UNIVERSIDAD POLITECNICA DE VALENCIA

DEPARTAMENTO DE COMUNICACIONES

Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones

Microservicios para Big Data Genómico en la Nube

Autora: Ana Cidad Vila

Director: Juan Carlos Guerri Cebollada

Tutor externo: David Roldán Martínez

Fecha de comienzo: 1/02/2018

Lugar de trabajo: Grupo de Comunicaciones Multimedia del iTEAM

Objetivos — (Times roman 10 pt.)

El objetivo de este TFM es recopilar los requisitos necesarios para dar soporte al ciclo de vida de una aplicación basada en microservicios genómicos en la nube y, tras realizar un análisis y comparativa de las herramientas existentes en el estado del arte, proponer la solución más adecuada que permita soportar dichos requisitos para el despliegue y gestión de microservicios que abarque desde el proceso de desarrollo, la integración y entrega continua hasta la gestión del ciclo de vida de los mismos.

El resultado obtenido busca ofrecer la solución tecnológica más adecuada para entornos de investigación donde se desarrolle software bioinformático y sistemas de información para la medicina personalizada, y en particular para el grupo de Sistemas Información Genómicos del Centro de Investigación en Métodos de Producción de Software (PROS) de la UPV, proporcionando argumentos para tomar la decisión de migrar sus herramientas bioinformáticas de búsqueda de variaciones genómicas a una plataforma de microservicios, así como poder utilizar esta solución en el ámbito del caso de estudio genómico del proyecto de excelencia "Un método de producción de software dirigido por modelos para el desarrollo de aplicaciones *Big Data*" (ref: TIN2016-80811-P).

Metodología — (Times roman 10 pt.)

La metodología empleada ha consistido en (1) realizar una revisión de las herramientas y plataformas de microservicios apoyadas sobre el concepto de orquestación de contenedores y puesta en explotación de los mismos, (2) identificar y cuantificar las necesidades del *Big Data Genómico* en la Nube mediante un análisis riguroso de publicaciones científicas y realización de entrevistas a expertos en la materia, y (3) una encuesta online para validar y cuantificar el impacto de los requisitos obtenidos. A partir de toda esta información, se ha aplicado una (4) estrategia de indicios, según la cual es posible estudiar un campo de análisis dado empleando como indicios o señales las aportaciones de los expertos más prestigiosos en el campo de análisis bajo estudio, junto con un (5) análisis de datos cualitativos y cuantitativos para comparar y finalmente seleccionar las tecnologías más adecuadas para dar solución a los retos planteados.

Desarrollos teóricos realizados — (Times roman 10 pt.)

Como resultado del proceso metodológico seguido, se ha llevado una comparativa teórica de las tecnologías de contenedores y plataformas más relevantes de gestión y orquestación de microservicios. Asimismo, se ha desarrollado un análisis de las necesidades en el ámbito del desarrollo del software del dominio genómico, y se ha llevado a cabo un estudio de los proyectos bioinformáticos más destacables que hacen uso de arquitecturas de microservicios. Toda esta información ha permitido desarrollar un estudio teórico que ha dado como fruto la propuesta final de este TFM.

Desarrollo de prototipos y trabajo de laboratorio — (Times roman 10 pt.)

El trabajo desarrollado aporta conclusiones relevantes para la primera fase de un proyecto de investigación llevado a cabo en el Centro PROS, en el que se desarrollará un método de producción de software para el desarrollo de aplicaciones *Big Data* y en el que se propone una solución tecnológica para poder llevar a cabo con éxito un caso de estudio en el dominio genómico.

Resultados — (Times roman 10 pt.)

El resultado de esta investigación ha sido: (1) un estudio cualitativo y cuantitativo de los requisitos de las aplicaciones informáticas del dominio *Big Data Genómico* para trabajar con arquitecturas de microservicios y en entornos de computación en la nube, (2) una comparativa de las tecnologías de contenedores, plataformas

de gestión y orquestación de microservicios más relevantes, y (3) la selección de las tecnologías más adecuada para su uso en el domino de *Big Data Genómico* en la nube.

Líneas futuras — (Times roman 10 pt.)

Estos resultados se utilizarán en grupo de Sistemas Información Genómicos del Centro PROS de la UPV para valorar el desarrollo de nuevos servicios bionformáticos utilizando la arquitectura de microservicios propuesta, para la migración de su herramienta VarSearch de búsqueda de variaciones a microservicios, así como para el desarrollo del caso de estudio genómico del proyecto de excelencia "Un método de producción de software dirigido por modelos para el desarrollo de aplicaciones *Big Data*" (ref: TIN2016-80811-P).

Publicaciones — (Times roman 10 pt.)

No se ha generado ninguna publicación hasta la fecha de defensa del TFM.

Abstract — (Times roman 10 pt.)

Las expectativas de crecimiento de la Genómica hacen prever que pronto se convertirá en la disciplina con mayor demanda de almacenamiento de datos. Además, en la Medicina de Precisión será necesario procesar estos datos, por lo que los retos computacionales a los que se enfrenta la Genómica son todavía mayores. Así, la Genómica está absolutamente ligada al *Big Data*, dando origen al *Big Data Genómico*, que provee de los algoritmos e infraestructura necesarios para la secuenciación, manejo, almacenamiento, visualización y análisis de datos genómicos. La generación de datos genómicos crece a ritmo exponencial en una nube virtual donde es preciso procesar los datos de forma distribuida (*Cloud computing*). Estos factores han hecho proliferar los proveedores cloud y han impulsado la búsqueda de nuevos paradigmas de diseño de aplicaciones distribuidas que permitan abordar los nuevos retos que plantea este escenario.

La evolución de las arquitecturas orientadas a servicios (SOA) junto con el auge en la tecnología de contenedores ha puesto de relevancia las arquitecturas basadas en microservicios como un patrón de diseño de aplicaciones de potencial aplicabilidad en el *Big Data Genómico*. En las arquitecturas de microservicios, diferentes funcionalidades se encapsulan como diferentes servicios desplegados en contenedores que pueden ser automáticamente gestionados y actualizados para evitar trabajar sobre una aplicación monolítica, facilitando el proceso de integración y entrega continua de aplicaciones.

El objetivo de este TFM es recopilar los requisitos necesarios para dar soporte al ciclo de vida de una aplicación basada en microservicios genómicos en la nube y, tras realizar una revisión de las herramientas existentes en el estado del arte, proponer la solución más adecuada que permita soportar dichos requisitos para el despliegue y gestión de microservicios que abarque desde el proceso de desarrollo, la integración y entrega continua y la gestión del ciclo de vida de los mismos.

