393                <col style="width: 20%;">
394                <!-- Result -->
395                <col style="width: 50%">
396                <!-- Unit -->
397                <col style="width: 5%">
398                <!-- Range -->
399                <col style="width: 10%">
400                <!-- Out of Range -->
401                <col style="width: 5%">
402              </colgroup>
```

Figure 15: *Result table dimensions.*

```
332             <td class="label" i18n:translate="">Primers</td>
333             <td tal:content="model/SampleCondition/title|nothing"></td>
334           </tr>
335           <tr tal:condition="batch">
336             <td class="label" i18n:translate="">Remarks</td>
337             <td tal:content="model/Remarks"></td>
```

Figure 16: *"Organism" and "Primers" fields.*

3.6. COMPLIANCE WITH REQUIREMENTS

We focused in the requirements showing in the MoSCoW method below (table 11). In table 4 we can see which compliance degree we have obtain in each requirement.

REQUIREMENTS	MuSCoW	COMPLIANCE (%)
Open agile tools	M	100
Open Access and customizable LIMS	M	100
Safety and backup	M	100
Trustworthy	M	100
Allow sample tracking	M	100
Identify who is using the application and what changes have done	M	100
Allow to print reports	S	100
Easy to use	S	75
Connected to sequencing equipment (instrument interfaces)	C	0
Sample automation	W	0

Table 11: *MoSCoW method of prioritization of requirements and compliance degree. M: must have; S: should have; C: could have; W: would not have.*

Open agile tools

We created a JIRA planner to follow up the plannification and GitHub repository to control the versions of the deliverables.

Open Access and customizable LIMS

Senaite-LIMS allows users to track samples, control access, get audit logs and create reports of the analysis done. Also, we can configure the software and even modify its code to adapt it to our needs.

Safety and back-up

Plone allows us to restore the database or make a backup if it is needed to recovery data or versions of our products. Also, both Senate and Plone allows access control and permissions though roles and users management.

Trustworthy

The large amount of information both on the Senaite website and in the various forums make Senaite a trustworthy tool. Also, in the time that we have been using it, we see that it is a very fast and flexible software.

Allow sample tracking

Samples are the main objects not only in Senaite, but also in every LIMS. Samples have unique identifiers. Related to the safety, there are some sample states to validate or refuse results and to publish reports. Each role have permissions to each state, so the

track of the sample is made possible, from the moment it is registered in the system until the results are published.

Identify who is using the application and what changes have done

As we mentioned before, Senaite allows every users access to audit logs in order to know who, what, when and why someone have done something into the system. Also, audit logs are important to identify errors, which is very important to solve incidents quickly.

Allow to print reports

Senaite have reports by default to print the results of the analysis done. But also, we can create new report templates or modify the existing ones through *senaite.impress* add-on to show more information.

Easy to use

This is one of the most difficult requirements to evaluate because it depends on the user. For admin user, it is easy to configure the system but the difficulty comes when installing the system or understanding the Plone and Zope configuration.

Instrument interfaces.

Senaite is prepared to export sample and analysis information in different formats and to import results. Nevertheless, in this work we do not have enough information about the instruments used in a sequencing laboratory and how to implement interfaces between them ad LIMS system.

Sample automation.

Senaite have an automated sample registration option. There is a calendar and users can program how many samples and which sample template must have them. However, we have not configured any automation for samples.

3.7. FUNCTIONAL DEMONSTRATION

In order to understand the functionality of the configured LIMS we detail a use case and a functional demonstration (see Annex II).

4. DISCUSSION

Currently, the principal protocol to track samples and data in most laboratories is based on using spreadsheets like Excel or customize data base like Access. This kind of data management is inefficient and can lead to some errors which are difficult to solve and to manage.

For this reason, to set up and implement a LIMS is a need not only to centralize and manage the laboratory activities or protocols, but also to obtain truthfully and quickly information for the clients. If we think in a sequencing laboratory, which can track big amounts of information, this software can allow technicians and laboratory staff an advantage. In conclusion, this kind of software is essential in a sequencing laboratory to have a sample tracking and control of the information.

There are different LIMS at our disposal, some of them based on licenses and some others based on open source philosophy. In this project we have configured Senaite-LIMS, an open source LIMS adapting it to the necessities and protocols of sequencing laboratories.

The main advantage to use this software is the easy configuration featuring, because the software have all kinds of basic tools that are used in a generic laboratory. Also, there is a great amount of information and support manuals behind that branch, which allowed us to understand the data structure and the logic of the program in a relatively fast way. Furthermore, we have been able to meet the established requirements, only missing those marked as “Should” and “Wouldn’t”.

Nevertheless, several problems appeared during the development of the project. First of all, we have had some difficulties due to a certain lack of technical knowledge. Especially, we found some problems to understand the structure of Plone environment and the structure of add-on files. In fact, Senaite developers recommend creating new add-ons to implement the functionalities required instated of modifying the existing ones in order to updates. In this project, we assume that risks because the changes were minimal and controlled, but new add-ons should be developed if we want to develop more complex functions. In addition, we have similar problems with the report template, but again, the changes were minimal.

On the other hand, it has been difficult to find solid information on real data from sequencers so we have no accomplish the instrument interfaces requirement. However, Senaite allows administrators to configure export and import files to create interfaces between instruments and the system. In addition, Senaite community is developing a new add-on called *senaite.instruments* (24). that create new export and import functionalities. Nonetheless, Senaite also allows users to attach files to the results, which can save functionality in this phase.

Finally, we have to mention that Senaite is developed on Python 2, a version of Python which is out of support since January 2020 (25). However, Senaite developers have been working to upgrade Senaite to the latest Plone 5 in combination with Python 3 (26).

5. CONCLUSIONS

The increase of DNA sequencing practices and data generation requires the use of computerized management tools, which are more powerful than lab notebooks and traditional spreadsheets and worksheets. We have configure the open source Senaite-LIMS tool to manage samples of a sequencing laboratory workflows, using agile software development tools to control requirements, risks, deadlines and deliverables. The use of this data management software can reduce costs and allow sequencing laboratories to track samples, manage and publish results and control data integration. However, some technical programming and good planification skills are required to develop more complex functionalities into the system.

6. GLOSSARY

- 2-D: Two-dimensional.
- 3-D: Three-dimensional.
- ABI: Applied Biosystems.
- bp: Base pair.
- ddNTPs: Dideoxynucleotides.
- DNA: Deoxyribonucleic Acid.
- dNTPs: Deoxynucleotides.
- ELN: Electronic Laboratory Notebook.
- GA: Genome Analyzer.
- Gb: Giga base.
- kb: Kilo base.
- LIMS: Laboratory Information Management Systems.
- Max: Maximum result allowed.
- Max warn: Maximum result warned.
- Min: Minimum result allowed.
- Min warn: Minimum result warned.

- MuSCoW: Must, Should, Could, Wouldn't.
- NA: Not Applicable.
- NGS: Next-Generation Sequencing.
- PCR: Polymerase Chain Reaction.
- PEC: Master programme evaluation activity.
- RNA: Ribonucleic Acid.
- SaaS: software as a Service.
- SBS: Sequencing by Synthesis.
- SOLiD: Sequencing by Oligo Ligation Detection.
- tRNA: Transfer RNA.

7. BIBLIOGRAPHY

1. Kchouk M, Gibrat JF, Elloumi M. Generations of Sequencing Technologies: From First to Next Generation. Biol Med. 2017;09(03).

2. Heather JM, Chain B. The sequence of sequencers: The history of sequencing DNA. Genomics. 2016;107(1):1–8.

3. Sanger F, Nicklen S, Coulson AR. DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci U S A [Internet]. 1977 Dec;74(12):5463–7. Available from: https://pubmed.ncbi.nlm.nih.gov/271968

4. Liu L, Li Y, Li S, Hu N, He Y, Pong R, et al. Comparison of next-generation sequencing systems. J Biomed Biotechnol. 2012;2012.

5. Quail MA, Smith M, Coupland P, Otto TD, Harris SR, Connor TR, et al. A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. BMC Genomics [Internet]. 2012;13(1):1. Available from: BMC Genomics

6. Kahn SD. On the future of genomic data. Science (80-). 2011;331(6018):728–9.

7. Batley J, Edwards D. Genome sequence data: Management, storage, and visualization. Biotechniques. 2009;46(5 SPEC. ISSUE):333–5.

8. Di Resta C, Galbiati S, Carrera P, Ferrari M. Next-generation sequencing approach for the diagnosis of human diseases: Open challenges and new opportunities. Electron J Int Fed Clin Chem Lab Med. 2018;29(1):4–14.

