ULPGC
Universidad de
Las Palmas de
Gran Canaria

Arquitectura de datos para la caracterización y análisis de microorganismos multirresistentes

Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:
Omar Verona Kämpfer
Tutores:
José Juan Hernández Cabrera
José Évora Gómez
Francisco Chamizo López

Trabajo de Fin de Grado

Arquitectura de datos para la caracterización y análisis de microorganismos multirresistentes

Autor
Omar Verona Kämpfer

Tutores
José Juan Hernández Cabrera
José Évora Gómez
Francisco Chamizo López

Grado en Ingeniería Informática

Las Palmas, 1 de Julio de 2020

Agradecimientos

A mi madre, sin su apoyo hubiera sido imposible haber llegado a donde estoy ahora, a pesar de mis dudas y mi oscilante trayectoria ella siempre estuvo ahí.

A mi pareja, por su paciencia y ayuda en cada tropiezo, la persona que me daba luz cuando todo eran sombras.

A mis tutores, por brindarme semejante oportunidad, permitiéndome conciliar dos de mis pasiones, la informática y la medicina.

Gracias.

Resumen

Objetivos

En el ámbito sanitario existen una serie de necesidades bioinformáticas insatisfechas, el volumen de los datos sigue creciendo y las plataformas existentes son incapaces de extraer suficiente valor de éstos. El objetivo de este trabajo es valorar los procesos del laboratorio de microbiología del Hospital Universitario de Gran Canaria Doctor Negrín, desarrollar una plataforma de datos adecuada y una aplicación para la generación de informes de vigilancia epidemiológica.

Metodología

Para poder estudiar el análisis de los procesos del laboratorio se realizan visitas al laboratorio y entrevistas o discusiones con miembros del mismo. Además, se revisa la documentación aportada por el laboratorio, las aplicaciones informáticas en uso y la estructura de los datos. Durante la elaboración de la plataforma de datos se plantea una arquitectura de Big Data cimentada en una ontología de eventos (event sourcing) desarrollada a partir del análisis anterior. Esta plataforma se realiza en Java con el framework Intino, que sigue los principios del Model Driven Engineering (MDE) para el desarrollo de software. Durante la creación de la aplicación para la generación de informes se hizo uso de Python y librerias comunes en el análisis de datos como Pandas o NumPy, además de PyQt y QtCreator para el desarrollo de la interfaz de usuario.

Resultados

Se ha desarrollado una plataforma de datos preliminar capaz de importar datos a partir de la plataforma antigua y de fácil integración con futuros módulos de análisis de datos o módulos de integración con herramientas bioinformáticas ya existentes.

Junto con esta plataforma se completó la elaboración de un generador de informes de sensibilidad acumulada para el laboratorio de microbiología.

Conclusiones

El volumen creciente de datos y la necesidad de uso de herramientas bioinformáticas no hechas a medida requiere de plataformas de datos flexibles. Éstas deben ser capaces de gestionar adecuadamente la información y facilitar la integración con otras plataformas o aplicaciones. Existe un costo sanitario muy grande derivado del análisis manual de los datos. La ineficiencia en el análisis de éstos, cuyo valor es caduco, supone un coste tanto para la salud de los pacientes como para los organismos sanitarios. El potencial para un análisis rápido y una integración eficaz con herramientas bioinformáticas puede aliviar enormemente este coste y aportar más valor del que se cabría esperar.

Abstract

Aims

In the health field there are a number of unmet bioinformatics needs, the volume of data continues to grow and existing platforms are unable to extract sufficient value from the data. The aim of this work is to evaluate the processes of the microbiology laboratory of the Hospital Universitario de Gran Canaria Doctor Negrín, to develop a suitable data platform and an application for the generation of epidemiological surveillance reports.

Methodology

In order to study the analysis of the laboratory processes, visits to the laboratory and interviews or discussions with members of the laboratory are carried out. In addition, the documentation provided by the laboratory, the computer applications in use and the structure of the data are reviewed. During the elaboration of the data platform, a Big Data architecture is proposed, based on an event ontology (event sourcing) developed from the previous analysis. This platform is made in Java with the Intino framework, which follows the principles of MDE for software development. During the creation of the application for report generation, Python and common libraries for data analysis such as Pandas or NumPy were used, as well as PyQt and QtCreator for the development of the user interface.

Results and findings

A preliminary data platform capable of importing data from the old platform and easily integrated with future data analysis modules or integration modules for existing bioinformatics tools. Together with this platform, the development of a cumulative sensitivity report generator for the microbiology laboratory was completed.

Conclusions

The increasing volume of data and the need for the use of non tailor-made bioinformatic tools requires flexible data platforms. These must be able to manage the information properly and facilitate the integration with other platforms or applications. There is a very high health cost derived from manual data analysis. Inefficiency in data analysis, whose value outdates easily, has a cost both for the health of patients and for health institutions. The potential for rapid analysis and effective integration with bioinformatics tools can greatly alleviate this cost and provide more value than would be expected.

Índice general

Índice de figuras

Índice de tablas

Índice de Códigos

1 Introducción

Uno de los grandes problemas a los que se enfrenta actualmente la sociedad no resulta solo de la inmensa cantidad de datos que generamos sino del valor que se esconde tras ellos. En 2015 se calculaba un volumen de más de 1000 Exabytes de datos generados, se estimó que se generaría 20 veces ese volumen de datos en menos de 10 años.[3] El ritmo al que generamos datos ha acelerado aún más con tendencias como la computación en la nube, el análisis de datos, Inteligencia Artificial (IA) e Internet of Things (IoT).[4] En 1.1 se ilustran las expectativas de crecimiento en el volumen de datos en el área de la secuenciación genética.

Las arquitecturas Big data nacen a raíz de este problema, la definición formal que propone De Mauro et al. [5, p. 6] es: "Big data es todo aquel activo de información caracterizado por un volumen, velocidad y variedad tan elevados que requieren de tecnologías y métodos analíticos específicos para su transformación en valor".

El sector de la salud no es una excepción, la mejora de la infraestructura, equipos de monitorización, sensorización de pacientes y servicios, etc. están suponiendo también una generación desproporcionada de datos que esconden un inmenso valor. La aplicación de Big data a los datos sanitarios no solo produce una mejora en el cuidado del paciente, sino una reducción de costes sanitarios, algo inestimable en los tiempos que corren.[6]

Este proyecto nace de las necesidades de gestión de datos identificadas por el servicio de Microbiología del Hospital Universitario de Gran Canaria Doctor Negrín. Fueron propuestos como hospital de referencia de Canarias para la nueva "Red de laboratorios para la Vigilancia de Microorganismos Resistentes".[7] Ésto supuso no solo la adquisición de nuevas responsabilidades, sino también de nuevo equipo como el que les permitiría una caracterización más precisa de los microorganismos mediante técnicas de secuenciado completo.

Éstas nuevas técnicas que se busca aplicar en el laboratorio generan una gran canti-

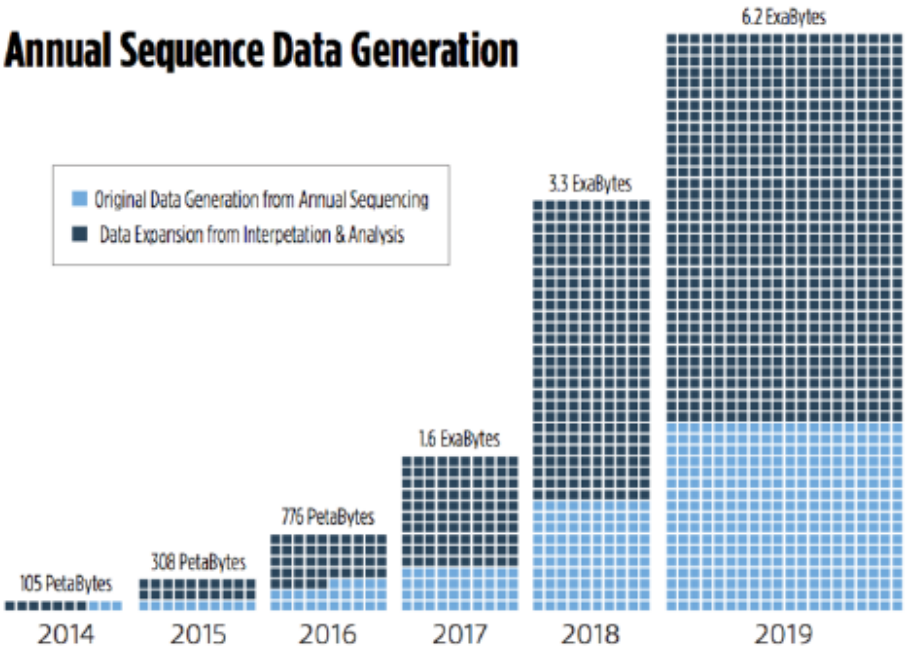

Figura 1.1: Estimación del crecimiento anual de la generación de datos de secuenciación (Kalinichenko et al. 2015 Fig. 1)

dad de información que se suma al ya abrumador volumen de datos de los que dispone el servicio de Microbiología. La solución propuesta consiste en la implementación de una plataforma de datos basada en Big data.

Dentro de las varias fases del ciclo de vida de Big data[8] están:

1. Identificación y recogida de datos
2. Almacenamiento de datos
3. Filtrado de datos y eliminación de ruido
4. Clasificación de datos y extracción
5. Limpieza de datos, validación y agregación
6. Análisis de datos y procesamiento
7. Visualización de datos

En el hospital identifican inicialmente problemas de almacenamiento y análisis/procesamiento de datos. El primer acercamiento al problema implicó un estudio de los diferentes procesos que se llevan a cabo en el laboratorio, teniendo como objetivo la elaboración de una ontología de eventos que serviría para diseñar el modelo de datos de nuestra plataforma. Nos encontraríamos en las dos primeras fases del ciclo de vida de Big data.[8]

Posteriormente recorremos cada una de las fases del ciclo de vida con la elaboración de un digestor de los datos actuales de la aplicación informática Drago, utilizada por el Servicio Canario de Salud, o la planificación de una integración futura con módulos de visualizado y análisis de datos.

Durante el desarrollo del proyecto se presenta una necesidad inminente de análisis de datos con el fin de elaborar los informes de resistencia acumulada anuales del hospital para la investigación de bacterias multirresistentes. Se realizan numerosas reuniones con el objetivo de definir los criterios de éste análisis que hasta ahora se ha ido realizado manualmente.