Autora: Ana Cidad Vila, email: ancivi@upvnet.upv.es
Director: Juan Carlos Guerri, email: jcguerri@dcom.upv.es
Tutor externo: David Roldán Martínez, email: darolmar@upvnet.upv.es
Fecha de entrega: 03-07-18

ÍNDICE

I. INTRODUCCIÓN, OBJETIVOS Y METODOLOGÍA

I.1 INTRODUCCIÓN

Está previsto que en el año 2025 se hayan secuenciado entre 100 millones y 2 millones de millones de genomas humanos, lo que generará una demanda de almacenamiento entre 2 y 40 exabytes (1 exabyte = 10^{18} byte), pudiendo superar el campo de la Genómica las expectativas de *YouTube*, *Twitter* o incluso datos astronómicos [1] [2]. Además de las necesidades de almacenamiento, en la medicina de precisión será necesario procesar estos datos de forma periódica, con el fin de mejorar la salud y curar enfermedades, por lo que los retos computacionales a los que se enfrenta la genómica son todavía mayores. Es por ello que la Genómica está absolutamente ligada al *Big Data*, no solo por las necesidades de almacenamiento, sino también por las de computación, dando origen a una especialización: el *Big Data Genómico*, que se encarga de proveer a los investigadores y a la práctica médica de los algoritmos e infraestructura necesarios para la secuenciación, manejo, almacenamiento, visualización y análisis de datos genómicos. En este ámbito se genera información y datos que crecen a ritmos exponenciales (*Big Data*) en una nube virtual donde es preciso procesar los datos de forma distribuida (*Cloud computing*). Todos estos factores han tenido como consecuencia la proliferación de proveedores cloud y han impulsado la búsqueda de nuevos paradigmas de diseño de aplicaciones distribuidas que permitan abordar los nuevos retos que este escenario plantea.

Frente al diseño monolítico, la evolución de las arquitecturas orientadas a servicios (SOA, *Service-Oriented Architecture*) junto con el auge en la tecnología de contenedores ha puesto de relevancia las arquitecturas basadas en microservicios como un patrón de diseño de aplicaciones de potencial aplicabilidad en el *Big Data Genómico*. En las arquitecturas de microservicios, diferentes funcionalidades se encapsulan como diferentes servicios desplegados en contenedores que pueden ser automáticamente gestionados y actualizados para evitar trabajar sobre una aplicación monolítica, facilitando el proceso de integración y entrega continua de aplicaciones (CI/CD, *Continuous Integration/Continuous Delivery*). Existen numerosas herramientas que pueden ser de utilidad para gestionar un despliegue basado en microservicios. Este es el caso de *VAMP*, *Rancher*, *MANTL*, *Fabric8* o plataformas de gestión de contenedores como *Mesos*, *Kubernetes*.

I.2 OBJETIVOS DEL TRABAJO FINAL DE MÁSTER

El objetivo de este TFM es recopilar los requisitos necesarios para dar soporte al ciclo de vida de una aplicación basada en microservicios genómicos en la nube y, tras realizar un análisis y comparativa de las herramientas existentes en el estado del arte, proponer la solución más adecuada que permita soportar dichos requisitos para el despliegue y gestión de microservicios que abarque desde el proceso de desarrollo, la integración y entrega continua hasta la gestión del ciclo de vida de los mismos.

El resultado obtenido busca ofrecer la solución tecnológica más conveniente para grupo de Sistemas Información Genómicos del Centro de Investigación en Métodos de Producción de Software (PROS) de la UPV, con el fin de que pueda disponer de argumentos y una propuesta tecnológica adecuada para tomar la decisión de migrar sus herramientas bioinformáticas de búsqueda de variaciones genómicas a una plataforma de microservicios, así como poder utilizar esta solución en el ámbito del caso de estudio genómico del proyecto de excelencia "Un método de producción de software dirigido por modelos para el desarrollo de aplicaciones *Big Data*" (ref: TIN2016-80811-P).

I.3 METODOLOGÍA DE TRABAJO DEL TFM.

El concepto de *Big Data Genómico* y de Servicios Genómicos en la nube unifica decisiones importantes en el ámbito tecnológico, científico, económico, operacional y estratégico. Por todo ello, en lugar de centrar el trabajo en el concepto en sí, se ha preferido enfocarlo hacia su utilidad práctica, de manera que los resultados del trabajo puedan aplicarse a situaciones reales. Los objetivos principales del estudio llevado a cabo en este trabajo han sido, por tanto:

- Identificar y cuantificar el impacto de los requisitos del *Big Data Genómico* y Servicios genómicos en la nube.
- Recopilar información sobre arquitecturas basadas en microservicios y su uso en proyectos genómicos de referencia.
- Recopilar estado del arte sobre plataformas de gestión de contenedores y de gestión de microservicios.
- Definir las características que deben cumplir este tipo de arquitecturas para su uso en entornos genómicos.
- Proponer la solución más adecuada que permita soportar dichos requisitos para el despliegue y gestión de microservicios que abarque desde el proceso de desarrollo, la integración y entrega continua y la gestión del ciclo de vida de los mismos en servicios genómicos *Big Data* en la nube.

El estudio se ha abordado siguiendo una estrategia de indicios, según la cual es posible estudiar un campo de análisis dado empleando como indicios o señales las aportaciones de los expertos más prestigiosos en el campo de análisis bajo estudio. No obstante, y con el fin de completar y reforzar los indicios se incluirán, cuando estén disponibles, datos cualitativos y cuantitativos útiles a este efecto.

Este TFM ha realizado una revisión de las herramientas y plataformas de microservicios apoyadas sobre el concepto de orquestación de contenedores y puesta en explotación de los mismos, llevando a cabo un análisis riguroso tanto de publicaciones científicas, como de literatura especializada y entrevistas (directas o indirectas a través de encuestas publicadas en formularios de *Google Forms*)

a expertos en la materia. A partir de toda esta información, se ha aplicado una estrategia de indicios, que exige el cumplimiento escrupuloso de tres reglas básicas:

1. Regla de neutralidad: los textos y autores incluidos en este trabajo se evaluarán únicamente en función de utilidad para los propósitos de este TFM.
2. Regla de oposición: si dos autores estuvieran en posiciones contrapuestas y ambas fueran relevantes, debe reflejarse este hecho, puesto que ambos forman parte del campo de análisis.
3. Regla de conexión: este TFM de investigación perseguirá relacionar textos y autores de distinta procedencia y experiencia.