9. Gibbon GA. A brief history of LIMS. Lab Autom Inf Manag. 1996;32(1):1–5.

10. Wood S. Comprehensive Laboratory Informatics : A Multilayer Approach. 2007; (September).

11. Standardization. IOBP (OBP). IO for. “ISO/IEC 17025:2017(en) - General requirements for the competence of testing and calibration laboratories”. 2017.

12. LIMSWiki. Laboratory information management system [Internet]. 2011 [cited 2020 Nov 2]. Available from: https://www.limswiki.org/index.php/Laboratory_information_management_system

13. Naralabs. SENAITE, Professional Open Source LIMS. The evolution of Bika LIMS. [Internet]. 2018 [cited 2020 Nov 11]. Available from: https://naralabs.com/en/blog/senaite-professional-open-source-lims-the-evolution-of-bika-lims

14. Senaite. Senaite LIMS features. [Internet]. [cited 2020 Nov 25]. Available from: https://www.senaite.com/features

15. Bika community. Bika Open Source LIMS. [Internet]. 2018 [cited 2021 Jan 4]. Available from: https://www.bikalims.org/

16. Atlassian. Jira software. [Internet]. 2017 [cited 2020 Oct 10]. Available from: https://www.atlassian.com/software/jira

17. Rosselló Villán V. Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa. [Internet]. Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa. 2019 [cited 2020 Jan 4]. Available from: https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/

18. Senaite. Senaite LIMS installation [Internet]. [cited 2020 Oct 10]. Available from: https://www.senaite.com/docs/installation

19. Plone community. Plone. [Internet]. 2016 [cited 2020 Nov 12]. Available from: https://github.com/plone/plone.docker

20. Servicio de Genómica de la Universidad Complutense de Madrid. Genómica y Proteómica [Internet]. [cited 2020 Oct 5]. Available from: https://www.ucm.es/gyp/genomica-1

21. Laboratorio Secuenciación IdiPAZ. Guía práctica para el usuario de Secuenciación Sanger IdiPAZ [Internet]. [cited 2020 Oct 10]. Available from: https://www.idipaz.es/PaginaDinamica.aspx?IdPag=194&Lang=ES

22. Servicio de Secuenciación de la Universidad de Salamanca. Secuenciación de ADN [Internet]. [cited 2020 Oct 5]. Available from: https://nucleus.usal.es/es/secuenciacion/servicios

23. Macrogen. Standard Sequencing [Internet]. [cited 2020 Oct 5]. Available from: https://www.macrogen.com/en/business/ces_service1.php

24. Senaite. Instrument results import and export add-on for SENAITE. [Internet]. 2020 [cited 2021 Jan 2]. Available from: https://github.com/senaite/senaite.instruments

25. Python Community. Sunsetting Python 2. [Internet]. 2020 [cited 2021 Jan 2]. Available from: https://www.python.org/doc/sunset-python-2/#:~:text=We have decided that January,a security problem in it.

26. Senaite. P8: Plone 5 + Python 3 Upgrade Guide. [Internet]. 2019 [cited 2021 Jan 2]. Available from: https://github.com/senaite/senaite.core/blob/master/P8_UPGRADE_GUIDE.md

8. ANNEXES

8.1. ANNEX I: GANTT DIAGRAM

We have defined the priority tasks in the same colours as tasks, so we can adjust the times and focus in mandatory ones. Also, milestones have been included in dark blue and phases in clear blue (figure 17).

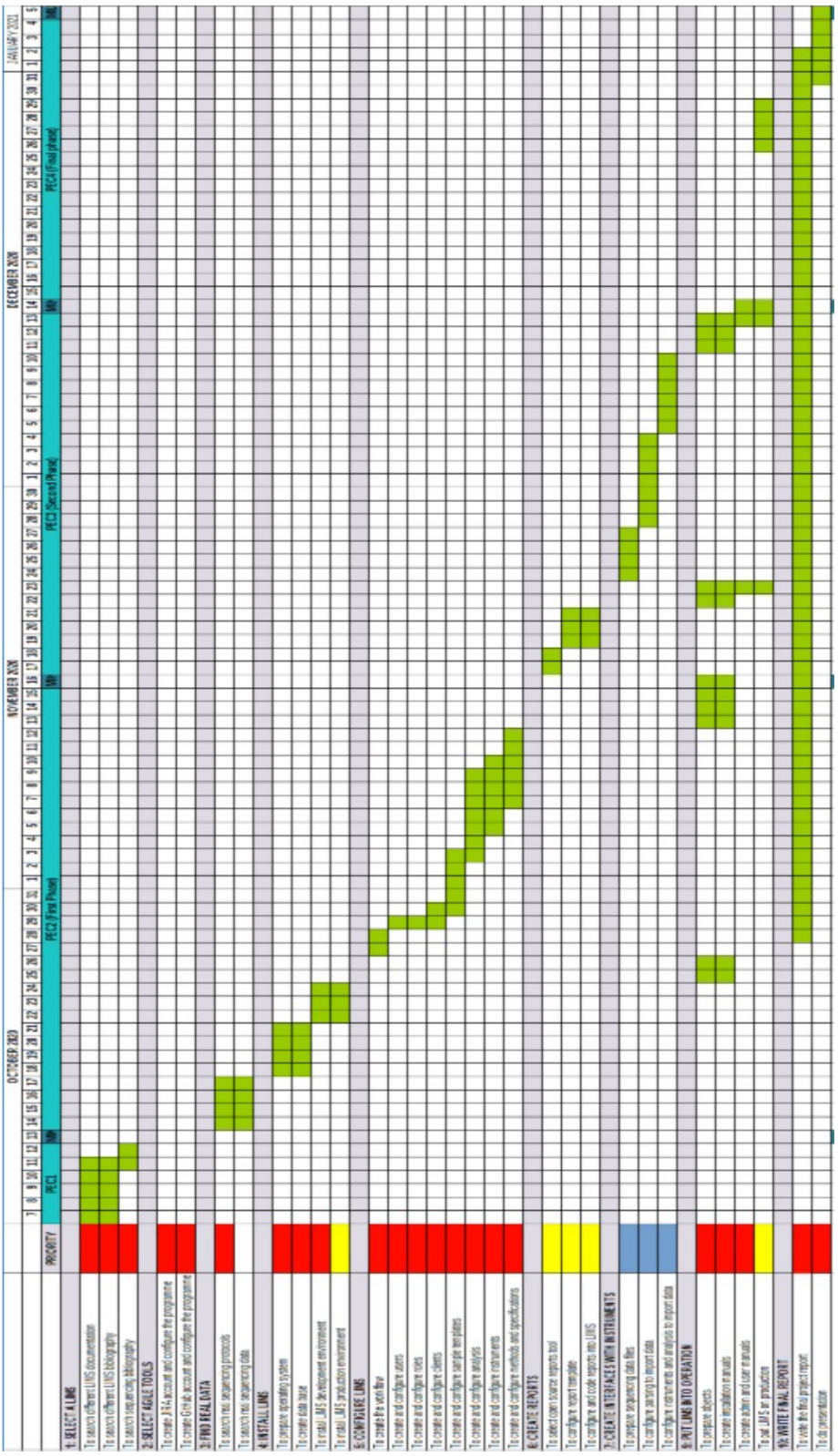


Figure 17: *Gantt diagram in Excel.*

8.2. ANNEX II: INSTALLATION GUIDE

We detail the steps to install Senaite-LIMS in our Ubuntu OS.

Create New User

We create a user called *senaite* in our OS terminal with the next command:

$ sudo adduser --home /home/senaite --shell /bin/bash senaite

After create this user, we have to give ot it privileges, so we modified *sudoers.r* file (figure 18).

```
#includedir /etc/sudoers.d
#Usuario actual
root ALL=(ALL:ALL) ALL
senaite ALL=(ALL:ALL) ALL
```

Figure 18: *sudoers.r file information.*

Miniconda download and installation

We download Python 2.7 version for our OS and create a new Python environment with conda:

$ wget https://repo.anaconda.com/miniconda/Miniconda2-latest-Linux-x86_64.sh

$ bash /home/senaite/Miniconda2-latest-Linux-x86_64.sh

$ source /home/senaite/.bashrc

$ conda create --name senaite python=2.7

$ conda activate senaite

System dependencies installation

Senaite requires to install some dependencies:

$ sudo apt install build-essential

$ sudo apt install python2.7 python2.7-dev

$ sudo apt install libxml2 libxml2-dev libxslt1.1 libxslt1-dev

$ sudo apt install libffi-dev libcairo2 libpango-1.0-0 libgdk-pixbuf2.0-0 libpangocairo-1.0-0 libgdk-pixbuf2.0-0

$ sudo apt install zlib1g zlib1g-dev

Plone download and installation

We download Plone 4.3.19 unified installer, that installs Plone and its dependencies for Ubuntu:

$ wget --no-check-certificate https://launchpad.net/plone/4.3/4.3.19/+download/Plone-4.3.19-UnifiedInstaller.tgz

$ tar xzf Plone-4.3.19-UnifiedInstaller.tgz

$ cd Plone-4.3.19-UnifiedInstaller

$ ./install.sh standalone --target=/home/senaite --instance=senaitelims –password=admin

Senaite installation

First of all, we have to modifiy *buildout.cfg* file, which is an automation tool written in Python.