Finalmente decide incorporarse también la elaboración de estos informes al proyecto.

2 Situación de partida

2.1 Problemática

La colaboración con el Hospital Universitario de Gran Canaria Doctor Negrín comenzó por sus dificultades con el uso de *Nullarbor*, *pipeline* de generación de informes microbiológicos completos a partir de aislados secuenciados.

Nullarbor es el software que adoptan para el manejo de los datos que serán generados en un futuro con las técnicas de whole genome sequencing (WGS). Este software, presenta grandes dificultades de uso para usuarios con poca experiencia en el ámbito de la informática. En el caso particular de Nullarbor es necesario usar una plataforma UNIX, ser capaz de manejar su inmensa cantidad de dependencias, que se encuentran diseminadas entre múltiples repositorios, y tener un manejo mínimo de interfaces basadas en línea de comandos. Este tipo de habilidades escapan al repertorio de habilidades que cabe esperarse de un microbiólogo, haciéndole perder un tiempo muy valioso.

Otro problema que se presenta con la adopción de Nullarbor es la gestión de los datos extraídos mediante WGS (*contigs*) y los datos que a su vez genera Nullarbor. No existe una plataforma o proceso que cierre el círculo completo y permita realizar una gestión adecuada de estos datos. Los datos son manejados de manera manual, se extraen del dispositivo y se transfieren a la plataforma donde esté instalado el Nullarbor (en nuestro caso, una máquina virtual de Linux). Posteriormente se ejecuta el *pipeline* con una serie de *scripts* de bash preconfigurados. Finalmente se espera al resultado, que puede demorarse muchas horas, y se toman los datos manualmente.

El proceso descrito anteriormente presenta varias complicaciones: excesiva intervención del usuario, transferencias de datos, esperas activas por procesos, gestión del almacenamiento de éstos datos de entrada y salida, etc. Esto puede repercutir en errores de usuario, ineficiencia en la gestión del proceso y pérdida de datos, entre otros.

La generación de informes de vigilancia epidemiológica a partir de los datos recogidos a lo largo del tiempo por el hospital es otro de los problemas identificados. Estos informes, dada la complejidad de los criterios de clasificación y contabilización de muestras, se venían realizando de manera manual. Los documentos incluyen porcentajes de sensibilidad a nivel de tipo de bacteria, de familia, estudiando la sensibilidad individual a un antibiótico concreto, a una familia de antibióticos, identificando bacterias multirresistentes, etc.

Entre las herramientas existentes para éste propósito está WHONET. Se trata de aplicación para el manejo de datos microbiológicos en el laboratorio. No solo consta de su propia plataforma de datos, sino que permite realizar análisis, detectar tendencias en la resistencia microbiológica o guiar decisiones de dosis en los tratamientos, entre otras cosas. Esta herramienta lleva en desarrollo desde 1989 por el Centro Colaborador para la Vigilancia de la Resistencia Antimicrobiana de la Organización Mundial de la Salud. [9]

Actualmente existe una integración parcial con WHONET por parte del hospital, periódicamente se realizan exportaciones de datos reglamentarias a la plataforma, pero incompatibilidades entre el formato de los datos de la plataforma local (DRAGO) y la plataforma WHONET hacen difícil aprovechar sus funcionalidades. También es prevalente la falta de entrenamiento en el uso de la plataforma más allá de la exportación reglamentaria con el uso poco intuitivo que tiene la aplicación.

Por encima de todo esto, algunos de los criterios exigidos por la nueva red de vigilancia epidemiológica nacional difícilmente van a ser satisfechos por completo, ejemplos de estos criterios son: la contabilización de sólo la primera muestra por paciente en un periodo, de sólo la última muestra con perfil de resistencia diferente, de la muestra de más resistencia de entre todas las tomadas, etc.

2.2 Objetivos

Este proyecto tiene como objetivo solventar varios de los problemas actuales y futuros a los que se enfrenta el laboratorio de microbiología del Hospital Universitario de Gran Canaria Doctor Negrín:

- Desarrollar una plataforma de datos flexible que solvente los problemas de alma-

cenamiento y transformación de datos para facilitar su análisis y procesado por herramientas bioinformáticas

- Implementar un ingestor de datos que permita migrar a la nueva plataforma datos extraídos de la plataforma informática actual del hospital

- Elaborar una interfaz de comunicación con la plataforma de datos para una fácil integración de ésta con otros nuevos módulos, como nuevas vistas o fuentes de datos

- Proporcionar un generador de informes de vigilancia epidemiológica de organismos resistentes para facilitar de manera inmediata la detección de muestras susceptibles de vigilancia

- Identificar otras necesidades de procesamiento de datos o visualización de éstos

Las necesidades que intenta cubrir este proyecto no son exclusivas del Hospital Universitario, todo sistema sanitario se enfrenta a un creciente volumen de datos de gran valor. En el contexto de la situación pandémica actual cobra aún más importancia una gestión eficaz de los datos sanitarios. Este proyecto no solo explora una solución que permita un uso efectivo de estos datos, sino también proporcionar una solución inmediata para la vigilancia epidemiológica del laboratorio de microbiología.

3 Estado del arte

El progreso en técnicas como la secuenciación de ADN han propiciado la generación de grandes conjuntos de datos de complejo manejo y estudio. La creación de nuevas herramientas diagnósticas y de estudio más potentes sumada a la creciente sensorización de los dispositivos contribuyen también a este problema.

Tecnologías como high-throughput sequencing (HTS), también conocida como next-generation sequencing (NGS) o WGS, están afectando la forma en la que se analiza el ADN bacteriano reemplazando a métodos moleculares para la detección microbiana.[2] Estas nuevas técnicas generan múltiples fragmentos de secuencias (*contigs*) en lugar de una sola secuencia del genoma completo. Estos *contigs* requieren un procesado adicional: cribado de fragmentos defectuosos, ensamblaje, etc., y conforman un conjunto de datos de gran volumen.

Una disciplina emergente y relativamente jóven de investigación busca abordar esta clase de problemas es la bioinformática.

Algunas definiciones de bioinformática que encontramos en la literatura son:

"La bioinformática es un área de investigación en la que informáticos, biólogos, físicos, matemáticos, y químicos combinan su habilidad para colaborar en diversas tareas, desde el descubrimiento de nuevos hechos en sistemas biológicos complejos hasta la racionalización de la organización de los sistemas de salud." [10]

"La bioinformática está conceptualizando la biología en términos de macromoléculas (entendidas desde el punto de vista físico-químico) y aplicando técnicas informáticas (derivadas de disciplinas como las matemáticas, las ciencias de la computación, y la estadística) para entender y organizar la información asociada a estas moléculas en una gran escala."[11]

"La bioinformática es un área de investigación interdisciplinar que aplica metodologías de las ciencias de computación, las matemáticas aplicadas y la estadística al estudio de fenómenos biológicos."[2]

Existen potentes herramientas informáticas para la solución de las dificultades actuales del WGS. Bien para el ensamblado de los *contigs* descritos anteriormente, la limpieza de estos, para realizar análisis de datos sobre el genoma como distancias filogenéticas entre microorganismos o predicciones de resistencia a antibióticos.

La tabla 3.1 da una visión general de las distintas aplicaciones de las que disponemos hoy día para la secuenciación genómica.

Uso	Software	Descripción	URL
Medidas de calidad y preprocesado en la lectura	FASTQC	Herramientas para mostrar estadísticas de secuenciado y para lecturas NGS	http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
	TRIMMOMATIC	Aplicación de línea de comandos para el recorte de aquellos datos de lecturas cortas con terminación individual o en pares	http://www.usadellab.org/cms/?page=trimmomatic
	FASTX-Toolkit	Una colección de aplicaciones de línea de comandos para el preprocesado de archivos FASTA/-FASTQ de lecturas cortas	http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
	PRINSEQ	Aplicación tanto web como de línea de comandos para el filtrado, reformateo o recorte de datos de secuenciación genómica y metagenómica, es capaz de generar resúmenes estadísticos en formato gráfico o tabular NGS	http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Detección de contaminación	Kraken	Clasificación taxonómica de las lecturas, es útil para análisis metagenómicos o para la detección de contaminación en muestras puras de cultivos	`https://ccb.jhu.edu/software/kraken/`
	MIDAS	Clasificación taxonómica de las lecturas, es útil para análisis metagenómicos o para la detección de contaminación en muestras puras de cultivos	`https://github.com/snayfach/MIDAS`
Software y *pipelines* de ensamblado	Velvet	Ensamblador *de novo* diseñado para lecturas cortas	`http://github.com/dzerbino/velvet/tree/master`
	SPAdes	Ensamblador *de novo* diseñado para lecturas cortas; también da la posibilidad de ensamblajes híbridos entre lecturas cortas y largas	`http://cab.spbu.ru/software/spades/`
	Canu	Ensamblador *de novo* diseñado para aquellas moléculas individuales con alto ruido como son las lecturas largas	`http://github.com/marbl/canu`

		INNUca	Un *pipeline* estandarizado, completamente automatizado, flexible, portable e independiente del patógeno para el ensamblaje de genoma bacteriano y control de calidad en lecturas cortas	`http://github.com/INNUENDOCON/INNUca`
		shovill	Un *pipeline* para el ensamblaje de genoma bacteriano que mejora la velocidad y precisión de SPAdes	`https://github.com/tseemann/shovill`
	Tipado *in silico*	Microbial InSilico Typer (MIST)	Generación rápida *in silico* de datos de tipado (e.g. MLST, MLVA) a partir de borradores de ensamblaje del genoma bacteriano	`http://bitbucket.org/peterk87/microbialinsilicotyper`
		ResFinder	Una herramienta web para la detección de genes de resistencia antimicrobial adquiridos en genoma bacteriano usando lecturas en bruto o borradores de ensamblaje de genoma	`https://cge.cbs.dtu.dk/services/ResFinder/`
		MLST1.8	Una herramienta web para la determinación de tipos MLST a partir del genoma bacteriano usando esquemas MLST públicos	`https://cge.cbs.dtu.dk/services/MLST`