Por otra parte, si la metodología se basa en el estudio de las aportaciones de expertos de reconocido prestigio, inmediatamente surge la cuestión de los criterios de identificación y selección que deben considerarse en dicho estudio. A este respecto, cabe señalar que la identificación de los autores cuyas aportaciones han sido consideradas dignas de mención en este trabajo se ha llevado a cabo bajo criterios eminentemente productivos y de relevancia, escogiendo aquellos autores y textos cuyas conclusiones han tenido mayor impacto y relevancia en su ámbito.

Incluir todo trabajo relacionado con el campo de análisis resulta inviable. Por lo tanto, y aunque parezca obvio, conviene resaltar que se han incluido únicamente aquellos trabajos útiles para la consecución de los objetivos de la investigación.

Teniendo en cuenta todas estas consideraciones, la metodología investigadora ha seguido el siguiente proceso (ver Fig. 1): en primer lugar, se ha realizado una revisión rigurosa de diferentes fuentes primarias (revistas especializadas, informes de consultoras, publicaciones científicas y otras fuentes existentes), identificando de cada publicación relevante los retos clave aplicables al desarrollo de software en entornos genómicos.

Basándonos en lo extraído de las fuentes primarias, se han realizado entrevistas a expertos en el dominio genómico, elaborando como resultado una lista de requisitos del desarrollo de software genómico.

Esta lista de requisitos se ha publicado en una encuesta online (ver ANEXO 1 – Encuesta: Necesidades del desarrollo de software Genómico), y se ha solicitado a expertos de distintas instituciones, principalmente de entornos de investigación, su validación y una valoración del 0 al 10 sobre el impacto de cada uno de los requisitos.

Con estos requisitos y una valoración de su impacto en “alto”, “medio” o “bajo”, se han analizado por un lado las tecnologías de contenedores y por otro las plataformas de gestión y orquestación de microservicios. Con cada tecnología se ha analizado el cumplimiento de cada uno de los requisitos, siendo el no cumplimiento de alguno de los de impacto “alto” suficiente para descartar una tecnología.

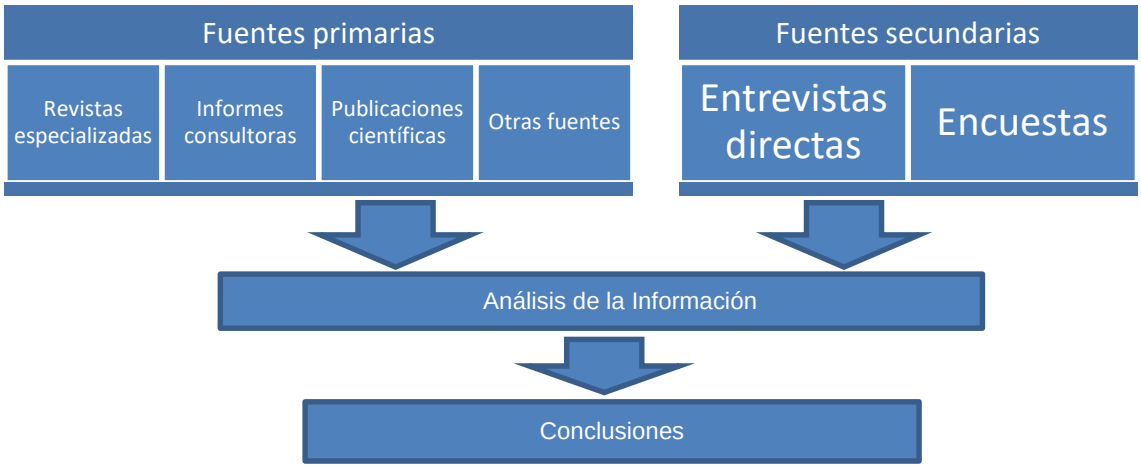


Fig. 1 Metodología de investigación empleada

I.4 ESTRUCTURA DEL TRABAJO

El contenido de esta memoria se estructura de la siguiente manera:

Capítulo I: Introducción, objetivos y metodología. En esta sección se pone el trabajo en contexto, dando una visión global del mismo, sus objetivos, metodología y estructura del trabajo.

Capítulo II: Desarrollo y resultados del trabajo. Este capítulo describe exhaustivamente el trabajo realizado en el marco de esta investigación, dividiendo el trabajo en tres secciones principales:

II.1 *Big Data Genómico* – se analizan los retos de esta nueva disciplina y los proyectos de referencia, identificando sus requisitos en el ámbito de desarrollo del software que servirán para hacer el análisis y posterior selección de la plataforma más adecuada para este dominio.

II.2 Arquitecturas de microservicios - se recopila información sobre el estado del arte de arquitecturas basadas en microservicios y de plataformas de gestión de contenedores y de gestión de microservicios.

II.3 Microservicios para *Big Data Genómico* - se razona la necesidad del uso de la nube en el entorno de la genómica, se analizan las ventajas de utilizar microservicios en el entorno bioinformático, se realiza una comparativa entre las plataformas de uso más extendido y, finalmente, se propone la solución tecnológica de microservicios más adecuada para el *Big Data Genómico* en la nube utilizando como punto de partida los requisitos del *Big Data Genómico*.

Capítulo III: Conclusiones y propuesta de trabajo futuro. Incluye las conclusiones más importantes del trabajo realizado y se proponen futuras líneas para dar continuidad al mismo.

II. DESARROLLO Y RESULTADOS DEL TRABAJO

II.1 BIG DATA GENÓMICO

Las Ciencias de la Vida se han visto afectadas por la generación de conjuntos de datos inmensos, procedentes especialmente de la llamada información ómica (genomas, transcriptomas, epigenomas y otros datos sobre células, tejidos y organismos). Los avances tecnológicos en las plataformas de secuenciación de próxima generación NGS (*Next-Generation Sequencing*) que utilizan semiconductores [3] o nanotecnología [4] han incrementado exponencialmente la velocidad a la que se generan datos biológicos en los últimos años, dando lugar a una explosión de información que se conoce con el nombre genérico de *Big Data Genómico.*

El impacto que ha tenido el *Big Data* en los más diversos sectores de la Sociedad (Comunicaciones, Medicina, Investigación, etc.) ha sido sobresaliente. Por ejemplo, en menos de 10 años el tiempo y el coste de secuenciación de un genoma se ha reducido en torno a una millonésima parte, hasta tal punto que actualmente es posible secuenciar el genoma humano por apenas un millar de euros. Y es esta información genética, precisamente, la base de la Medicina Predictiva en la que el perfil genético del paciente se emplea para determinar el tratamiento más adecuado a un diagnóstico dado.