$ cd /home/senaite/senaitelims

$ vim buildout.cfg

Inside this file, we have to add the add-ons to be installed, in our case, we add *senaite.lims, senaite.json, senaite.impress* and *senaite.core* (figure 19).

```
72 eggs =
73     Plone
74     Pillow
75     senaite.lims
76     simplejson
77     senaite.core
78     senaite.jsonapi
79     senaite.impress
```

Figure 19: *buildout.cfg file with senaite add-ons (eggs).*

We also have to modify the version section, as we indicate on figure 20.

```
169 ##############################################
170 # Versions Specification
171 # ----------------------
172 # Version information supplied here will "pin" Python packages to a particular
173 # version number, even when you use the "newest" flag running buildout.
174 # Specifying versions for all packages is a good idea and can prevent
175 # accidental changes when you add new packages to your buildout.
176 # Note that versions specified here will override those specified earlier
177 # in the configuration, including those from the Plone and Zope version
178 # config files.
179 #
180 [versions]
181 zc.buildout =
182 setuptools =
183 Pillow = 5.1.0
184 cssselect2 = 0.2.2
185 soupsieve = 1.9.5
186
187 buildout.sanitycheck = 1.0.2
188 collective.recipe.backup = 4.0
189 plone.recipe.unifiedinstaller = 4.3.2
```

Figure 20: *buildout.cfg file versions section.*

Finally, we have to create a requirements file (*requirements.txt*) to ensure that *pip, setuptools* and *zc.buildout* are availabile in a compatible version:

$ cd /home/senaite/senaitelims

$ cat << EOF > requirements.txt

setuptools==39.2.0

zc.buildout==2.13.2

pip==19.3.1

EOF

8.3. ANNEX III: FUNCTIONAL DEMONSTRATION

Starting Senaite

To start Senaite in foreground mode we have to use the next commands, which have been added to our *senaite_lims_start.sh*:

$ cd homesenaite/senaitelims

$ bin/instance fg

We can also directly execute the file itself, which has been saved on *"home/senaite/senaitelims/desktop"* directory. After starting Senaite instance, we can clic on Senaite desktop shortcut.

Log into Senaite

First of all, we have to log in to the system with *Laboratory Clerk* credentials (figure 21).

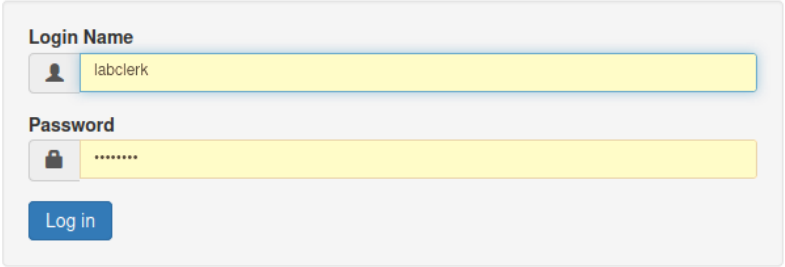


Figure 21: *Log-in window. See section 3.4.1 to see the credentials.*

The next step is to create samples. For that, we are going to create a batch for a client first. We must click on “Batches” tab and in “Add” button to create a new batch. Then, we must complete some information and click on “Save”. In the figure 22 we show an example with the information of a new batch for one of the clients (*Public Health Institute*) for a drug resistance study.

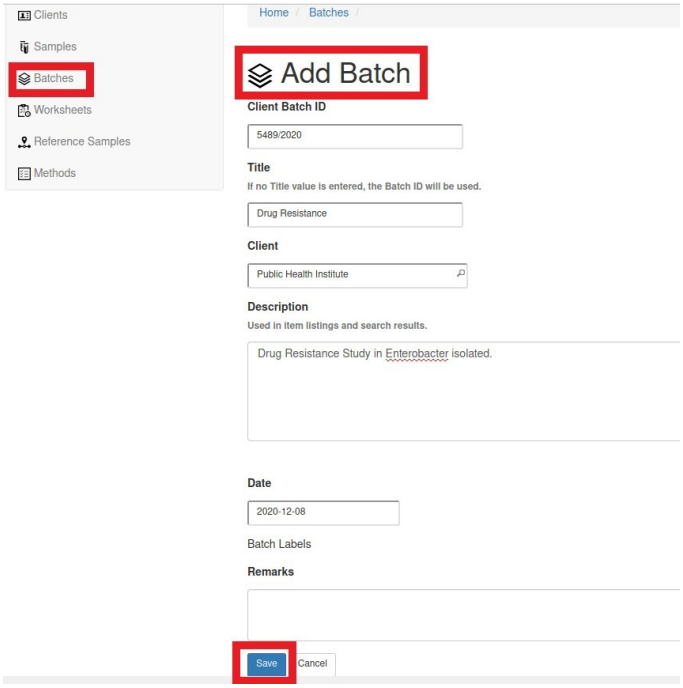


Figure 22: *“Batches” window.*

Once we have created the new batch, we will be able to watch its information and to create samples related to it. Click on “Samples” tab and “Add” some samples (figure 23).

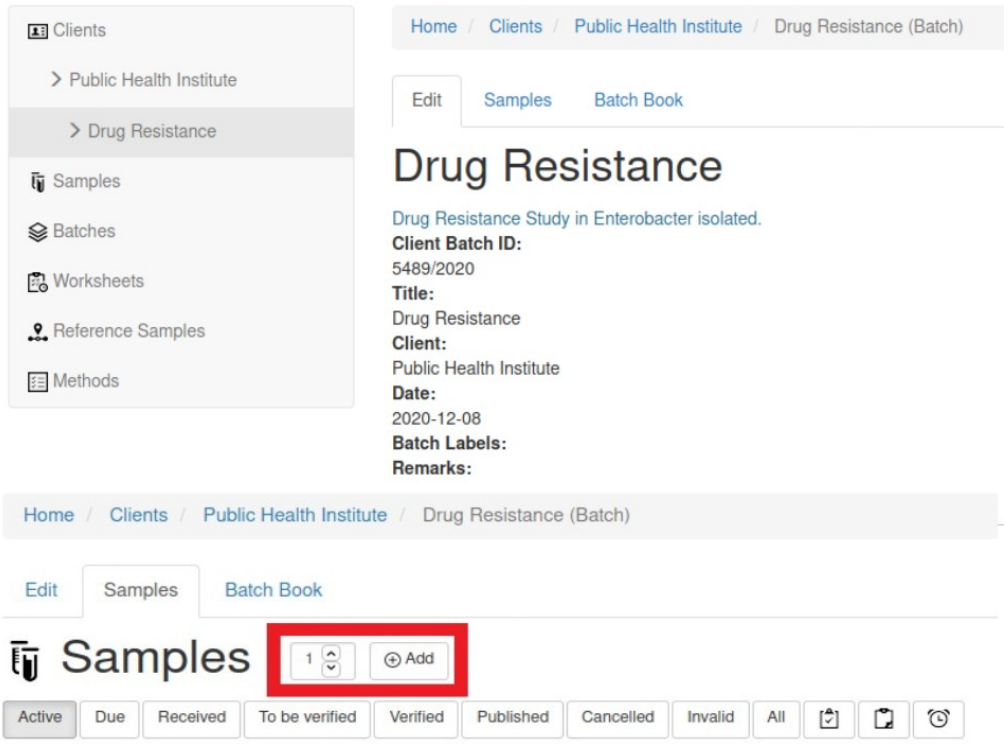


Figure 23: *Add samples window thorugh a batch.*

We have selected 3 samples and a new window have opened. The window bellow is the *Analysis Request* (figure 24), which is the sample registration template. We can select the information of the samples in this window. There are some fields that are mandatory. In this example, we are going to create 3 samples of Tissue.