	Mlst2.9	Una aplicación de linea de comandos que es capaz de extraer el MLST a partir de genomas bacterianos usando esquemas MLST públicos	`https://github.com/tseemann/mlstCFSANSNP`
	Snippy	Un *pipeline* para la rápida identificación de variantes haploides y la construcción de filogenía usando single nucleotid polymorphism (SNPs) del genoma central	`http://github.com/tseemann/snippy`
Aproximaciones gen a gen	BGSdb	Una base de datos accesible por web diseñada para guardar y analizar información fenotípica y genotípica relacionada, incluyendo el motor de llamada de alelos para una aproximación gen a gen; es la base de datos tanto de PubMLST como de PasteurMLST	`https://github.com/kjolley/BIGSdb`
	Enterobase	Base de datos curada y recurso online para el tipado molecular de *Salmonella*, *Escherichia coli*, *Yersinia spp.* y *Moraxella spp.* usando un acercamiento gen a gen	`http://enterobase.warwick.ac.uk/`

	Genome Profiler	Algoritmo independiente de llamada a alelos gen a gen que usa la vecindad genética conservada para resolver paralogías genéticas.	http://sourceforge.net/projects/genomeprofiler/
	chewBBACA	Algoritmo exhaustivo independiente de alta eficacia de llamadas a alelos gen a gen basado en la codificación de secuencias de ADN, incluye un conjunto de herramientas para dar una vista general del rendimiento de los esquemas	https://github.com/B-UMMI/chewBBACA
Anotación genética	Prokka	Anotación funcional rapida de genomas bacterianos produciendo archivos de salida que cumplen los estándares definidos	http://github.com/tseemann/prokka
	RAST	Servicio completamente automatizado para la anotación de genomas bacterianos y de árqueas	http://rast.nmpdr.org/
	MicroScope	Plataforma de análisis exhaustivo para la anotación genética y el análisis de genomas bacterianos	http://www.genoscope.cns.fr/agc/microscope/home/index.php

	NCBI Pathogen Detection	Una plataforma online para compartir y comparar datos de cepas epidémicas/de brotes; actualmente contiene bases de datos para 20 tipos de especies bacterianas, centrándose en patógenos ligados a los alimentos y a infecciones ligadas a la atención sanitaria	`https://www.ncbi.nlm.nih.gov/pathogens/`
Alineamiento del genoma	Harvest	Un conjunto de herramientas para el alineamiento y visualización del genoma-núcleo para un analisis rápido y de alto rendimiento de genomas bacterianos intraespecíficos	`http://harvest.readthedocs.io/en/latest/`
	Mauve	Alineador para un analisis comparativo de genomas bacterianos completos	`http://darlinglab.org/mauve/mauve.html`
Agrupamiento por homología y estudios de asociación	Roary	*Pipeline* de pangenomas independiente de alta velocidad para genomas bacterianos	`http://sanger-pathogens.github.io/Roary/`
	Scoary	Estudios de asociación a nivel de pangenoma usando la salida del Roary	`https://github.com/AdmiralenOla/Scoary`

	Neptune	Software diseñado para detectar firmas genomicas dentro de poblaciones bacterianas	`https://github.com/phac-nml/neptune`
Inferencia filogenética	RAxML	Estimación de la máxima verosimilitud de la filogenia tanto paralela como secuencial que trabaja con los alineamientos de secuencias de nucleótidos y proteínas	`https://sco.h-its.org/exelixis/software.html`
	FastTree	Computa áboles filogenéticos de máxima verosimilitud a partir de alineamientos de múltiples secuencias en grandes nucleótidos o proteínas	`http://www.microbesonline.org/fasttree/`
	Gubbins	Computa la máxima verosimilitud a partir de alineamientos tras eliminar aquellas regiones que contienen densidades elevadas de sustituciones de bases	`https://github.com/sangerpathogens/gubbins`

	PHYLOViZ 2.0	Software independiente desarrollado en Java para inferencia filogenética, visualización y análisis de métodos de tipado que generan perfiles alélicos y los datos epidemiológicos asociados a partir de secuencias	`http://www.phyloviz.net/`
Herramientas de visualización de datos	Microreact	Una aplicación web para la visualización e intercambio de datos epidemiológicos genómicos	`http://microreact.org`
	Phandango	Aplicación web para una exploración rápida de datos genómicos poblacionales de gran escala combinando los datos de salida de diferentes métodos de análisis genómicos	`https://github.com/jameshadfield/phandango`
	iTOL	Aplicación web para mostar, anotar y gestionar árboles filogenéticos	`http://itol.embl.de/`

	GenGIS 2	Aplicación con gráficos 3-D e interfaces en Python que permiten a los usuarios combinar los datos de mapas digitales y las secuencias	`http://kiwi.cs.dal.ca/GenGIS/Main_Page`
Plataformas y *pipelines* analíticos multipropósito	Centre for Genomic Epidemiology Toolbox	Un conjunto de herramientas web y servicios para el tipado molecular de patógenos, ensamblado genómico, predicción fenotípica (e.g. preddición de la resistencia) y construcción de la filogenia	`http://cge.cbs.dtu.dk/services/`
	Integrated Rapid Infectious Disease Analysis (IRIDA) platform	Una plataforma basada en Galaxy para la investigación en tiempo real de brotes infecciosos usando datos genómicos, incluye un módulo para la gestión de datos de secuencia y flujos de trabajo, un *framework* para ontologías (GenEpiO) y herramientas de visualización de datos	`https://irida.corefacility.ca/documentation/downloads/index.html`

	Integration genomics in surveillance of foodborne pathogens (INNUENDO) platform	Una plataforma para a investigación en tiempo real de brotes infecciosos y la vigilancia de patógenos ligados a alimentos usando datos genómicos, incluye módulos para la gestión de datos de secuencia, ensamblado con medidas QA/QC, un *pipeline* de análisis gen a gen, *framework* para ontologías y herramientas de visualización	`https://github.com/INNUENDOCON/INNUENDO_platform`
	Nullarbor	Un *pipeline* para la generación de informes microbiológicos de salud pública a partir de aislados secuenciados, incluye datos específicos de secuenciación, identificador de especies, subtipos y estudio de los SNPs	`http://github.com/tseemann/nullarbor`

Tabla 3.1: Ejemplos de software actual para la secuenciación del genoma (Carriço et al. 2018 Table 1)

El campo de la bioinformática dispone de múltiples herramientas muy potentes capaces de satisfacer varias de las necesidades actuales de análisis y gestión de datos. Pero estas herramientas tienen varios problemas:

- Muchas herramientas tienen una barrera de entrada muy grande para nuevos usuarios, asumiendo unos conocimientos informáticos demasiado elevados
- Las plataformas de datos de las instituciones sanitarias suelen ser bastante inflexibles y longevas, se hace necesaria la existencia de un nexo de conversión de estos datos a datos manejables por estas herramientas
- La escalabilidad de las plataformas de datos actuales de las instituciones no está preparada para asumir la enorme cantidad de datos generados tanto por técnicas como el WGS como por las herramientas de procesado posterior, generación de informes, análisis, etc.

4 Resultados

En relación a los objetivos propuestos del trabajo se han obtenido los siguientes resultados:

- Ontología de los eventos identificados en el laboratorio de microbiología
- Plataforma de datos basada en *Big data* con soporte para los eventos contemplados en nuestra ontología
- Herramienta que permite la migración de los datos exportados como valores separados por coma (CSV) desde la plataforma *DRAGO* a nuestra plataforma, los datos migrados se limitan a los usados en el laboratorio de microbiología
- Aplicación con una interfaz gráfica de fácil uso para la elaboración de informes de vigilancia epidemiológicos, específicamente, de los informes de sensibilidad acumulada
- Evaluación de las necesidades actuales y futuras de procesamiento de datos en el laboratorio de microbiología

La plataforma de datos desarrollada es fácilmente extendible dada su naturaleza modular. Se trata de un *datahub* que busca centralizar y adoptar todos los datos generados en el laboratorio. Este *datahub* proporciona facilidades de integración con futuros módulos de análisis y visualización de datos a través de terminales Java Message Service (JMS) u otras Interfaces de Programación de Aplicaciones (API) que definamos. El gran interés del *datahub* viene de las posibilidades que ofrece para la extracción de valor de los datos y la escalabilidad en el almacenamiento de éstos.

La herramienta de migración de datos estructura los datos en base a los eventos definidos en nuestra ontología permitiendo una depuración de los mismos y su posterior importación de éstos en el *datahub*.

Otra de las necesidades satisfechas en paralelo es una aplicación que permite elaborar informes de sensibilidad por segmentos de población, servicios o centros hospitalarios

en base a filtros elegidos por el usuario en una interfaz gráfica sencilla. Se permite la selección de entre diferentes criterios de evaluación de los datos. Estos criterios fueron definidos por el laboratorio y preprogramados en la aplicación. Los resultados se presentan en diferentes CSV: uno con el CSV original y dos nuevas columnas con el número de ingreso y el número de episodio de cada solicitud, otro CSV con los datos de sensibilidad de los microorganismos según antibiótico y un tercer CSV con la frecuencia de microorganismos pertenecientes a ciertos fenotipos complejos definidos por el laboratorio.

5 Metodología de desarrollo

5.1 Análisis

El primer contacto con el laboratorio de microbiología, y principal motivo para el inicio de esta colaboración con el hospital, se da por sus dificultades para instalar la herramienta bioinformática *Nullarbor*. Nullarbor es un *pipeline* usado para la generación de informes de microbiología completos a partir de datos de secuenciación de aislados.

Se comienza estudio de las complicaciones derivadas del uso de esta herramienta con la instalación de la misma en una máquina virtual de Linux. Se encuentran grandes dificultades para la instalación de sus dependencias, dificultades de difícil resolución para un usuario sin una formación informática especializada. La herramienta carece de interfaz gráfica, su control es a través de una interfaz de línea de comandos. Aunque el *pipeline* dispone de una documentación bastante completa, los usuarios no informáticos no suelen sentirse cómodos con este tipo de interfaces. El control de la aplicación se estaba llevando a cabo mediante una serie de *scripts* preconfigurados que le fueron entregados al hospital por otro laboratorio familiarizado con el uso de la herramienta.

El laboratorio de microbiología del Hospital Universitario requiere del uso de *Nullarbor* dada su próxima adopción de ténicas de WGS. Estas técnicas generan *contigs*, secuencias de ADN superpuestas cuyo ensamblado permite la reconstrucción de la secuencia original, en este caso, el genoma del microorganismo. El proceso de ensamblado y estudio de la secuencia obtenida requiere de herramientas bioinformáticas especializadas. La herramienta elegida por el hospital es *Nullarbor*.