Big Data es un término empleado para describir conjuntos de información de gran tamaño y/o elevada complejidad para los que los procedimientos convencionales de procesamiento no resultan adecuados. Probablemente, la definición más aceptada es la de las 5V de Huang, Jing, Yi y Zhen [5]: Volumen, Velocidad, Variedad, Verificación/Veracidad y Valor. Sin embargo, esta definición estará condicionada por avances tecnológicos futuros. La infraestructura *Big Data*, por otra parte, es un *framework* variopinto que incorpora componentes como Hadoop (hadoop.*Apache*.org), bases de datos NoSQL, procesamiento paralelo masivo (MPP, *Massively Parallel Processing*) y cuyo objetivo es almacenar, procesar y analizar *Big Data*. En cuanto al análisis *Big Data* abarca la recolección, manipulación y análisis de conjuntos de datos diversos y masivos que contienen una variedad de tipos de datos entre los que se incluye información genómica o electrónicos de información de pacientes [6].

En los próximos años se prevé que se requieran más recursos de computación para gestionar los datos genómicos que para *Twitter* y *Youtube*, por lo que existe una preocupación entre los biólogos y bioinformáticos que temen que su disciplina no pueda soportar el flujo de datos genómicos que está por venir [2]. Otros expertos afirman que no es posible la comparación del *Big Data Genómico* con otras disciplinas *Big Data*, aunque están de acuerdo en que las necesidades de computación de la Genómica serán enormes y que irán creciendo a medida que los costes de secuenciación del genoma disminuyan. Se estima que en 2025 se hayan secuenciado ya entre 100 y 2.000 miles de millones de genomas humanos [1], que supondrán una demanda de almacenamiento de entre 2 y 40 exabytes (1 exabyte= 10^{18} bytes) puesto que el número de datos que deben almacenarse para un

único genoma es casi 30 veces mayor que el tamaño del propio genoma. Sin embargo, el almacenamiento es solamente una parte del problema: el procesamiento y la distribución de los datos procedentes de análisis genómicos plantean retos todavía mayores [2]. Además, si nos centramos en la Medicina Personalizada, disponer del genoma no es suficiente: para cada individuo es necesaria información adicional procedente de otras -ómicas con el fin de comparar los distintos estados de una enfermedad [7].

II.1.1 Retos en la manipulación de datos genómicos

Aunque se han estudiado alrededor de más de 6.000 desórdenes mendelianos a nivel genético, la Comunidad Científica no ha sido capaz de acordar unánimemente su influencia en la salud y las enfermedades [8], y ello a pesar de que la cantidad de información disponible ha crecido enormemente en los últimos años. Por ejemplo, una de las bases de datos del NCBI (*National Center for Biotechnology Information*) es la SRA (*Sequence Read Archive*), que pone a disposición pública información de secuencias biológicas en bruto y de alineamientos procedentes de plataformas de secuenciación. En los últimos ocho años, el tamaño de la SRA ha sufrido un incremento exponencial [9], como muestra la Fig. 2.

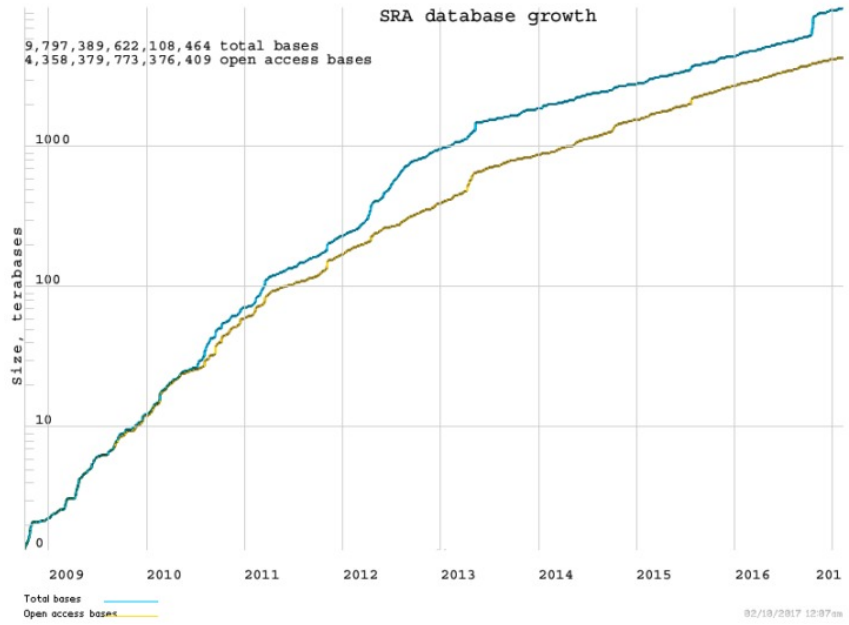


Fig. 2 Crecimiento de la base de datos SRA [9]

A pesar de que el desarrollo de las tecnologías NGS (*Next Generation Sequencing*) ha facilitado la secuenciación de un genoma o un exoma completos, hoy en día todavía existen retos importantes en términos de gestión, análisis e interpretación de información genómica. Llegados a este punto, conviene establecer algunos órdenes de magnitud (ver Fig. 3). Un genoma humano está constituido por alrededor de tres mil millones de pares de bases y su secuenciación ocupa más de 100 GB en formato BAM o 1 GB en formato VCF.