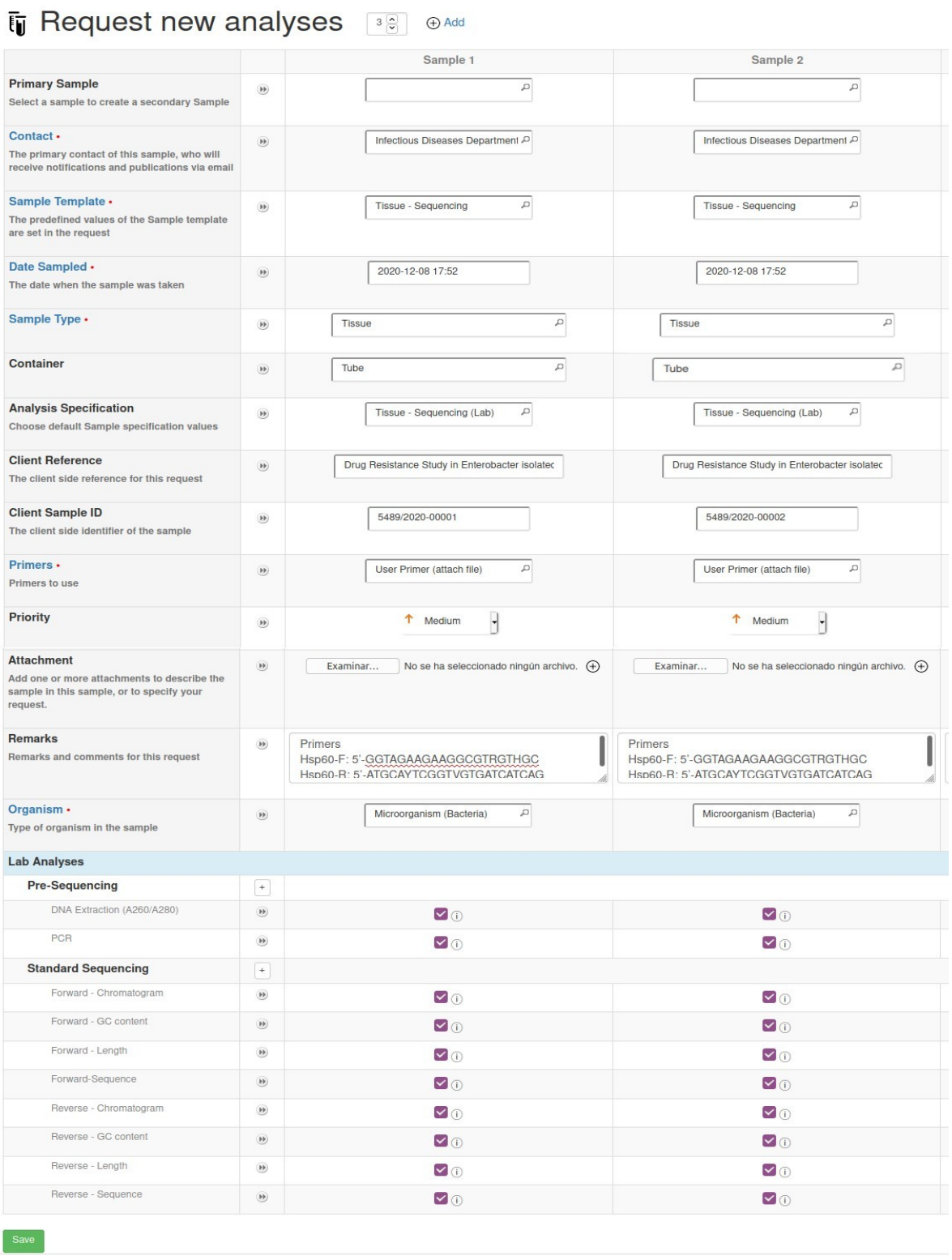


Figure 24: *Analysis request window.*

We can appreciate that all analysis are selected by default because *Tissue* samples include all of them. Nevertheless, if we select other sample type, analysis will be change. We click on "Save" and the three samples will be created inside the batch. The batch will be in "Open" state and the samples in "Received" state (figure 25). We can click on each sample to see or modify the information.

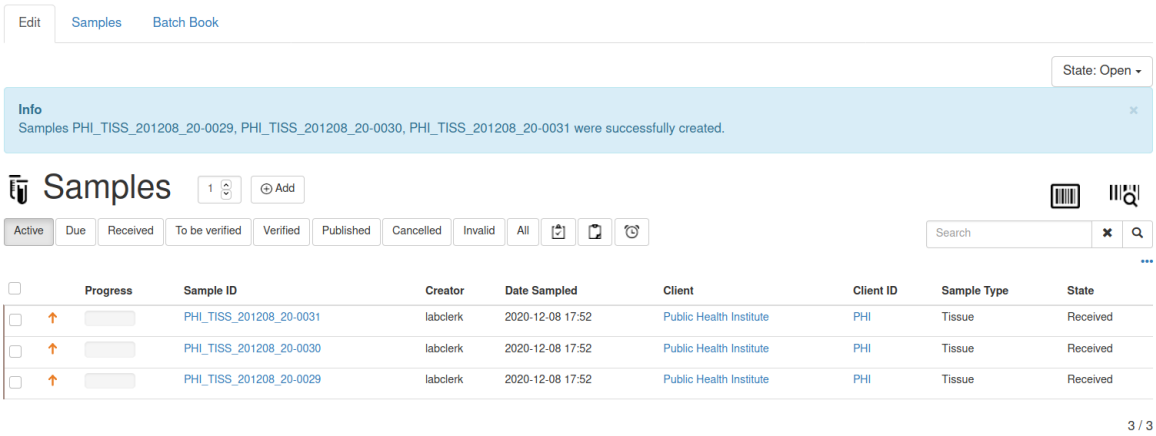


Figure 25: *Sample information into the batch window.*

The next step is to create worksheets and to assign them to the analysts. So, we click on "Worksheets", add a new one and assign it to *labanalyst* (figure 26).

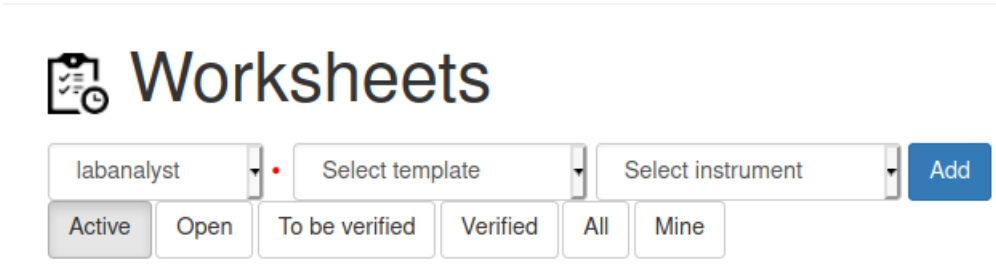


Figure 26: *Worksheet assignation.*

A new worksheet has been created and we can add some analysis to it. In this first worksheet we are going to add the *pre-sequencing* analysis of the three samples created before. So, we can filter the analysis writing “Pre-Sequencing” on the searcher to select the analysis of DNA Extraction and PCR of the samples and click on “Assign”. This six analysis will be assigned to *labanalyst* (figure 27).

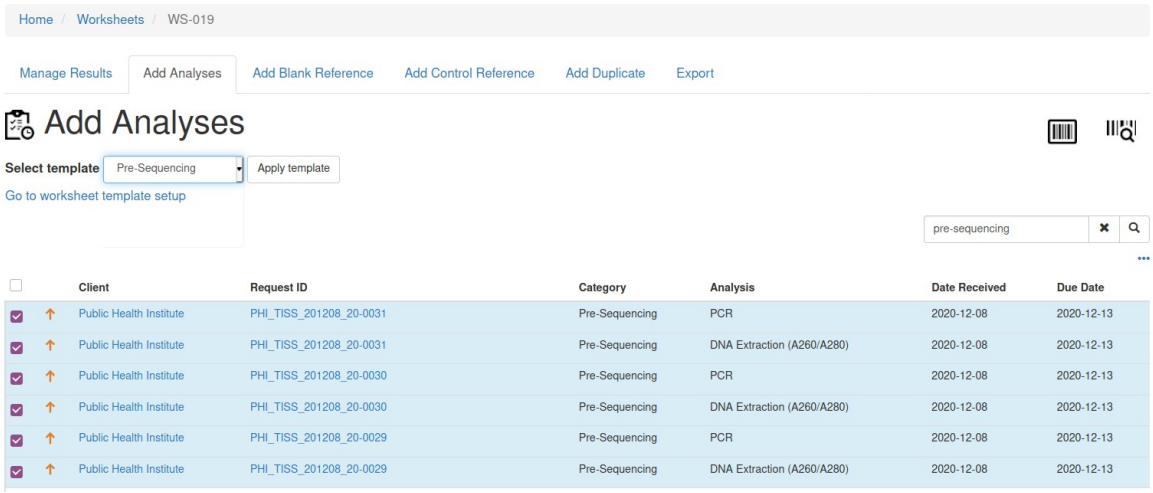


Figure 27: *Analysis added to the worksheet.*

We also are going to add a blank reference sample as negative control using *Destiled Water* (figure 28).