La naturaleza de los datos usados por la herramienta, los *contigs*, hace que el volumen bruto de información sea muy grande. La herramienta *Nullarbor*, a su vez, genera un gran volumen de datos intermedios: datos de ensamblado, mapas genómicos anotados, datos de resistencia, distancias filogenéticas, etc. Esta enorme cantidad de datos hace patente las futuras complicaciones que se acabarán dando en la gestión de los mismos y la necesidad de buscar una arquitectura capaz de dar solución a estos problemas.

Las dificultades de uso de esta herramienta bioinformática dejan reflejada la necesidad de desarrollo de facilitadores en el uso de éstas o de la propia elaboración de herramientas bioinformáticas de fácil uso.

Posteriormente se realizan múltiples visitas al laboratorio y se estudian los distintos procesos del laboratorio de microbiología con el fin de elaborar una ontología capaz de describir la realidad de éste. Dentro de los procesos identificados respecto a las muestras microbiológicas se encuentran:

- Toma de muestra
- Solicitud de estudio
- Cultivo
- Identificación morfológica preliminar
- Elaboración del antibiograma

La toma de muestra puede ser de diferentes tipos: muestras fecales, respiratorias, cutáneas, etc. Un caso de especial interés es la distinción entre muestras del entorno como el aire, superficies, agua de diálisis, etc. que no están asociadas a un individuo, y las muestras individuales. Las muestras tienen asociadas una solicitud (urgente o no urgente) realizada al laboratorio por un solicitante, generalmente un o una médico y, tendrán también asociadas, salvo en el caso de las muestras del entorno, un individuo. A su vez cada muestra cuenta con un orígen compuesto por el centro y servicio en el que se tomó.

En cuanto al proceso de cultivo, cada muestra puede tener 1 o más aislados, siendo cada aislado un tipo de microorganismo diferente. En una identificación preliminar los miembros del laboratorio aplican su criterio experto para identificar la morfología de los aislados de la muestra: bacilo, coco, levadura, grampositivo o gramnegativo.

Durante el cultivo de la muestra, ésta se expone *in vitro* a diferentes antibióticos para el posterior estudio de su Concentración Mínima Inhibidora (CMI). El CMI se trata de la concentración mínima de un antimicrobial que inhibe visiblemente el crecimiento de un microorganismo tras un día de incubación. Con el CMI se puede determinar la sensibilidad de una bacteria, que puede ser Sensible (S), Intermedia (I) y Resistente (R). Esta información es vital para saber qué antibiótico y dosis usar *in vivo* o para el

estudio de la evolución de la resistencia en el tiempo. [12]

En ocasiones se llevan a cabo procesos como la aplicación de técnicas específicas para ayudar a la identificación que no están incluidas en el flujo normal de procesos por los que pasa una muestra en el laboratorio.

Respecto a la plataforma informática usada actualmente en el laboratorio existen varios identificadores:

- Identificador de solicitud de estudio de la muestra externo
- Identificador de solicitud de estudio de la muestra interno
- Identificador de aislado

El identificador de solicitud externo proviene de la aplicación a través de la cual el o la médico solicita el cultivo y estudio de un microorganismo, este identificador carece de relevancia para los procesos internos del laboratorio. El identificador de solicitud interno se le asigna al llegar la solicitud al laboratorio, este identificador es compartido por todos los aislados que encontremos en el futuro en esta muestra. El identificador de aislado está compuesto por el identificador de solicitud interno concatenado con su número de secuencia (e.g., si una muestra tiene como número de solicitud interno ”3333” y contiene dos aislados, el primer aislado tiene como identificador ”33331” y el segundo aislado ”33332”).

La ontología de eventos preliminar definida es la siguiente:

[Sample]
ts:
id:
type:
where:

[Sample.Request]
urgency:
who:

[Sample.Environmental]
type:

[Sample.Individual]
id:
gender:
birthday:

[Culture]
ts:
id:
type:

[Culture.Isolate]
code:

morphology:
gram:

[**Antibiogram**]
ts:
id:
isolate:

[**Antibiogram.Identification**]
technique:
result:
organism:

[**Antibiogram.AntibioticResponse**]
technique:
antibiotic:
response:
cmi:

[**Antibiogram.AntibioticResponse**]
technique:
antibiotic:
response:
cmi:

[**Antibiogram.AntibioticResponse**]
...

Se identifican por tanto los eventos de *Sample* para representar la muestra, *Culture* para el cultivo, y *Antibiogram* para el antibiograma. A su vez una *Sample* tiene asociada una *Request* para la petición, y puede o no tener una faceta *Environmental* si es una muestra de entorno o *Individual* en caso de ser de un individuo.

El evento *Culture* a su vez tendra uno o más *Isolate* que representan los distintos aislados que se pueden encontrar en una muestra tras realizarse un cultivo. El *Antibiogram* puede o no tener asociada una *Identification* en caso de aplicarse técnicas de identificación adicionales fuera de la norma, además tendrá una o más *AntibioticResponse* por cada antibiótico usado para la elaboración del antibiograma.

Los eventos constan de timestamp (ts) o marcas de tiempo, identificadores manteniendo el sistema de identificación interno del laboratorio, y la información relevante a cada evento, que se presenta a continuación:

Cada *Sample* contará con un *type* para modelar el tipo (respiratoria, fecal, sanguínea, etc.) y un *where* para su origen (hospital y servicio). La *Request* asociada tendrá una *urgency* según la urgencia de la petición (urgente o no urgente) y un *who* para referirse al que realizó la solicitud. Su *Environmental*, en caso de estar presente, consta de un *type* (aire, superficie, etc.). El *Individual*, en caso de haberlo, representa al individuo del que se tomó la muestra y contará con su identificador, que en este caso es el número de historia clínica (NHC), además de un *gender* para el género y un *birthday* para su fecha de nacimiento.

Los *Isolate* cuentan con un *code* para su identificador numérico interno dentro del cultivo, una *morphology* (bacilo, coco o levadura) y un *gram* según sea grampositivo o gramnegativo.

Finalmente cada *Antibiogram* tendrá una referencia al aislado sobre el que se realizó el antibiograma en *isolate*. Su *Identification*, en caso de estar presente, nos informará de qué técnica adicional se aplicó con *technique*, qué resultado se obtuvo con *result* y cuál fue el veredicto de la identificación en *organism*. *AntibioticResponse* cuenta con *technique* para representar la técnica empleada en la evaluación de la respuesta al antibiótico, *antibiotic* con el antibiótico asociado a la respuesta actual, *response* con la respuesta a dicho antibiótico (S, I o R) y *cmi* con el valor obtenido evaluando la concentración mínima inhibidora.

Durante el análisis, una de las limitaciones identificadas más apremiantes es la elaboración de los informes de vigilancia epidemiológica. Son informes anuales con diferentes datos de frecuencia, análisis de la sensibilidad, etc., que permiten obtener un mapa de la resistencia microbiológica en el servicio. El objetivo del laboratorio es que sean trimestrales pero dada la dificultad de los análisis de datos que exige el informe, actualmente se elaboran de manera manual, haciendo inviable la trimestralidad del informe.

Se llevaron a cabo diferentes reuniones para definir las directrices a seguir en la elaboración de estos informes. Los diferentes criterios de interés en la elaboración de los informes de sensibilidad son los siguientes:

- Criterio 1 - Primer microorganismo aislado por paciente en el periodo de tiempo seleccionado independientemente del tipo de muestras o perfil de sensibilidad antibiótica
- Criterio 2 - Primer microorganismo aislado por paciente con distinto antibiograma en el periodo de tiempo seleccionado independientemente del tipo de muestra. Se deben incluir todos los primeros aislamientos representantes de los fenotipos observados, considerando fenotipos diferentes aquellos en los que existe un cambio de S o I a R o viceversa de uno o más antibióticos.
- Criterio 3 - Considerar el aislado más resistente de entre todos los aislados del mismo microorganismo que tenga un paciente en un periodo de tiempo definido
- Criterio DTR - Determinar el porcentaje de microorganismos aislados con:

- Fenotipo DTR - Resistente a todos los siguientes antibióticos: piperacilina-tazobactam (PTZ), aztreonam (ATM), cefotaxima (CTX), ceftriaxona (CRO), ceftazidima (CAZ), cefepime (FEP), imipenem (IMP), meropenem (MEM), ertapenem (ERT), ciprofloxacino (CIP), levofloxacino (LVX), moxifloxacino (MOX) y en el caso de Acinetobacter baumannii complex, además a ampicilina-sulbactam (SAM)

- Fenotipo resistencia a carbapenémicos (CR) - Resistencia a alguno de los siguientes antibióticos

  * Enterobacter spp., Escherichia coli, Klebsiella spp.: ERT, IMI, MEM

  * Pseudomonas aeruginosa, Acinetobacter baumannii complex: IMP, MEM

- Fenotipo de resistencia a cefalosporinas de espectro extendido (ECR) - Resistencia a alguno de los siguientes antibióticos: CTX, CRO, CAZ, FEP

  * Enterobacter spp., Escherichia coli, Klebsiella spp.: CTX, CRO, CAZ, FEP

  * Pseudomonas aeruginosa: CAZ, FEP

  * Acinetobacter baumannii complex: CTX, CRO, CAZ, FEP

- Fenotipo de resistencia a fluorquinolonas (FQR)

  * Enterobacter spp., Escherichia coli, Klebsiella spp.: CIP, LVX, MOX

  * Pseudomonas aeruginosa: CIP, LVX

  * Acinetobacter baumannii complex: CIP, LVX, MOX

En el criterio DTR hay que tener en cuenta que la pertenencia de un aislado a estos fenotipos es excluyente y jerarquizada en el orden DTR > CR > ECR > FQR, se consideran a los aislados como diferentes cuando el tiempo entre la aparición de ambos aislados para un mismo paciente es superior a treinta días. En el caso de aislamientos con menos de treinta días de diferencia se considerará únicamente al aislado perteneciente al fenotipo jerárquicamente superior. En las infecciones mixtas se consideran a todos los aislados de forma independiente.

Existen directrices adicionales sobre qué consideramos *paciente* en los análisis de carácter general. Se define la existencia de *ingresos* y *episodios*. Todas las muestras de un mismo individuo (entendiéndose esto por todas las muestras ligadas al mismo

NHC) se consideran del mismo ingreso siempre que el tiempo entre la última muestra y la siguiente sea de catorce o más días (siendo este valor configurable). Un episodio comprende todas las muestras en las que entre la última toma y la siguiente ha pasado menos de una semana (valor configurable). A efectos del análisis de datos se considera que hay tantos pacientes como números de ingreso, salvo que el criterio aplicado tenga sus propias directrices respecto al tiempo entre muestras (como en el criterio DTR).