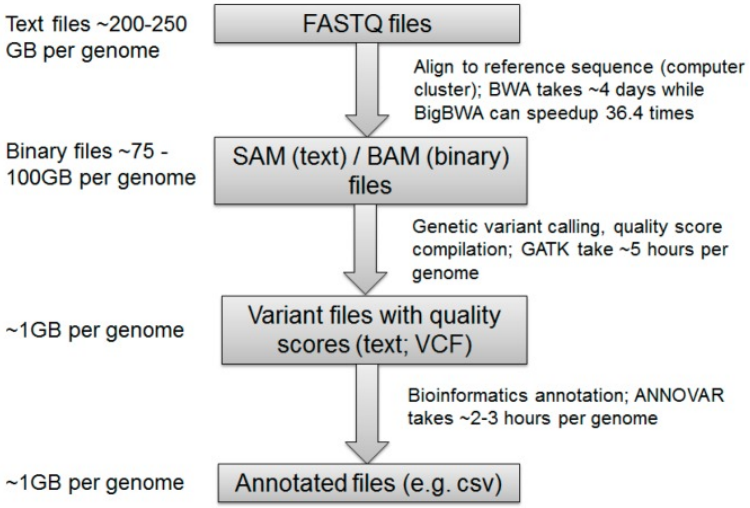


Fig. 3 Almacenamiento del genoma humano [9]

Las infraestructuras *Big Data* facilitan enormemente el análisis de estos datos. Por ejemplo, la versión *Big Data* del alineador Burrows-Wheeler multiplica su velocidad por 36 comparado con su versión original [10].

En efecto, la información médica que se produce es cada vez mayor y se genera a un ritmo exponencial. Tanto es así que se estima que el conocimiento médico se duplica cada 5 años y en el 2020 habrá 200 veces más información por especialidad de la que ningún médico pueda absorber, con lo que ya es prácticamente imposible estar al día, lo que induce a dudas y a aumentar la variabilidad en la práctica clínica. Otras cifras destacables son los 265.000 millones de euros que se podría ahorrar el sistema de salud norteamericano implantando correctamente esta tecnología [11], los 25.000 petabytes de información que se manejarán en el sector en 2020 [12], los 30.000 millones de euros del tamaño estimado para la industria del *Big Data* en la salud en 2022 [13].

Actualmente, hay grandes empresas farmacéuticas usando *Big Data* para desarrollar sus fármacos, incluso colaborando con gobiernos. Por ejemplo, el gobierno japonés ha establecido alianzas estratégicas con compañías farmacéuticas locales con el fin de abaratar a corto-medio plazo la producción de fármacos a partir del análisis cruzado de millones de datos procedentes de ensayos clínicos, estudios y publicaciones científicas del todo el mundo.

Compañías como Boehringer Ingelheim, ya recurren en sus estudios a los datos que se generan con la práctica clínica real (*Real World Data*) para avalar los beneficios de sus novedades terapéuticas. Un estudio comparó el uso de recursos entre pacientes tratados por primera vez con *Pradaxa* (*Dabigatran*) y otros con *Warfarina*, con datos de un registro de Estados Unidos, en el que participaron 3.890 pacientes con fibrilación auricular no valvular [14]. Con técnicas de *Big Data* se podría haber ampliado el estudio a millones de pacientes con enfermedades muy prevalentes como diabetes, hipertensión, hipercolesterolemia o migraña.

Bayer es otra de las compañías que ha visto interesante el analizar datos clínicos de la vida real. Así el estudio Xapass, cuyos resultados se presentaron recientemente ha analizado a más de 11.000 pacientes en Japón en tratamiento con *Rivaroxaban* [15].

La suiza Roche ha visto el gran potencial que se abre en este campo y recientemente ha adquirido a la empresa Flatiron Health, especializada en la creación de registros y generación de datos de vida real en pacientes con cáncer en Estados Unidos [16].

II.1.2 Retos en la manipulación de datos clínicos

El sector sanitario es una de las áreas en las que el *Big Data* está siendo especialmente relevante debido, entre otras cosas, a la implantación de la Historia Clínica Electrónica (EHR, *Electronic Health Registry*), a la explosión de las -omics (genómica, proteómica, etc.) y que la mayoría de los datos de que se disponen son no estructurados.

Fig. 4 Retos en la manipulación de datos clínicos

Las aplicaciones de *Big Data* en el sector sanitario ofrecen un alto potencial para mejorar la eficacia y eficiencia de la atención sanitarias gracias a que han hecho avanzar la Medicina Personalizada. En [11], por ejemplo, se ha estimado que si en EEUU se utilizaran tecnologías *Big Data* se generarían alrededor de 300 billones de cifras de negocio, de los cuales el 8 % supondrían una disminución en los costes de la atención sanitaria.

En España, los datos de los últimos informes SEIS [17] muestran una claro incremento del gasto público en TIC sanitarias, pasando de un presupuesto TIC por persona en el 2013 de 13,53€ a 14,60€ en el 2016 (un incremento del 8% en solo 3 años). Cada vez son más las Comunidades Autónomas

que han digitalizado sus historias clínicas [18], con lo que va a ser cuestión de tiempo que de toda España se puedan extraer datos agregados por patologías, tratamientos, franjas de edad, sexo, etc. El poder procesar esos datos, agregarlos e interpretarlos nos abre un escenario nuevo que nos permitirá hablar de reembolso de fármacos condicionados a los resultados en salud o a los posibles ahorros de costes que se generen por su introducción en el sistema sanitario público. Esto unido a que cada vez se tienen más computados los costes de cada procedimiento quirúrgico o de cada acto médico o asistencial gracias a las herramientas que se están generalizando de contabilidad analítica hace que este proceso sea irreversible.

Al mismo tiempo, se abren grandes oportunidades gracias al aprovechamiento conjunto de datos, debidamente anonimizados, por parte de las compañías farmacéuticas y las administraciones sanitarias en el campo de la investigación clínica, por ejemplo, para estudios epidemiológicos y para la gestión sanitaria, pues podríamos ser capaces de analizar cómo la variabilidad en las prescripciones o en la práctica clínica entre distintos centros sanitarios afecta a la población.

Hasta hace prácticamente diez años, la mayoría de las organizaciones clínicas de EEUU gestionaba la información de los pacientes a mano en fichas de colores. Sin embargo, en los últimos cinco años el porcentaje de utilización de registros médicos electrónicos (EHR, *Electronic Health Register*) ha crecido drásticamente en algunos países como EEUU [19]. Los datos extraídos de los EHR de cada paciente son muy variopintos y pueden incluir desde códigos internacionales de identificación de patologías (ICD, *International Classification Diseases*), medicamentos, tratamientos, así como datos de actividad nutricional o física aportados por el propio paciente. En cualquier caso, el volumen de los datos extraídos de los EHR puede llegar a ser considerable.

No obstante, no es el volumen de la información el único reto, sino que existen otros como la seguridad (los datos sanitarios son datos de nivel de seguridad alto, según la LODP Española) o la inmediatez (en situación de urgencia es necesario que la información esté disponible en tiempo real) que también deben considerarse.