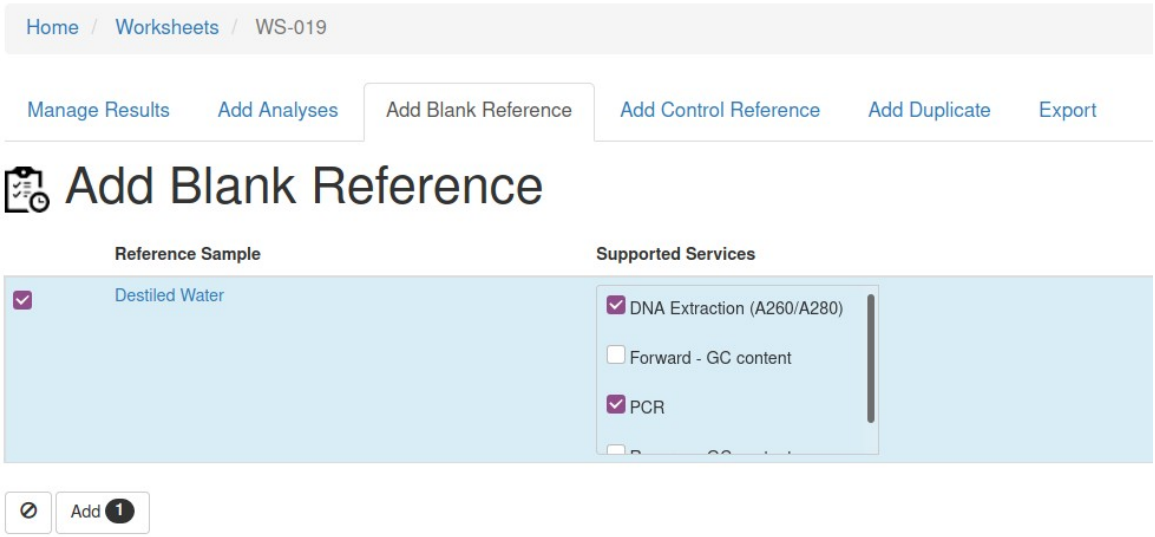


Figure 28: *Adding a blank reference sample to the worksheet.*

After that, we have a total of eight analysis in this worksheet, six of our samples and two of the blank sample.

At this point, we log out and log in as *labanalyst*. We click on "Worksheet" and select the one that we have created in the step before. We are ready to enter results. As we can see, the limits are established by the specifications (figure 29).

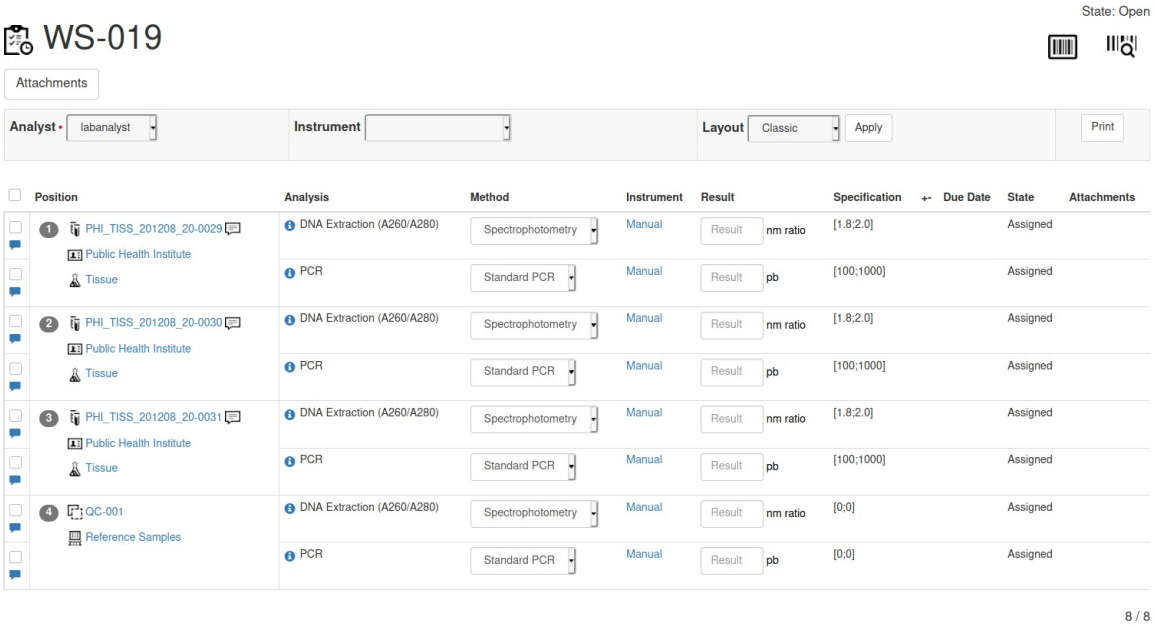


Figure 29: *Entering results to the analysis service.*

If the result introduced is out of range or out of shoulder range Senaite will display an icon with a message (figure 30).

Figure 30: *Results out of range (up) or out of shoulder range (down).*

After entering the results, first we must click on “Save” and the analyst may change them if necessary (writing the modification in to the audit log). If the analyst is sure about the results he/she can click on “Submit” (figure 31) but results can not be modified (only by *labman* user).

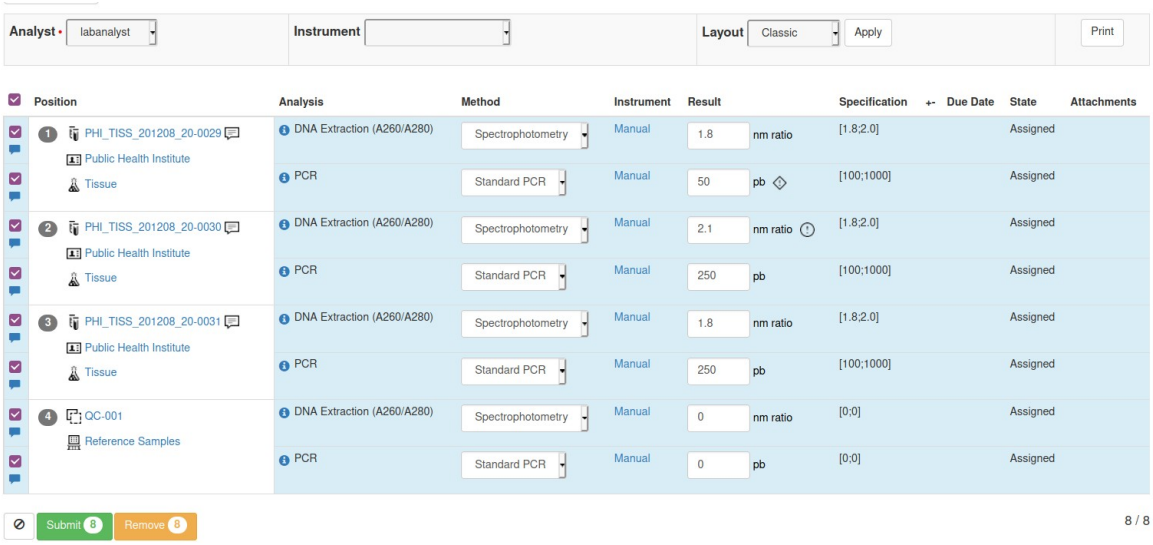


Figure 31: *Submitting results by labanalyst.*

We are going to repeat the same steps but now we create a new worsheet with *labclerk* (W-020), including *Sequencing* analysis services and assigning them to *labanalyst2* user. Also, we add a Blank Reference sample too, so a total of 28 analysis will be displayed on this worksheet, 24 related to the samples and 4 to blank control (figure 32).