La identificación de éstos *ingresos* y *episodios* es esencial para la evaluación correcta de la resistencia epidemiológica ya que el objetivo es evaluar el mayor número de aislados posible, y con la definición de los *ingresos* evaluamos de manera independiente distintos aislados del mismo microorganismo para un mismo individuo.

Es de interés proporcionar distintos filtros para evaluar la sensibilidad:

- Por origen (servicio o centro)
- Edad
- Género
- Tipo de muestra

De esta forma se pueden realizar estudios segmentados de la evolución de la sensibilidad para cada microorganismo.

5.2 Diseño

La primera fase del desarrollo consiste en la elaboración de una plataforma de datos basada en *Big data* haciendo uso de la ontología de eventos definida en la primera fase. Como valor entregado, además de la plataforma de datos, se cuenta con un digestor de datos que permite introducir datos de la plataforma informática actual del laboratorio en nuestra nueva plataforma.

En una segunda fase se realiza un estudio de los requisitos sobre análisis de datos para la elaboración de los informes de vigilancia epidemiológicos y se elabora un generador de informes capaz de satisfacer parte de los requisitos encontrados.

5.2.1 Desarrollo de la plataforma de datos

Se elige para el desarrollo de la plataforma de datos el framework *Intino* desarrollado en el Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI). Este framework de desarrollo en *Java* usa MDE para facilitar la elaboración de soluciones a problemas complejos.

El MDE considera a los modelos como artefactos centrales en el campo de la ingeniería del *software*, permitiendo la creación o ejecución automática de sistemas *software* a partir de estos modelos. Se hace necesario definir qué entendemos por modelo, siendo el modelo una abstracción del sistema usado frecuentemente en sustitución del sistema que tenemos bajo estudio (en este caso el laboratorio de microbiología). Se trata por tanto de una visión simplificada del sistema.[13]

Dentro del MDE encontramos conceptos centrales como *metamodelo*, que es un modelo que define la estructura de un lenguaje de modelado. El nivel de abstracción al que queramos o requiramos llegar depende del sistema modelado, en nuestro caso en particular sólo se modelará el sistema en específico estudiado, el laboratorio de microbiología, no requiriendo la definición de metamodelos específicos. El MDE se suele relacionar con conceptos como *Software factory*, una colección estructurada de *software assets* usados en la creción de tipos de *software* específicos, puede incluir procesos, DSL, plantillas, entornos de desarrollo integrados, configuraciones y vistas.[13]

Intino hace uso de DSL para el modelado de nuestra solución y genera diferentes estructuras de código recurrentes en las arquitecturas *Big data*. Provee muchas utilidades como: soporte para la definición de una API Transferencia de Estado Representacional (REST), la creación de terminales JMS, librerías con diferentes utilidades para el manejo de los mensajes de la plataforma, etc.

Para esta definición del modelo es por tanto imprescidible un buen entendimiento del sistema estudiado, siendo vital la capacidad de elaboración de una ontología adecuada capaz de modelar la realidad estudiada. En 5.1 se puede ver un ejemplo de la facilidad para definir un modelo usando el DSL Ness a partir de la ontología obtenida durante la fase de análisis.

Código 5.1: Ejemplo de definición del modelo con el DSL Ness

```
dsl Ness
```

```
Event Sample
    Attribute ts as DateTime
    Attribute id as Text
    Attribute type as Text
    Attribute where as Text
    Component(multiple = false) Request
        Attribute urgency as Bool
        Attribute who as Text
    Component(multiple = false) Environmental
        Attribute type as Text
    Component(multiple = false) Individual
        Attribute id as Text
        Attribute gender as Word("MALE, FEMALE")
        Attribute birthday as DateTime

Event Culture
    Attribute ts as DateTime
    Attribute id as Text
    Attribute type as Text
    Component Isolate
        Attribute code as Text
        Attribute morphology as Text
        Attribute gram as Word("POSITIVE, NEGATIVE")

Event Antibiogram
    Attribute ts as DateTime
    Attribute id as Text
    Attribute isolate as Text
    Component Identification
        Attribute technique as Text
        Attribute result as Word("POSITIVE, NEGATIVE")
        Attribute organism as Text
    Component AntibioticResponse
        Attribute technique as Text
        Attribute antibiotic as Text
        Attribute response as Word("RESISTENTE, INTERMEDIO, SENSIBLE")
        Attribute cmi as Text
```

Como se puede observar, este modelo tiene como elemento central los *eventos*, para entender el concepto de *evento* es necesario explicar en qué consiste el *Event-sourcing*. Martin Fowler nos dice que la idea fundamental del *Event-sourcing* es la de tener todo cambio de estado de una aplicación capturado como un *evento*. Este *evento* debe estar almacenado de manera secuencial según el momento del tiempo en el que se da. Esto permite no solo poseer un registro de todos los cambios de la aplicación, sino recrear una secuencia de eventos que nos permita ver el estado de la aplicación en un instante determinado.[14]

Si aplicamos el *Event-sourcing* al sistema modelado, somos capaces de simular completamente la evolución del mismo en el tiempo. De ahí que las piezas básicas de nuestro modelado sean los *eventos*.

Los eventos introducidos en la plataforma o generados por ésta se introducen en

tanques. Cada tanque tiene asociado un tipo de evento y proveé de métodos para el trato de éstos permitiendo, por ejemplo, obtener un *stream* de datos ordenados en el tiempo o los eventos generados en un periodo en específico, además de facilidades para la introducción de datos.

Este almacenamiento en tanques se traduce en la escritura de mensajes en ficheros de texto plano organizados por tipo de evento y periodo en archivos *zim*, que son archivos *zip* de mensajes. El formato que tiene cada evento en el fichero de texto sigue el modelo definido anteriormente. Este modelo es flexible, pudiendo actualizarse en el tiempo si fuera necesario, los eventos antiguos que existan en la base de datos pueden regenerarse para ajustarse al nuevo modelo aplicando las transformaciones que sean necesarias en cada caso.

El framework permite definir terminales ligados a cada uno de los tanques de la plataforma, además de sus *publishers* y *subscribers*. Los *publishers* son aquellos que introducirán mensajes en el tanque asociado al terminal. Los *subscribers* por otro lado, serán notificados de cada nuevo evento que se registre en el tanque asociado.

Para entender por completo el alcance de este diseño arquitectónico es necesario explicar una serie de términos relacionados con las arquitecturas de *Big data*:

El *datahub* es el elemento central de nuestro diseño, se trata de un centro de datos que busca integrar todos los datos en un mismo punto. El *datahub* proporciona un punto común al que deben acudir todas las unidades de negocio para obtener cualquier tipo de dato, colecciona datos de distintas fuentes y los organiza permitiendo una posterior distribución y un fácil acceso.

El *datalake* o lago de datos se define como un repositorio de datos basado en tecnologías de bajo coste que puede contener datos en bruto con o sin estructura. El objetivo del *datalake* es centralizar toda la información generada por una organización independientemente de su valor percibido. Una vez los datos se encuentran en el *datalake*, éstos están disponibles para cualquier análisis, procesamiento o transformación que se requiera. [15]

Un *datamart* no es más que un segmento de los datos del *datalake* centrada en un ámbito concreto del negocio, e.g., en una empresa no tiene la misma utilidad cierta información para la unidad de ventas que para la de finanzas o marketing. Los *data-*

marts facilitan la extracción de valor de los datos reduciendo, por ejemplo, el coste de los análisis de datos de una unidad de negocio.

Los *mounters* tienen la responsabilidad de montar estos *datamarts* a partir de los datos existentes en el *datalake*.

Los *adapters* generan nuevos eventos cuando se producen determinados eventos, e.g., pueden definirse procesos transformacionales o análisis de datos a realizar sobre un determinado tipo de eventos siempre que se produzcan, el resultado de estos procesos, a su vez, debe guardarse también en el *datalake*, produciendo un nuevo evento.

Un *feeder* se encarga de introducir eventos en el *datalake*.

Finalmente los *digester* tratan datos legados (*legacy data*) depurándolos y realizando las transformaciones necesarias para su integración en el *datalake* como evento.

En este proyecto se ha desarrollado un *datahub* para los eventos identificados en la ontología del laboratorio de microbiología. Además de un *digester* que registra eventos en el *datalake* a partir de los archivos con CSV exportados desde la plataforma informática DRAGO.

El desarrollo de este *digester* se realizó en Java, en el propio módulo *datahub*. Ésta digestión consistió en una lectura y análisis de estos archivos CSV con ciertas consideraciones:

Parte de la labor de introducción de datos en la plataforma informática DRAGO es manual, lo que conlleva la presencia de inconsistencias en los datos, errores en los registros o datos faltantes. Fue necesaria la toma de algunas decisiones como la desestimación de la capitalización de las letras o cantidad de espacios.

En el caso de aquellos registros con datos faltantes, véase, la fecha de nacimiento de un paciente o la fecha de registro de una petición, se consultó con el laboratorio cómo proceder. En algunos casos se tomó la decisión de asignar una fecha inverosímil que marcara el caso, en otras directamente no se tuvo en cuenta el registro.