II.1.3 Minería de datos Big Data en Estudios Genómicos

Además del crecimiento en la capacidad computacional y de almacenamiento de los datos sanitarios, se plantea la pregunta de si es posible extraer otra información asociada a la propia de esos datos a partir de algún tipo de análisis de la misma. Esto es lo que se conoce como minería de datos (*datamining*).

En general, el proceso de Minería de Datos tiene por objetivo descubrir patrones o relaciones interesantes en los datos y se divide en varias etapas, tal y como puede verse en la Fig. 5:

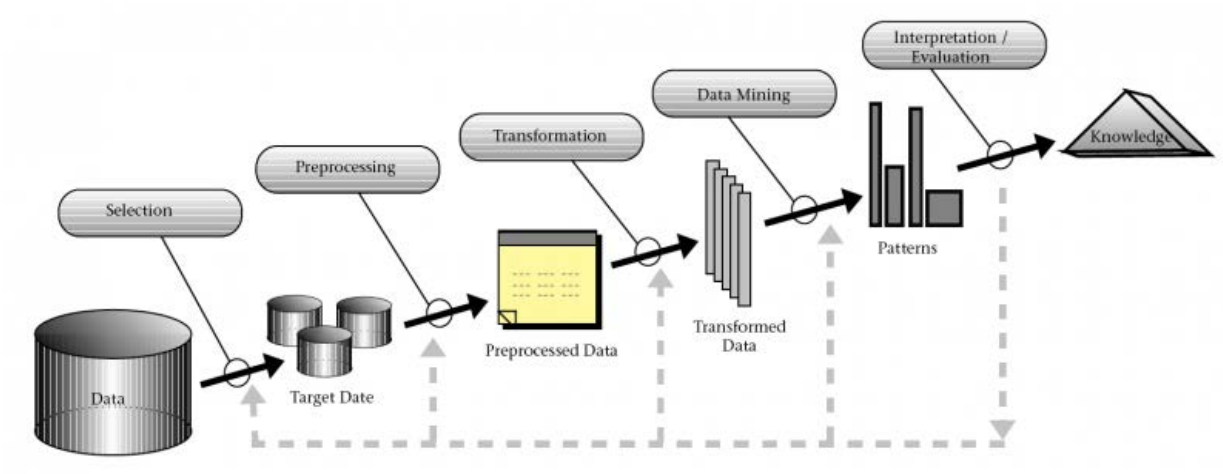


Fig. 5 Minería de Datos como uno de los pasos del Descubrimiento del Conocimiento

En el área de medicina genómica, el objetivo es inferir modelos clínicamente relevantes a partir de datos moleculares y dar así sustento a la toma de decisiones. Actualmente, los datos moleculares están disponibles en tres formas:

- Datos de genotipos, representados por un conjunto de polimorfismos de único nucleótido. Son alteraciones que ocurren en la secuencia genómica y que afectan a un único nucleótido. Dado que cada individuo posee muchos de esos polimorfismos en su genoma, su presencia forma un patrón único para esa persona.
- Datos de expresión de genes, los cuales pueden ser medidos por varias técnicas como *microarrays* de ADN o variantes de la PCR. De esta manera, es posible obtener una instantánea de la actividad de los genes en un tejido en particular para un momento dado.
- Datos de expresión de proteínas, las cuales pueden ser analizadas mediante estudios a gran escala del proteoma para brindar información sobre abundancia de proteínas específicas, variaciones y modificaciones. La información obtenida es útil para el armado de un perfil proteico característico en el diagnóstico, pronóstico y predicción terapéutica ante alguna enfermedad.

Con la creciente acumulación a nivel exponencial de diversos tipos de datos biológicos, el uso de la Minería de Datos en la Genómica a gran escala se está convirtiendo en una necesidad cada vez más buscada puesto que permite sentar las bases de la Medicina Personalizada.

II.1.4 Requisitos del Big Data Genómico

Tras el análisis exhaustivo de las fuentes primarias, y fruto de entrevistas realizadas con expertos en la materia, se han identificado los siguientes requisitos que servirán, por un lado, para motivar la necesidad de trabajar en la Nube, por otro para justificar la necesidad de trabajar con arquitecturas de microservicios, y finalmente para realizar la selección de las plataformas de empaquetado, gestión y orquestación de microservicios más adecuadas:

R1. *Open Source*

R2. Tecnologías de uso más generalizado en la comunidad bioinformática

R3. Seguridad y privacidad

R4. Gestión de grandes volúmenes de datos

R5. Reutilización

R6. Trabajo colaborativo

R7. Reproducibilidad

R8. Eficiencia/rendimiento

R9. Escalabilidad

R10. Estandarización

R11. Usabilidad / facilidad de uso

R12. Inmediatez

A continuación, se clasifica cada uno de estos requisitos, justificando su necesidad en el sector basándose en las fuentes primarias consultadas, e incluyendo una valoración del impacto (alto, medio o bajo) realizada con el apoyo de fuentes secundarias (entrevistas a expertos del sector genómico), un análisis cuantitativo y cualitativo de fuentes bibliográficas, y una valoración de los expertos realizada por medio de un encuesta publicada en *Google Forms* explicando cada requisito y solicitando una valoración entre 0 y 10 del impacto individual de cada uno de ellos. Hasta la fecha de presentación de este trabajo, se habían conseguido contribuciones de 12 grupos de investigación diferentes. Con estas aportaciones se ha calculado la nota media de la Valoración de los expertos:

Id.	R1	**Tipo de Requisito**	No funcional	
Título		*Open Source*		
Justificación		En un entorno universitario y en general de investigación, es una tendencia innegable el uso de soluciones *Open Source*. Además de los motivos económicos, la CRUE-TIC [20] y la Declaración de Berlín [21] apoyan este tipo de proyectos porque ofrecen la posibilidad de acceder al código fuente, mejorarlo y participar activamente en una comunidad abierta, algo que encaja con la labor social de fomento del Conocimiento propia de una institución educativa o de investigación en general. Además la comunidad científica considera que el software de código abierto es la mejor opción para acelerar la bioinformática [22].		
Referencias		[22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36]		
Impacto		Alto	**Valoración expertos**	8,58