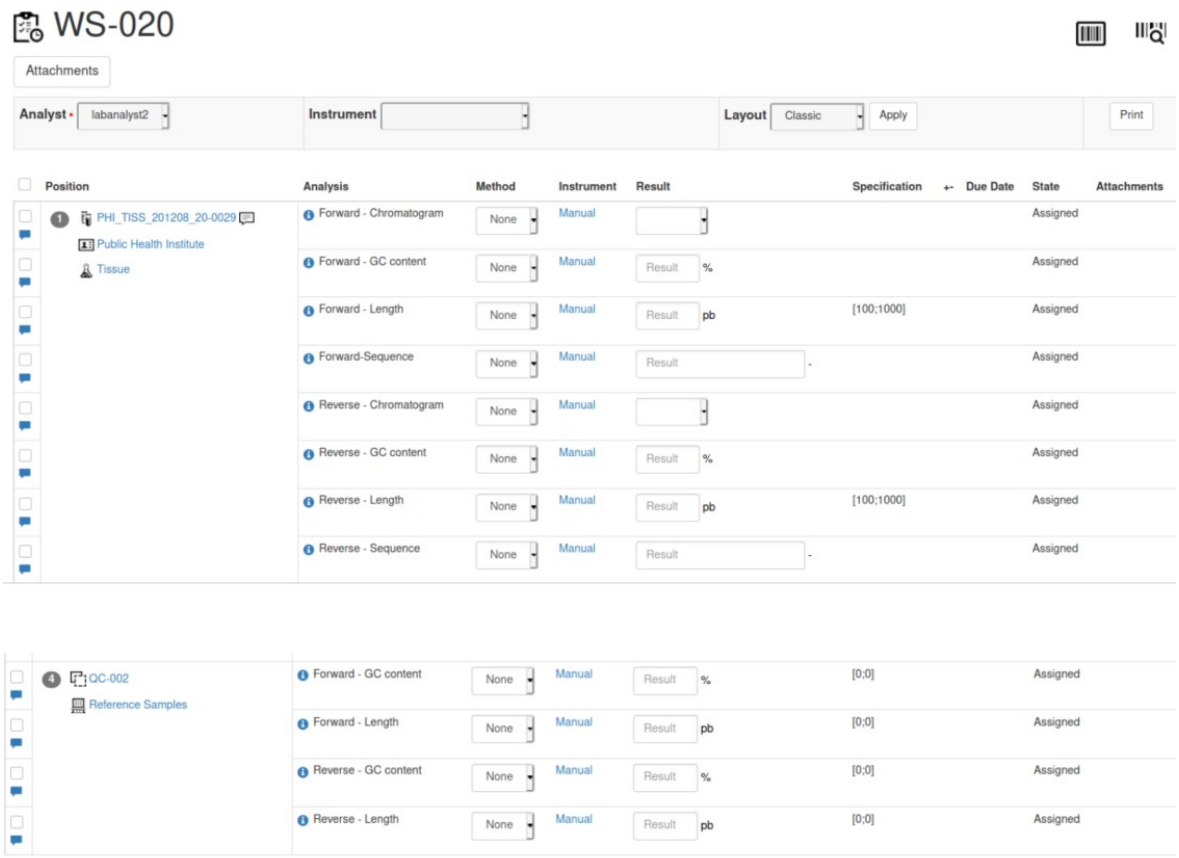


Figure 32: *New worksheet created including Sequencing analysis services.*

Into this new Worksheet, we must attach a file for *Forward-Chromatogram* and *Reverse-Chromatogram* services. The way to do this is clicking on "Attachments". Then, we select a file and the analysis were we want to add it (figure 33).

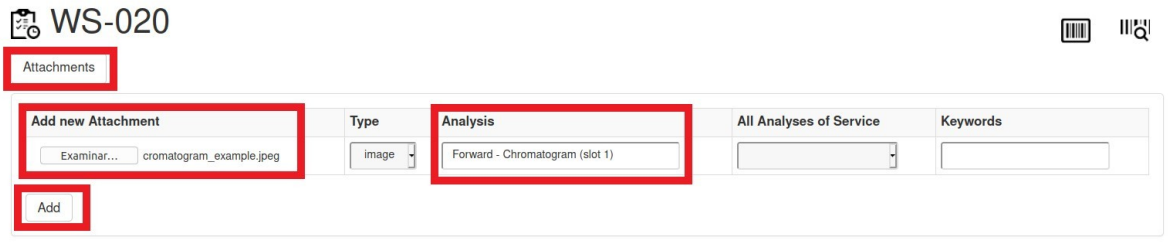


Figure 33: ***F****ile attachment.*

We can check that the attachments have been adding to the correct analysis (figure 34).

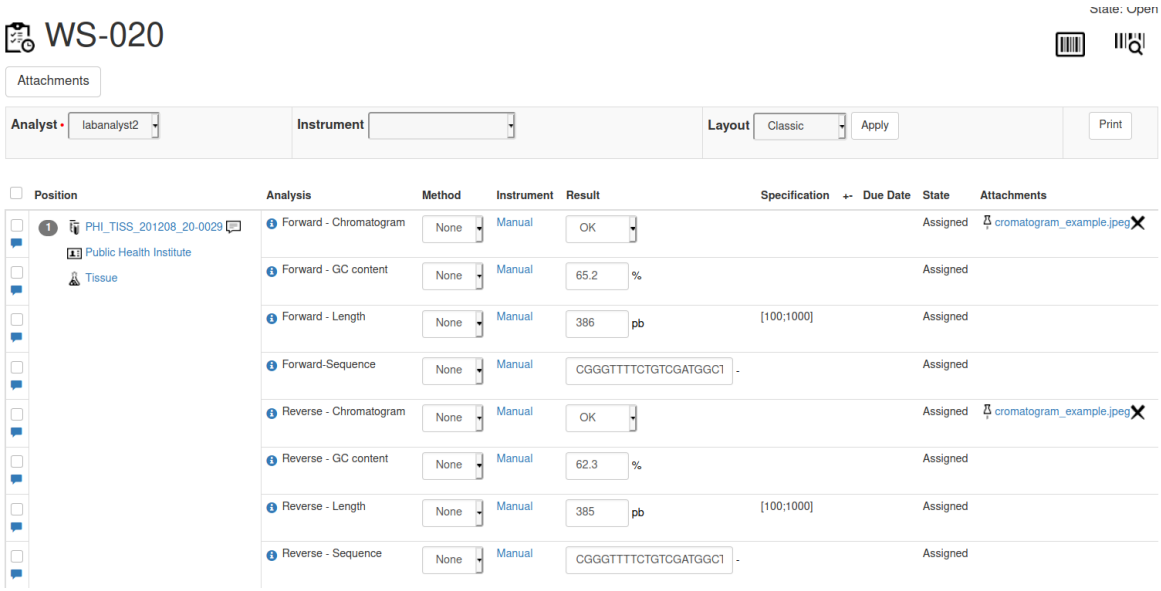


Figure 34: *Files attached to the analysis.*

After enter the results and the attachments, we click on “Submit”.

Publish results

Once *labanalyst* and *labanalyst2* have been entered all the results in their worksheets, we log in to Senaite as *labclerck* again to verify the results. We click on “Batches” and select the batch we created before. The analysis will be verified (figure 35).

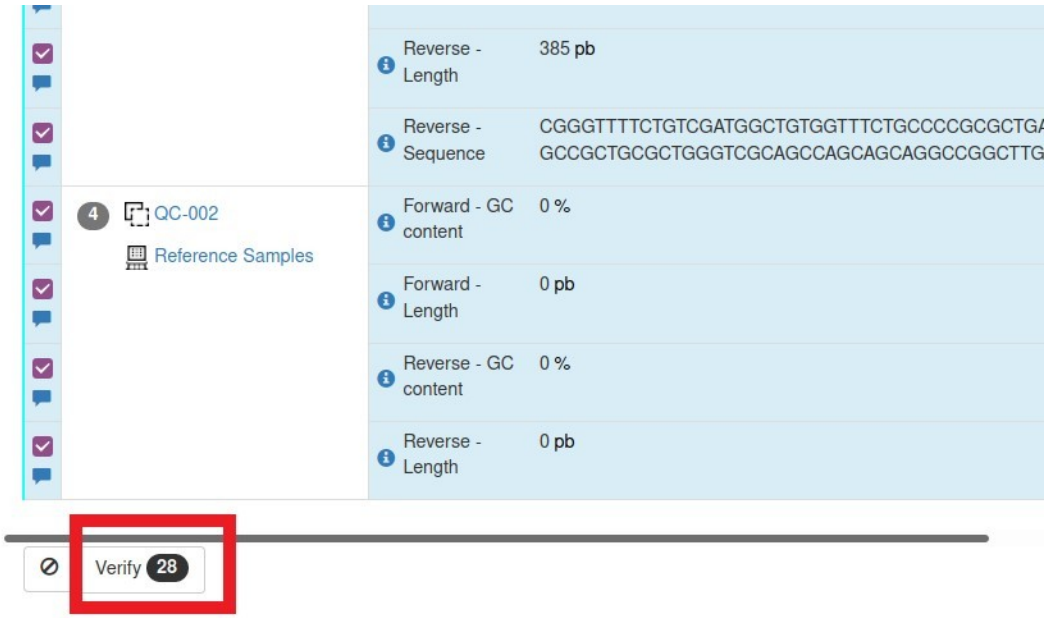


Figure 35: *Results verification.*

If we click on “Worksheets” we can see that the two worksheets created before will change their state to “Verified” (100% of progress).