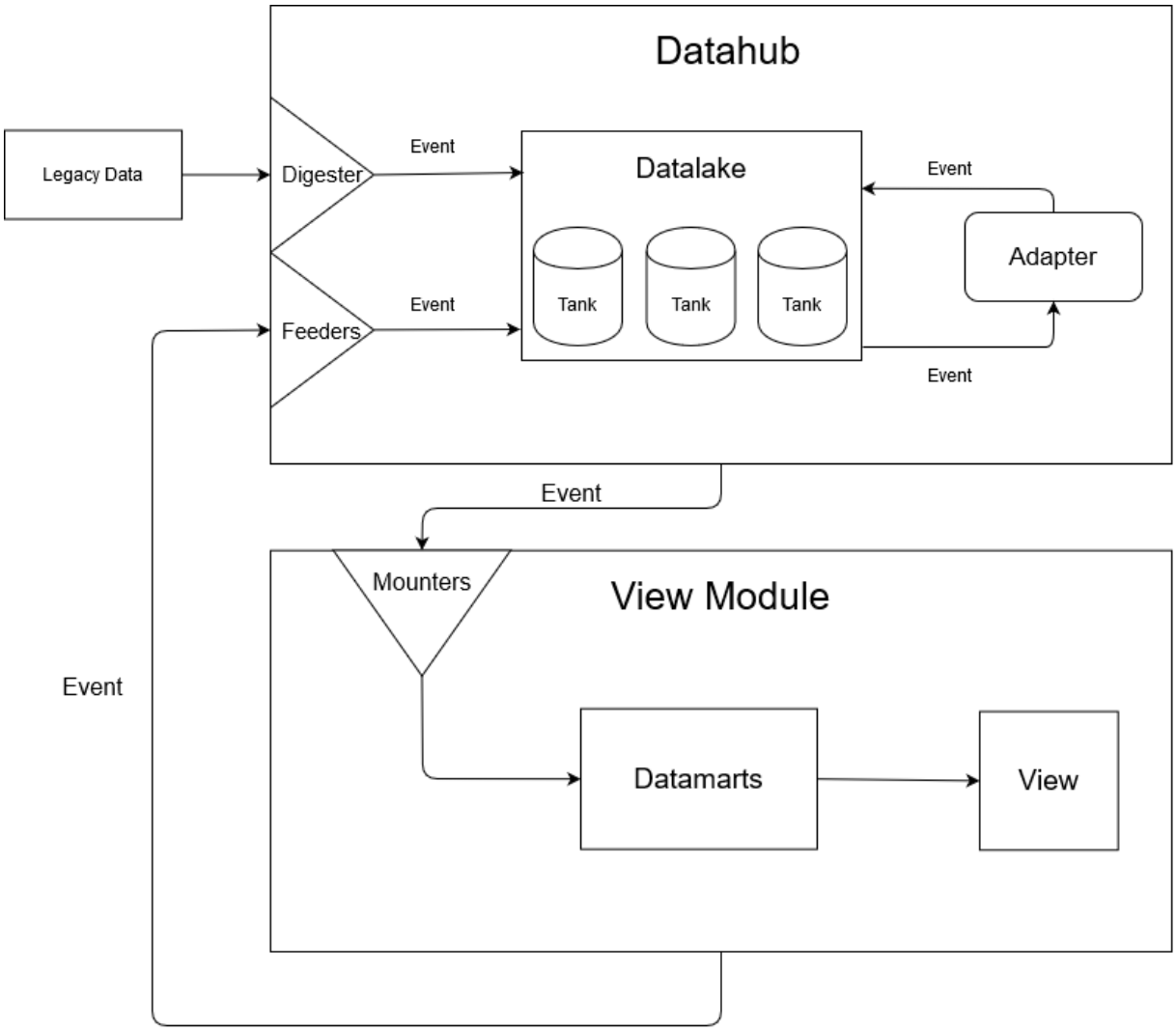

Figura 5.1: Diagrama de una arquitectura basada en *Big data*

5.2.2 Desarrollo del generador de informes

Para la elaboración de los informes de microbiología se estudiaron diferentes opciones, tanto *R* como *Python* tienen herramientas de gran potencia para el análisis de datos. A pesar de tener cierta formación en *R* se opta por usar *Python* por su flexibilidad y para explorar librerías de uso frecuente como *pandas* y *NumPy*.

La librería *pandas* tiene una estructura de datos, el *dataframe*, que se puede entender como un diccionario de *Series*, que es la estructura primaria de *pandas*. Este *dataframe* permite realizar operaciones de gran potencia sobre los datos y estructurarlos de manera sencilla y ordenada. Se puede entender este *dataframe* como una tabla de datos o tabla *SQL*. Se puede indexar cada *Series* manualmente o se indexa autoincrementalmente, además *pandas* proporciona utilidades para la generación del *dataframe* a partir de ficheros CSV o *excel*.

Como primer paso se genera el *dataframe* a partir del CSV exportado de la plataforma *DRAGO*. Con las utilidades de *pandas* se itera la estructura de datos aplicando las transformaciones necesarias, eliminando aquellas entradas descartadas por el criterio aplicado y contabilizando los datos a obtener. Los datos de frecuencias y sensibilidad analizados son a su vez guardados en nuevos *dataframe* para ser finalmente exportados como CSV.

La estructura del CSV de resultados muestra en cada fila un microorganismo y en cada columna un antibiótico. Se muestra el número de aislados evaluados por microorganismo para un determinado antibiótico, el número de aislados sensibles al antibiótico y el porcentaje de sensibilidad al mismo. Se genera además otro CSV con los números de ingreso y episodio añadidos a la tabla de datos original. En caso de seleccionarse incluir el análisis del fenotipo DTR se sumará al resultado un CSV con las frecuencias microorganismos DTR, CR, ECR y FQR.

Python también cuenta con herramientas como *PyQt* [16] que permite hacer uso del framework gráfico de C++ *Qt* [17] para la elaboración sencilla de interfaces gráficas y *QtCreator* para diseñar estas interfaces. Se decide elaborar una interfaz sencilla que permita al usuario definir los criterios a aplicar en el informe de sensibilidad acumulado.

Qt es un framework flexible y es soportado por *Windows*, *macOS*, *Linux*, *iOS* y *Android*. *PyQt* permite hacer uso de *Qt* por medio de *bindings* implementados como

módulos de *Python*. *QtCreator* es una aplicación gráfica que facilita el diseño de las interfaces, reduciendo la carga programática del diseño de la interfaz.

En 5.2 tenemos la vista principal de la aplicación de generación de informes, tiene diferentes apartados donde definir las rutas de los ficheros, siendo el CSV de resistencia aquel desde el que tomar los datos. Para la selección de la ruta se hace uso del explorador de archivos 5.3. La interfaz de la aplicación permite elegir incluir o no el análisis del fenotipo DTR 5.4.

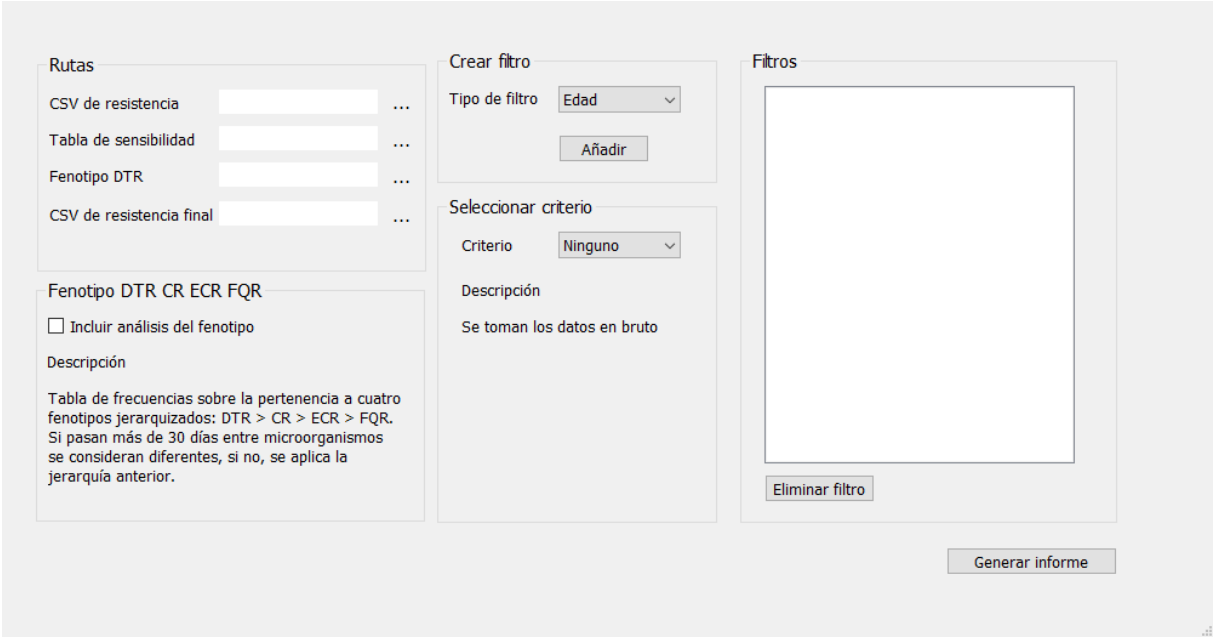

Figura 5.2: Vista principal de la aplicación

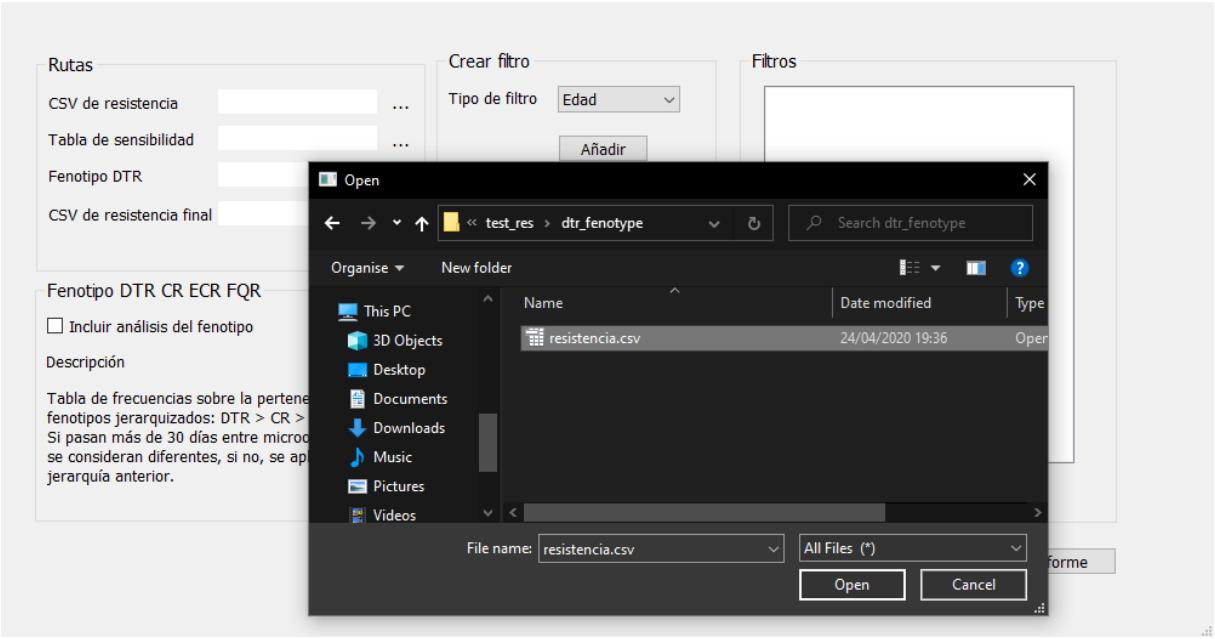

Figura 5.3: Vista de selección de archivos

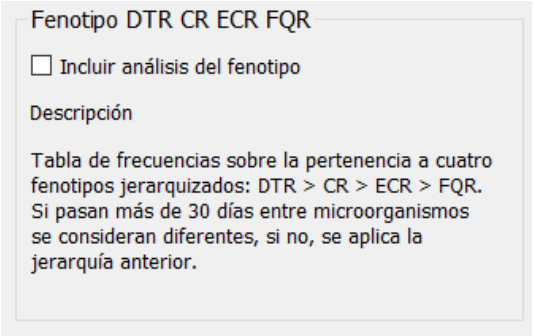

Figura 5.4: Vista de selección de filtro

El requisito de filtrado de los datos por distintos valores está cubierto con el apartado de la vista dedicado a la creación de filtros. 5.5 5.6 5.7 5.8 5.9