Id.	R2	**Tipo de Requisito**	No funcional
Título		Tecnologías de uso más generalizado en la comunidad bioinformática	
Justificación		Según crece la complejidad de las herramientas bioinformáticas, resulta difícil para los pequeños laboratorios y grupos de investigación mantener en el tiempo técnicos capaces de modificar y desarrollar posteriormente todas las fases de un	

	pipeline bioinformático. Por ello cada vez se hace más necesaria la existencia de plataformas que permitan la experimentación y el uso de una gran variedad de herramientas y algoritmos existentes, con servicios divididos en módulos gestionables e intercambiables. Por este motivo se ha valorado de forma especial, la tendencia general de la comunidad científica en el dominio de la bioinformática a la hora de escoger una plataforma de gestión y orquestación de microservicios u otra, pues elegir la plataforma más extendida garantiza la existencia de documentación, mantenimiento y evolución de los nuevos microservicios desarrollados, así como la facilidad a la hora de integrar y hacer uso de servicios existentes. Para poder tener el estado del arte en este dominio como referencia, en la sección II.3.3 se han identificado los proyectos de referencia más destacables junto con las tecnologías que emplean.		
Referencias	[37] [38] [39] [24] [40] [25] [26] [27] [41] [42] [43] [44] [45] [46] [47] [48] [30] [36] [32]		
Impacto	Alto	**Valoración expertos**	8,45

Id.	R3	**Tipo de Requisito**	No funcional	
Título		Seguridad y privacidad		
Justificación	El incremento de la disponibilidad de los datos genómicos tiene grandes implicaciones para la privacidad personal. El genoma tiene características esenciales que, entre otras, incluyen la asociación con riesgo de enfermedades, identificación de capacidades y revelación de parentescos, que hacen de los datos genómicos información altamente sensible. Es necesario mantener tanto la autenticidad e integridad de los datos (los datos son correctos y no se pueden alterar), como la privacidad (solo el perfil autorizado puede acceder a estos datos) [49].			
Referencias	[50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60]			
Impacto	Alto	**Valoración expertos**	8,83	

Id.	R4	**Tipo de Requisito**	No funcional	
Título		Gestión de grandes volúmenes de datos		
Justificación	Tal y como se detalla en [12] en el ámbito genómico, uno de los principales problemas que se encuentran son los grandes volúmenes de información que hay que compartir/transferir y la gran cantidad de recursos de computación necesarios para manejarlos.			
Referencias	[1] [2] [6] [9] [11] [12]			
Impacto	Alto	**Valoración media expertos**	8,66	

Id.	R5	**Tipo de Requisito**	No funcional
Título		Reutilización	
Justificación	En el ámbito de las ciencias de la vida, además del reto del *Big Data*, la principal dificultad se encuentra en que la mayoría de investigadores biomédicos no disponen da capacidad para realizar por sí mismo análisis de los grandes conjuntos		

	de datos actuales utilizando las herramientas e infraestructura computacional adecuada, de forma que pueda ser completamente comprendido y reutilizado por otros. Es un requisito en este ámbito, especialmente por los retos introducidos por la secuenciación NGS y para no "reinventar la rueda", el disponer de plataformas y metodologías que permitan reutilizar herramientas desarrolladas por la comunidad bioinformática [40].		
Referencias	[37] [24] [25] [26] [27] [61] [42] [43] [30] [31] [36] [33] [34] [35]		
Impacto	Alto	**Valoración media expertos**	8,5

Id.	R6	**Tipo de Requisito**	No funcional
Título		Trabajo colaborativo	
Justificación	En la investigación avanzada en general, y especialmente en comunidades de software libre e investigación en abierto (*open research*), se requiere una gran interacción y colaboración. Con el fin de alcanzar objetivos de investigación más ambiciosos y de mayor impacto internacional, además de compartir información, es necesario poder trabajar de forma colaborativa en nuevos algoritmos para su mejora y compartición [62]		
Referencias	[37] [38] [24] [25] [26] [27] [61] [28] [42] [43] [30] [31] [36] [33] [34] [35] [62]		
Impacto	Medio	**Valoración media expertos**	7,75

Id.	R7	**Tipo de Requisito**	No funcional
Título		Reproducibilidad	
Justificación	El campo actual del análisis de la secuenciación NGS está en constante fluctuación, y no existe un amplio consenso sobre lo que son las "buenas prácticas". En esta situación es especialmente importante poder reutilizar y adoptar diferentes aproximaciones analíticas publicadas en la literatura, pero en las actuales publicaciones de análisis computacionales no existen suficientes detalles para llevarlos a cabo [40]. Para reproducir el experimento computacional más simple es necesario tener como mínimo acceso a los datos primarios y tener acceso al software y su versión, los parámetros de configuración y el nombre del genoma de referencia, y con frecuencia no toda esa información aparece en las publicaciones científicas de referencia. Es una necesidad disponer de herramientas que permitan reproducir los experimentos de una forma sencilla, fiable y transparente para el investigador.		
Referencias	[40] [45] [43] [37] [41] [28] [61] [42] [63]		
Impacto	Medio	**Valoración media expertos**	8

Id.	R8	**Tipo de Requisito**	No funcional
Título		Eficiencia/rendimiento	
Justificación	Considerando las necesidades de computación necesaria para procesar volúmenes de información tan grande y en la actualidad, los análisis genómicos actuales conllevan periodos de tiempo muy grandes. Es de gran relevancia elegir la		

	plataforma que realice una mejor gestión de los recursos software y hardware. La eficiencia o mejora en el rendimiento se traducirá en un ahorro de costes y tiempos. [64].		
Referencias	[64] [46] [65] [31] [36] [33] [34] [35] [30] [31] [32]		
Impacto	Medio	**Valoración media expertos**	7,91

Id.	R9	**Tipo de Requisito**	No funcional
Título		Escalabilidad	
Justificación	Uno de los retos que han surgido de la revolución del *Big Data Genómico*, es el de desarrollar herramientas de computación escalables que puedan dar soporte al crecimiento exponencial del flujo de datos masivos generados. Además, con el objetivo de poder hacer un uso versátil, las herramientas deben ser escalables, y ofrecer la capacidad adecuada de dar soporte a configuraciones tanto sencillas y complejas [66].		
Referencias	[44] [45]		
Impacto	Medio	**Valoración media expertos**	8,5

Id.	R10	**Tipo de Requisito**	No funcional
Título		Estandarización	
Justificación	La implementación clínica de las tecnologías NGS requiere la estandarización e integración de las bases de datos y "*pipelines*" de análisis, así como de herramientas informáticas para la toma de decisiones. El uso de herramientas que utilicen estándares ampliamente aceptados por la comunidad, garantiza su compatibilidad con desarrollos existentes y futuros, lo que ayudará en general a impulsar el sector bioinformático [67] .		
Referencias	[39] [23] [24] [44] [45] [30]		
Impacto	Medio	**Valoración media expertos**	8,16