After that, log in to Senaite as *labmanager* and click on “Batches”. We can see our batch in a 94% of progress. If we click on “Samples” inside the batch we can see the three samples with a 95% of progress and verified. We can see the results and information clicking on each one.

The final step is to publish the results. We can select the three samples and click on “Publish” (figure 36). We also can click on “Invalidate” if the results are incorrect. In this case, a repeat sample will be created and it will be mark with and “R” in the ID.

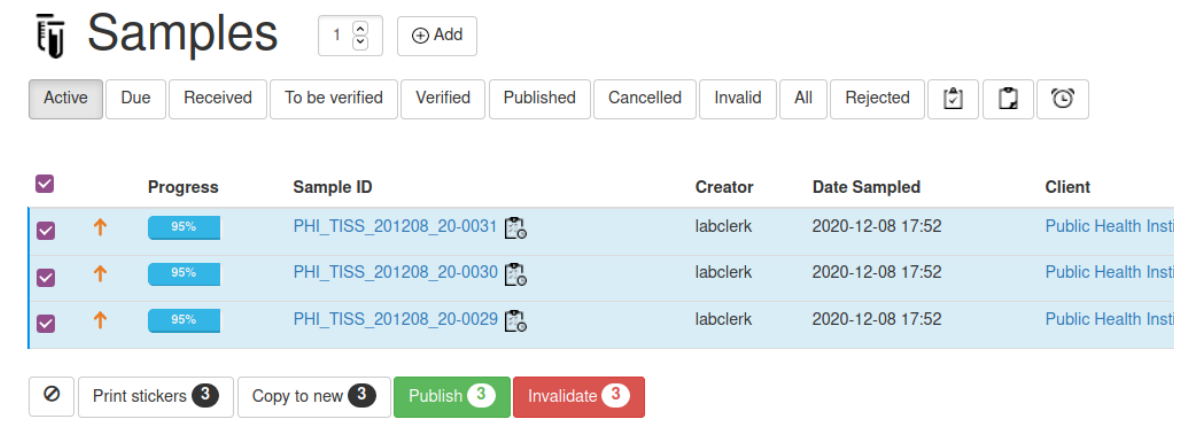


Figure 36: *Publishing the results.*

Click on "Publish". A report will appear.

Report

The first window to appear shows information about the samples and their states and client information. Also, we can choose dimensions and configuration of the report (landscape or portrait) and attachments per row (figure 37).

Figure 37: *Report configuration.*

After the configuration options the report will appear. Information about our laboratory and the client will be showed on the first page (figure 38).


Analysis Report

SENAITE LIMS

Public Health Institute
Infectious Diseases Department
infectuius.diseases@phi.com

Sequencing Laboratory
Supervisor : Laboratory Manager
Spain

Figure 38: *First page of the report.*

On second page of the report we can get some information about the sample (figure 39).

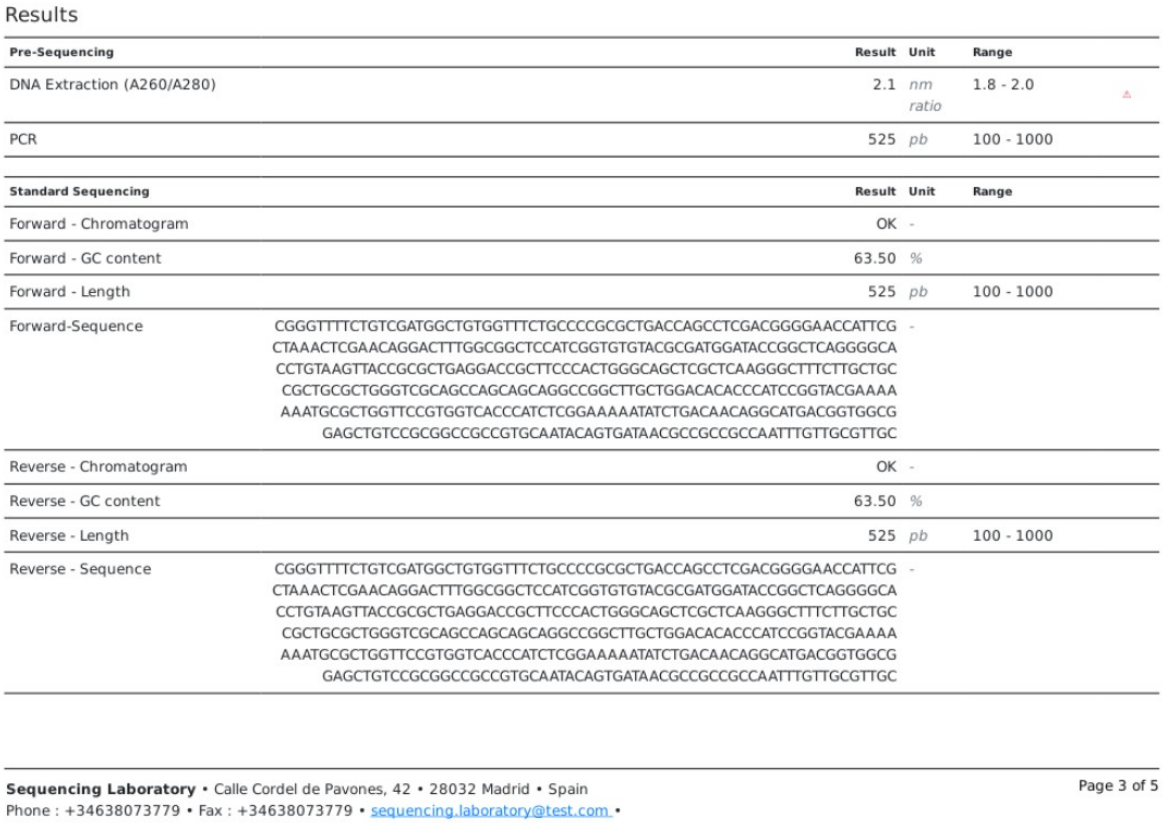

Summary

PHI_TISS_201208_20-0031-R01

Sample ID	PHI_TISS_201208_20-0031-R01
Batch ID	B-009
Client Batch ID	5489/2020
Client	Public Health Institute
Client SID	5489/2020-00003
Sample Type	Tissue
Organism	Microorganism (Bacteria)
Specification	Tissue - Sequencing (Lab)
Primers	User Primer (attach file)
Specification	Tissue - Sequencing (Lab)
Date Received	2020-12-08 18:23
Date Verified	2020-12-09 11:13
Date Published	2020-12-13 17:48
Published by	labman (labman@test.com)

Figure 39: *Second page of the report.*

In the next pages we find information about the results and specifications (figure 40). Also, we can check if the result is out of range.



Results

Pre-Sequencing	Result	Unit	Range
DNA Extraction (A260/A280)	2.1	*nm ratio*	1.8 - 2.0
PCR	525	*pb*	100 - 1000

Standard Sequencing	Result	Unit	Range
Forward - Chromatogram	OK	-	
Forward - GC content	63.50	%	
Forward - Length	525	*pb*	100 - 1000
Forward-Sequence	CGGGTTTTCTGTCGATGGCTGTGGTTTCTGCCCCGCGCTGACCAGCCTCGACGGGGAACCATTCG CTAAACTCGAACAGGACTTTGGCGGCTCCATCGGTGTGTACGCGATGGATACCGGCTCAGGGGCA CCTGTAAGTTACCGCGCTGAGGACCGCTTCCCACTGGGCAGCTCGCTCAAGGGCTTTCTTGCTGC CGCTGCGCTGGGTCGCAGCCAGCAGCAGGCCGGCTTGCTGGACACACCCATCCGGTACGAAAA AAATGCGCTGGTTCCGTGGTCACCCATCTCGGAAAAATATCTGACAACAGGCATGACGGTGGCG GAGCTGTCCGCGGCCGCCGTGCAATACAGTGATAACGCCGCCGCCAATTTGTTGCGTTGC	-	
Reverse - Chromatogram	OK	-	
Reverse - GC content	63.50	%	
Reverse - Length	525	*pb*	100 - 1000
Reverse - Sequence	CGGGTTTTCTGTCGATGGCTGTGGTTTCTGCCCCGCGCTGACCAGCCTCGACGGGGAACCATTCG CTAAACTCGAACAGGACTTTGGCGGCTCCATCGGTGTGTACGCGATGGATACCGGCTCAGGGGCA CCTGTAAGTTACCGCGCTGAGGACCGCTTCCCACTGGGCAGCTCGCTCAAGGGCTTTCTTGCTGC CGCTGCGCTGGGTCGCAGCCAGCAGCAGGCCGGCTTGCTGGACACACCCATCCGGTACGAAAA AAATGCGCTGGTTCCGTGGTCACCCATCTCGGAAAAATATCTGACAACAGGCATGACGGTGGCG GAGCTGTCCGCGGCCGCCGTGCAATACAGTGATAACGCCGCCGCCAATTTGTTGCGTTGC	-	

Sequencing Laboratory • Calle Cordel de Pavones, 42 • 28032 Madrid • Spain
Phone : +34638073779 • Fax : +34638073779 • sequencing.laboratory@test.com •

Figure 40: *Result information.*

In the next pages we find the attachments (figure 41).