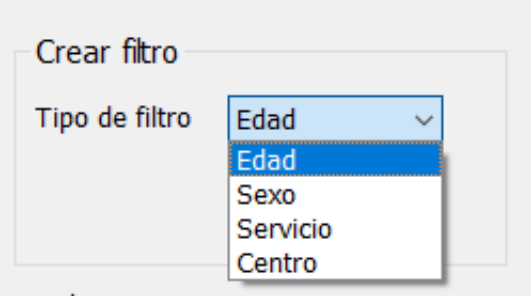

Figura 5.5: Vista de selección de filtro

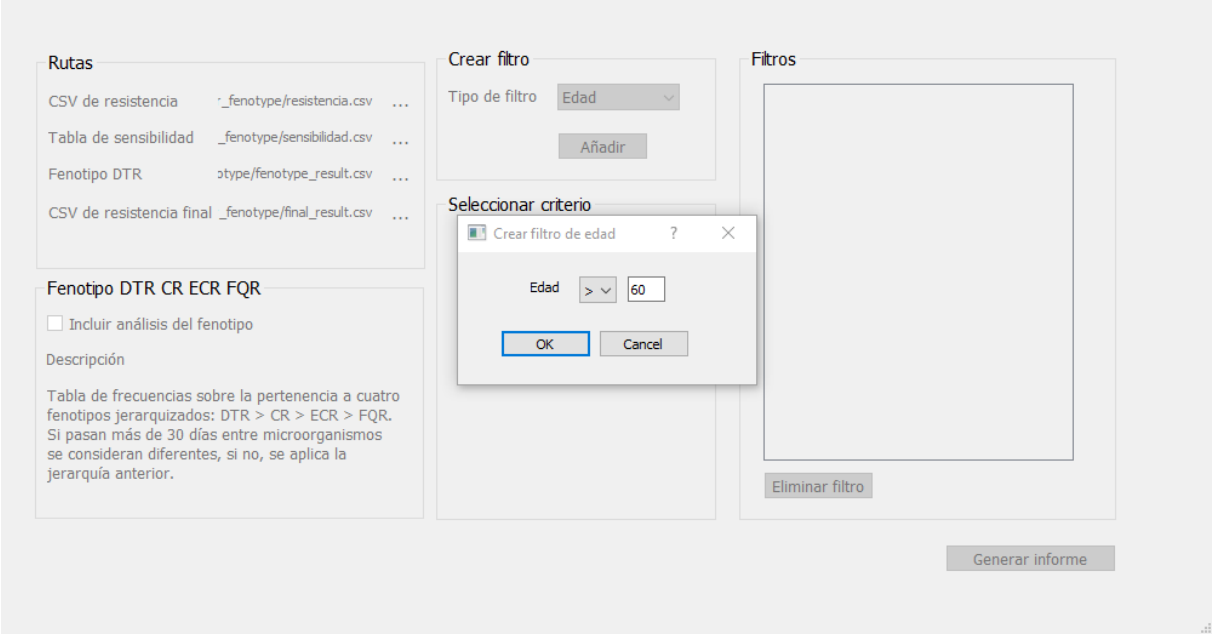

Figura 5.6: Vista de creación del filtro de edad

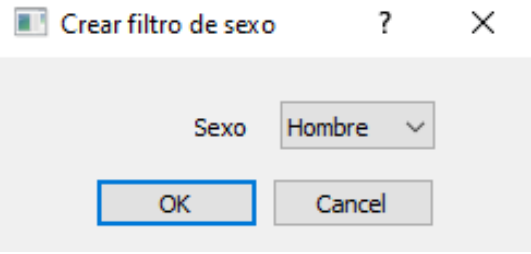

Figura 5.7: Vista de creación del filtro de sexo

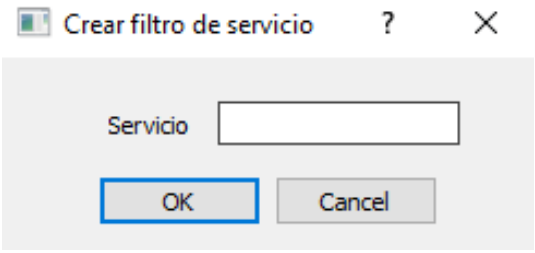

Figura 5.8: Vista de creación del filtro de servicio

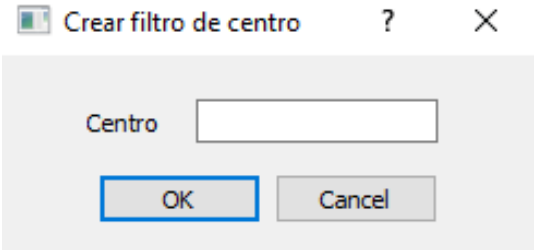

Figura 5.9: Vista de creación del filtro de centro

Cada uno de los criterios a aplicar son excluyentes, la vista de selección de criterio permite seleccionar cuál aplicar, disponiendo de una descripción del criterio elegido. 5.10 5.11

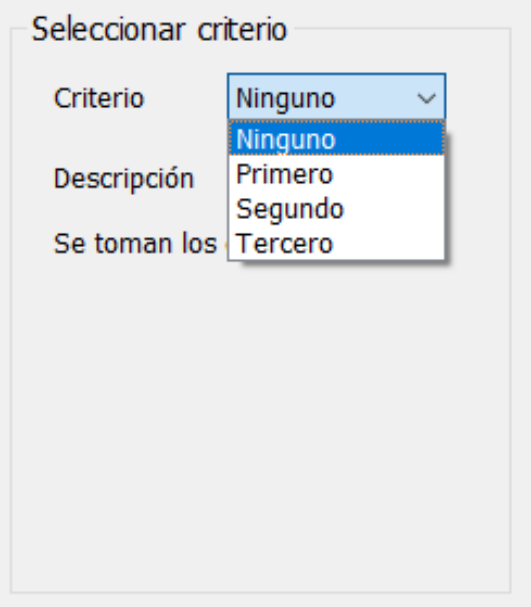

Figura 5.10: Vista de selección del criterio a aplicar

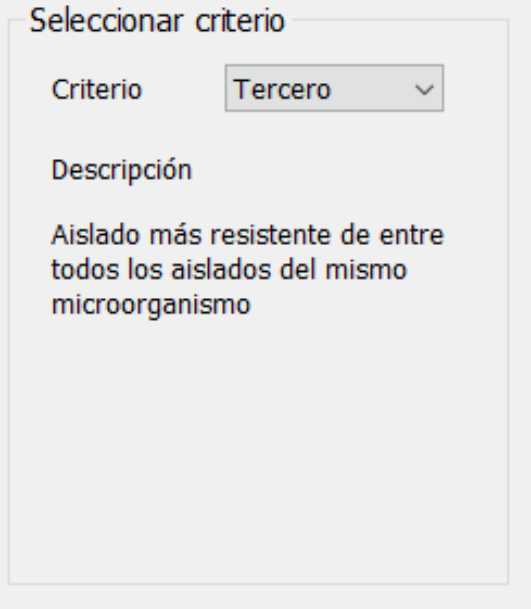

Figura 5.11: Vista de descripción del criterio seleccionado

Finalmente se dispone de una lista donde visualizar cada uno de los filtros elaborados, permitiendo eliminar aquellos no deseados. 5.12

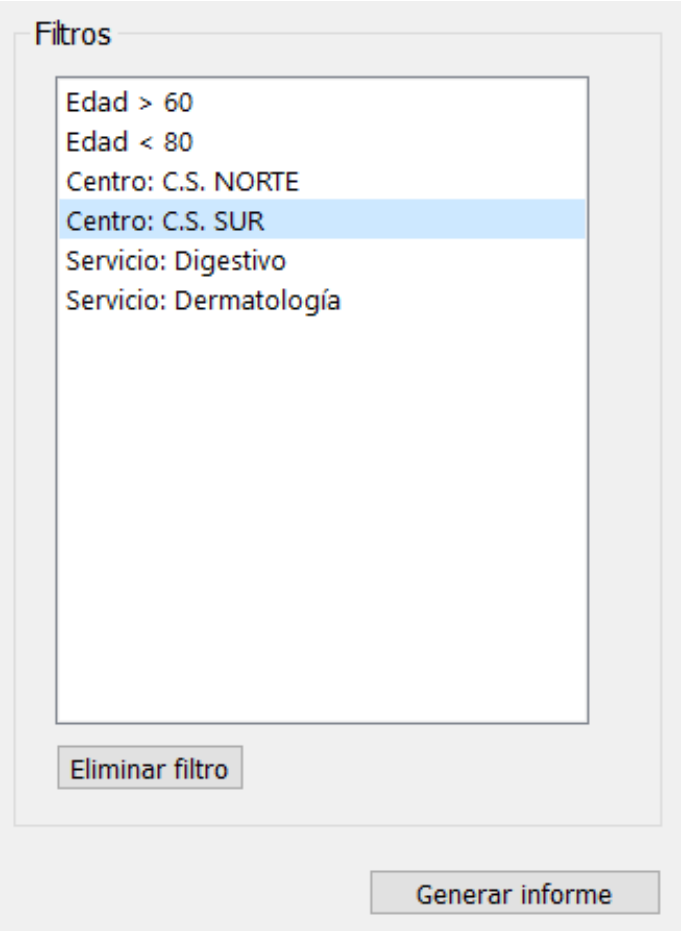

Figura 5.12: Vista de los filtros elaborados y generación de informe

6 Conclusiones y vías futuras

La plataforma de datos desarrollada cumple con los objetivos planteados al inicio del proyecto. La arquitectura es fácilmente escalable y lo suficientemente flexible para ser adaptada en el futuro a nuevas fuentes de información o formatos de datos.