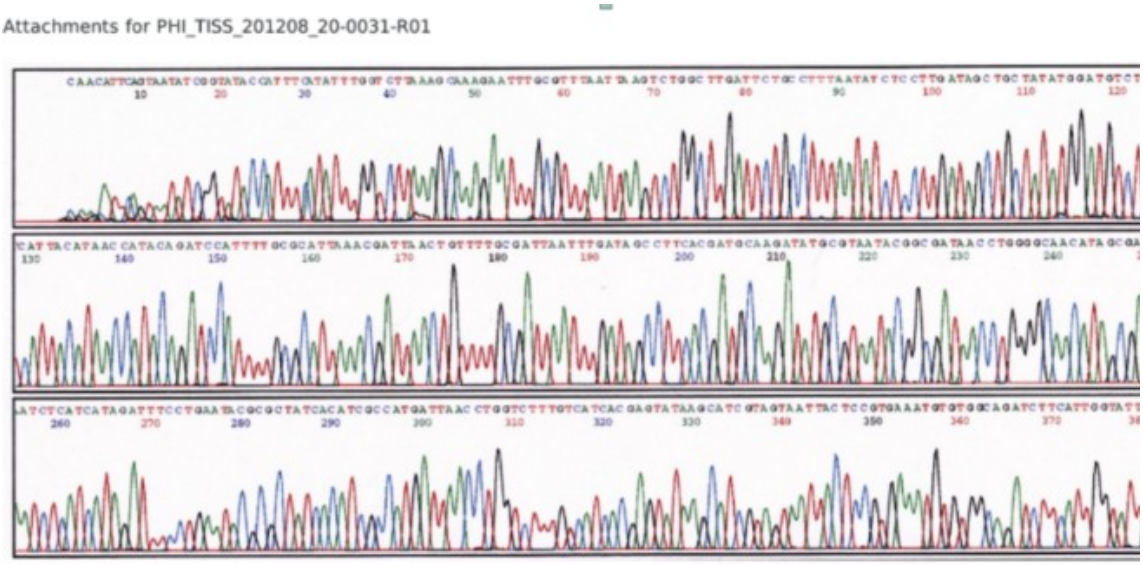


Figure 41: *Attachment information.*

Finally, the last page shows the sign of the laboratory responsible and publisher (figure 42).

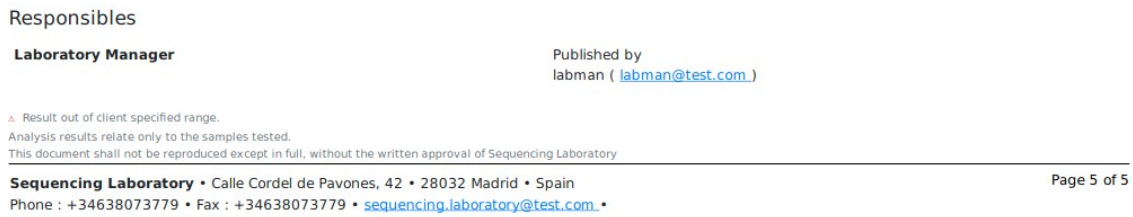


Figure 42: *Information about laboratory responsibles and the publisher.*

Once we have check the preview of the report we can click on “Save” button and the three reports will be saved, one per sample. From “Analysis Report” tab we select the three samples and click on “Publish”. This step is mandatory to change the state of the samples to “Published” and 100% of progress. We can also download the PDF (figure 43).

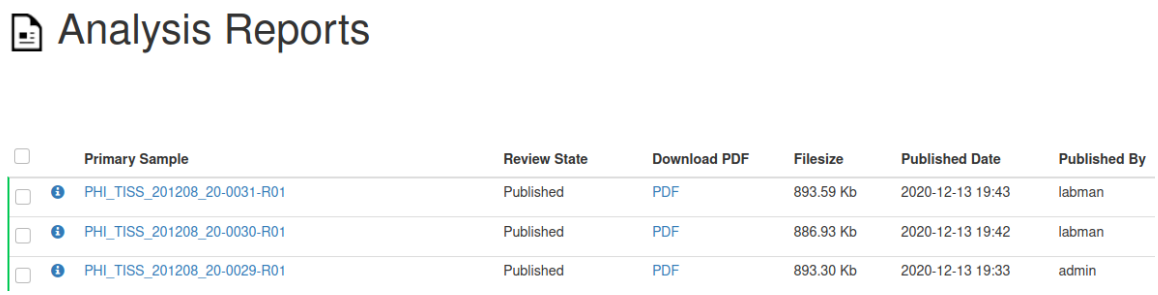


Figure 43: *Analysis Reports window to publish the samples.*

Finally, we must close the bacth created. Click on "Batches" and select the batch created before and click on "Close" (figure 44). The batch will be closed now.

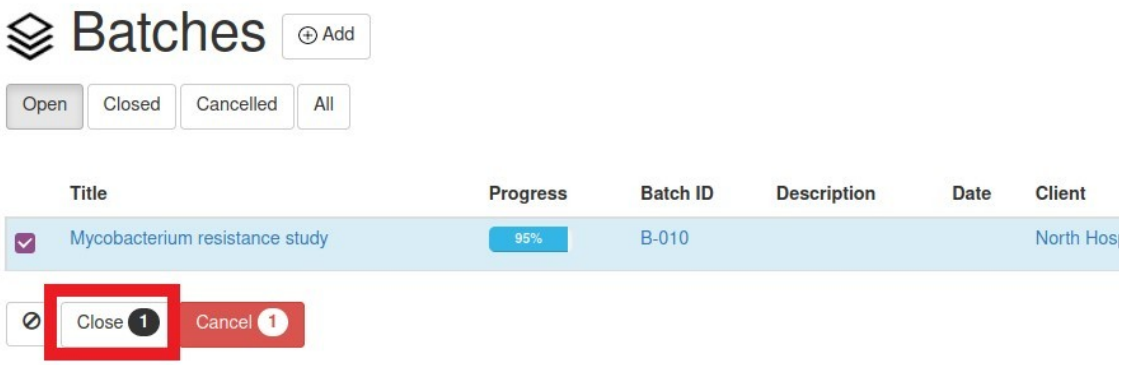


Figure 44: *Closing the batch.*

Audit log

The final step we want to mention is how to see audit log of a sample. We can click on the sample and click on *Audit Log* tab to see not only the different changes of the sample, but also the date and time and who have done the modifications (figure 45). This functionality is available for all the items of Senaite.

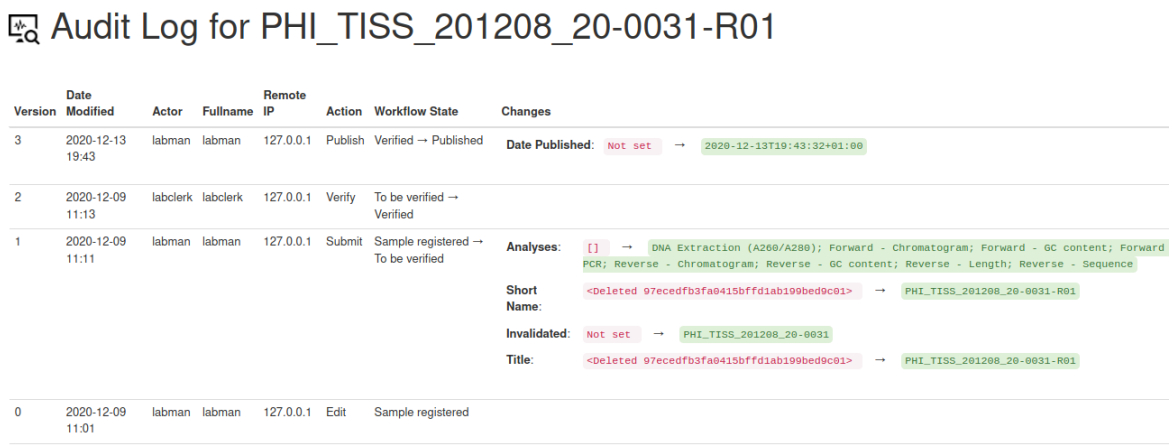


Figure 45: *Example of sample Audit Log.*