El ingestor de datos integrado en la plataforma es capaz de integrar correctamente los datos legados del laboratorio en nuestra plataforma, realizando la depuración y transformaciones necesarias en cada caso.

Los DSL de los que hace uso *Intino* permiten declarar cómodamente diferentes terminales JMS o APIs REST con los que integrar el *datahub* con otros módulos de visualización o análisis de datos.

El generador de informes proporciona una interfaz sencilla para la realización de análisis antes largos y tediosos por su elaboración manual. La intefaz provee de suficientes opciones de configuración como para aportar valor al microbiólogo.

A nivel personal este proyecto me ha servido para adquirir experiencia trabajando en entornos interdisciplinares. Me ha dado la oportunidad de explorar el ámbito de la bioinformática y aprender diferentes tecnologías de análisis de datos como *pandas* y *NumPy*. También me he familiarizado con el uso de la librería *PyQt* para la implementación de la graphical user interface (GUI) y *QtCreator* para el diseño de la interfaz.

El diseño de las arquitecturas de datos basada en *Big data* ha requerido la comprensión de diferentes conceptos en el contexto del *Big data* como:

- *Datahub*
- *Datalake*
- *Datamart*

- *Data warehouse*
- *Event*
- *Feeder*
- *Digester*
- *Tank*
- *Mounter*

Ha requerido mi familiarización en distintos patrones de diseño como *Event sourcing* y *Modular design*.

El uso del framework *Intino* exigió el estudio de metodologías de desarrollo como *Model-driven engineering* y los *Domain-specific languages*.

Al final de este proyecto me siento más experimentado practicando *Test-driven development* dado su extenso uso a lo largo de éste y la necesidad de probar exhaustivamente el funcionamiento del programa.

Relacionando las necesidades del proyecto con los conocimientos adquiridos en mi recorrido por la universidad, considero que a nivel de arquitecturas y diseño del *software* lo impartido en la universidad me ha servido para aprender con más facilidad los patrones y arquitecturas utilizados en este proyecto.

A nivel de tecnologías sentí que lo aprendido en la universidad era deficiente, al menos en mi especialización, *Ingeniería del Software*. Considero esencial el aprendizaje de librerías de tratado de datos como *pandas* y un mayor uso de lenguajes como *Python*, que es cada vez más popular en el sector.

En cuanto a áreas de oportunidad futuras, la plataforma de datos desarrollada tiene varias posibilidades de extensión. No solo es ampliable el tipo de eventos a recoger, las facilidades para la integración con el *datahub* ofrece infinitas oportunidades de extensión.

Una necesidad que se pone de manifesto durante el análisis de los requisitos del la-

boratorio es la elaboración de mapas de calor que aporten una visualización gráfica y clara de los niveles de resistencia a nivel de centro e, incluso, a nivel de habitación o área del centro hospitalario. Este mapa permitiría identificar posibles microorganismos multirresistentes presentes en el entorno, bien en un quirófano en concreto o en un aparato diagnóstico. Aportando así información o guía sobre el origen de diferentes infecciones nosocomiales.

Actualmente, en la plataforma de datos del laboratorio de microbiología, no cuentan con la información del número de habitación, pero en futuras revisiones de la plataforma se propondrá la inclusión de este dato en la solicitud de estudio de la muestra.

Una de las principales motivaciones para el desarrollo de la plataforma de datos son las complicaciones derivadas del uso de WGS y de la integración con otras herramientas bioinformáticas como *Nullarbor*. En un futuro sería interesante la implementación de módulos de transformación de datos o de integración con diferentes herramientas, con la consecuente ampliación del *datahub*, que debe ser capaz de registrar la información que a su vez generan estos nuevos módulos.

Son de interés también las oportunidades de automatización existentes en los procesos del laboratorio. Es posible programar la elaboración de análisis periódicos, como la generación de los informes trimestrales, de manera automática. Un objetivo a perseguir en el futuro es la integración continua con la plataforma de datos actual del hospital sin necesidad de realizar exportaciones de los mismos.

Se recomienda la realización de nuevos estudios del proceso de laboratorio para la identificación de fuentes de información no consideradas, oportunidades de sensorización o eventos no reflejados en la ontología actual.

Recientemente, a la luz de la pandemia ocurrida con el *severe acute respiratory syndrome coronavirus 2* (SARS-CoV-2) es más importante que nunca la existencia de plataformas de datos eficaces. El volumen de información generado es monumental y la eficacia en el análisis y difusión de los datos es crítica. No solo han crecido los datos en los laboratorios con los tests a la población, sino que también han aparecido nuevos mecanismos de control del movimiento de individuos, se han ideado nuevas vías de sensorización para identificar vectores infecciosos, brotes y focos de infección.

Una plataforma como la planteada en este trabajo sería capaz de sustentar eficaz-

mente un ecosistema como el descrito. Cada nuevo sensor puede ser integrado perfectamente como publicadores en los terminales del evento al que estén asociados, los nuevos eventos que pudieran aparecer son declarados en el modelo de la aplicación con el DSL correspondiente.

Cada análisis aparecería en la forma de *adapter* de eventos, generando nuevos eventos con la transformación de la información pertinente. Los destinatarios de la difusión de estos datos no serían más que suscriptores a los terminales de comunicación de la aplicación, pudiendo esos módulos tener programada cualquier vía de difusión que podamos imaginar ofreciendo, por ejemplo, la posibilidad de disponer de todos los datos en tiempo real.

La plataforma sería escalable con facilidad y tendría flexibilidad para adoptar futuros eventos, sensorizaciones, análisis o visualizadores de datos con gran facilidad. El framework *Intino* y su *mde* permitirían una actualización rápida del *datahub* para adoptar todo nuevo elemento que se pudiera requerir.

Esta pandemia no es un proceso aislado, es un acontecimiento que ya se ha dado en el pasado y que es bastante probable que vuelva a darse en el futuro. Es imprescindible que aprendamos de esta situación y mejoremos adecuadamente nuestra infraestructura y protocolos para reducir el impacto de una pandemia futura.

Bibliografía

[1] Leonid Kalinichenko, A. Fazliev, E. Gordov, Kiselyova Nadezhda, Dana Kovaleva, Oleg Malkov, Igor Okladnikov, Nikolay Podkolodny, Natalya Ponomareva, Alexey Pozanenko, Sergey Stupnikov, and Alina Volnova. New data access challenges for data intensive research in russia. 10 2015.

[2] João Carriço, Mirko Rossi, Jacob Moran-Gilad, Gary Domselaar, and Mário Ramirez. A primer on microbial bioinformatics for non-bioinformaticians. *Clinical Microbiology and Infection*, 24, 01 2018. doi: 10.1016/j.cmi.2017.12.015.

[3] Okyay Kaynak and Shen Yin. Big data for modern industry: Challenges and trends [point of view]. *Proceedings of the IEEE*, 103:143–146, 02 2015. doi: 10.1109/JPROC.2015.2388958.

[4] Jacques Bughin, Michael Chui, and James Manyika. Clouds, big data, and smart assets: Ten tech-enabled business trends to watch. *McKinsey Quarterly*, 56:75–86, 01 2010.

[5] Andrea De Mauro, Marco Greco, and Michele Grimaldi. A formal definition of big data based on its essential features. *Library Review*, 65:122–135, 03 2016. doi: 10.1108/LR-06-2015-0061.

[6] David Bates, Suchi Saria, Lucila Ohno-Machado, and Anand Shah. Big data in health care: Using analytics to identify and manage high-risk and high-cost patients. *Health affairs (Project Hope)*, 33:1123–31, 07 2014. doi: 10.1377/hlthaff.2014.0041.

[7] AEMPS. AEMPS agencia española de medicamentos y productos sanitarios. `https://www.aemps.gob.es/informa/notasinformativas/laaemps/2019/creada-la-red-de-laboratorios-para-luchar-contra-microorganismos-resistentes-a-los-antibioticos/`. Accessed: 2019-12-12.

[8] Ms Komal. A review paper on big data analytics tools. 05 2018.

[9] A. Agarwal, Ketoki Kapila, and Kumar S. Whonet software for the surveillance of antimicrobial susceptibility. *Medical Journal Armed Forces India*, 65:264–266, 07 2009. doi: 10.1016/S0377-1237(09)80020-8.

[10] Mariaconcetta Bilotta, Giuseppe Tradigo, and Pierangelo Veltri. *Bioinformatics Data Models, Representation and Storage*, pages Pages 110–116. 01 2019. ISBN 9780128096338. doi: 10.1016/B978-0-12-809633-8.20410-X.

[11] Nicholas Luscombe, Dov Greenbaum, and M Gerstein. What is bioinformatics? a proposed definition and overview of the field. *Methods of information in medicine*, 40:346–58, 02 2001. doi: 10.1055/s-0038-1634431.

[12] J.M. Andrews. Determination of minimum inhibitory concentration. *The Journal of antimicrobial chemotherapy*, 48 Suppl 1:5–16, 08 2001. doi: 10.1093/jac/dkf083.

[13] Alberto Silva. Model-driven engineering: A survey supported by a unified conceptual model. *Computer Languages, Systems & Structures*, 20, 06 2015. doi: 10.1016/j.cl.2015.06.001.

[14] Martin Fowler. Event sourcing. `https://martinfowler.com/eaaDev/EventSourcing.html`. Accessed: 2020-05-19.

[15] Pwint Khine and Zhao Wang. Data lake: A new ideology in big data era. 12 2017.

[16] Riverbank Computing. Pyqt - python v2 and v3 bindings for the qt company's qt application framework. `https://www.riverbankcomputing.com/static/Docs/PyQt5/`. Accessed: 2020-05-17.

[17] The Qt Company. Qt - crossplatform application framework. `https://www.qt.io/`. Accessed: 2020-05-17.

[18] Brendan Collins. Big data and health economics: Strengths, weaknesses, opportunities and threats. *PharmacoEconomics*, 34, 06 2015. doi: 10.1007/s40273-015-0306-7.

[19] Open source NumFocus sponsored. pandas - data analysis and manipulation tool. `https://pandas.pydata.org/`, . Accessed: 2020-05-17.

[20] Open source NumFocus sponsored. Numpy - fundamental package for scientific computing. `https://numpy.org/`, . Accessed: 2020-05-17.

[21] Hugobrunelì Ere, Jordi Cabot, and Frédéric Jouault. Combining model-driven engineering and cloud computing. 06 2010.