Id.	R11	**Tipo de Requisito**	No funcional
Título		Usabilidad / facilidad de uso	
Justificación	Mejorar la usabilidad de los recursos bioinformáticos permite a los investigadores encontrar, interactuar, compartir, comparar y manipular información relevante de una forma más eficiente y efectiva. Esto redunda en una mejor percepción de los procesos biológicos con el potencial último de conseguir nuevos resultados de investigación. Las barreras de usabilidad pueden generar importantes obstáculos por no satisfacer la experiencia de usuario y forzar a los investigadores a emplear tiempo y esfuerzos innecesarios para completar sus tareas. En un entorno en que el número de bases de datos crece, así como la diversidad de usuarios, es necesario elegir las soluciones tecnológicas que cumplan los más altos estándares de usabilidad [68].		
Referencias	[40] [41] [23] [24] [42]		
Impacto	Bajo	**Valoración media expertos**	7,54

Id.	R12	**Tipo de Requisito**	No funcional	
Título		Inmediatez		
Justificación		En situación de urgencia es necesario que la información clínica de un paciente esté disponible en tiempo real [18]. Aunque en los análisis de datos genómicos suele ser un proceso lento y en la actualidad no se espera esta inmediatez, sí que lo es para los sistemas que también traten con información clínica.		
Referencias		[69] [70]		
Impacto		Bajo	**Valoración media expertos**	7,16

II.2 ARQUITECTURAS DE MICROSERVICIOS

La explosión de los dispositivos móviles y de la llamada Internet de la Cosas (IoT, *Internet of Things*) ha puesto encima de la mesa la necesidad de aplicaciones en tiempo real prácticamente en cualquier ámbito de nuestra vida cotidiana. En efecto, hoy en día existe un requisito básico: el acceso inmediato a información en tiempo real e incluso anticipadamente a las necesidades del usuario.

Desde el punto de vista de las aplicaciones, un entorno de este tipo plantea los siguientes requisitos:

- Incorporación, almacenamiento y procesamiento de grandes cantidades de información compleja en tiempo real.
- Necesidad de responder rápidamente y de adaptarse a las dinámicas cambiantes del mercado y de las necesidades del usuario.
- Necesidad de encontrar mecanismos eficientes de construir y desarrollar aplicaciones.
- Garantizar que se cumplen las expectativas de seguridad, calidad de servicio y disponibilidad.

Como consecuencia, se han replanteado los paradigmas de desarrollo de software haciendo especial hincapié en las arquitecturas, los procesos de desarrollo y los procesos de puesta en producción. En esta línea, las arquitecturas basadas en microservicios constituyen un patrón emergente para el desarrollo de aplicaciones distribuidas basadas en la entrega ágil y el despliegue escalable, tanto en ubicaciones locales como en la nube.

La principal característica de una aplicación basada en microservicios es la flexibilidad respecto de las "antiguas" aplicaciones monolíticas. Básicamente, se trata de descomponer el monolito en componentes mucho más sencillos (microservicios) y especializados que se comuniquen entre sí. Estos servicios (o microservicios) se desarrollan y despliegan de manera independiente y son responsabilidad de equipos centrados casi exclusivamente en ellos. El resultado es una disminución de la dependencia entre los componentes de la aplicación, lo que simplifica a los desarrolladores construir, mejorar y escalar partes concretas de la aplicación sin poner en riesgo el conjunto de la

misma. Los equipos de trabajo se centran en desarrollo o en el refinamiento de funciones específicas en lugar de intentar abarcar la aplicación completa, abriendo la puerta a actualizaciones diarias (e incluso más frecuentes). Adicionalmente, un enfoque orientado a microservicios también tiene su impacto en las pruebas del software ya que las convierte en más sencillas y más rápidas.

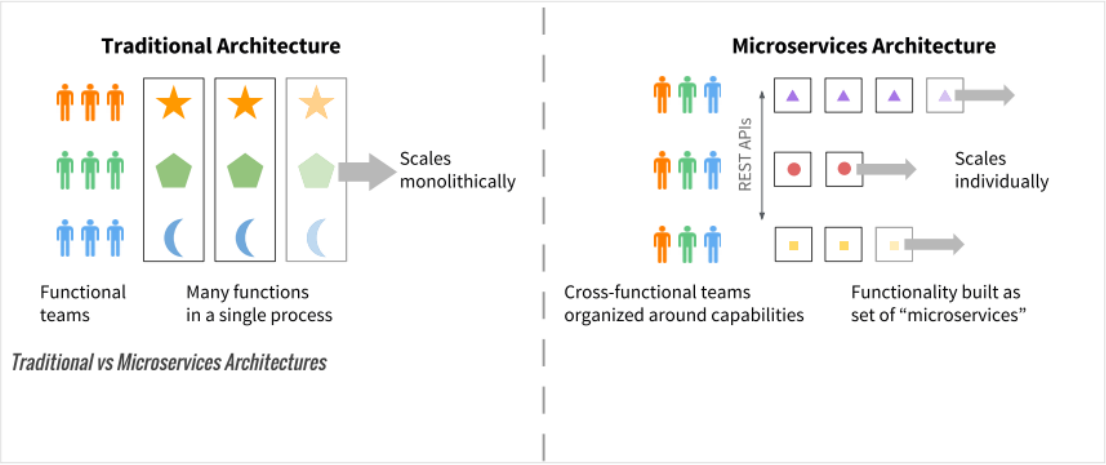


Fig. 6 Arquitectura monolítica tradicional vs Arquitectura de microservicios [71]

Las compañías líderes en Internet, como *Netflix*, *Facebook*, *Google*, *Twitter*, *Apple* y *Yelp* ha comprendido estas nuevas necesidades y ya han migrado todas sus aplicaciones corporativas a microservicios.

II.2.1 Microservicios

El paradigma de microservicios, como hemos dicho anteriormente, consiste en la concepción de una aplicación como un conjunto de microservicios especializados que se comunican entre sí, generalmente, a través de HTTP [72]. Cada microservicio se centra alrededor de una (o unas pocas) funcionalidades de negocio y es autónomo del resto, es decir, que puede desarrollarse, desplegarse y gestionarse de manera independiente al resto de microservicios de la arquitectura [71].

En general, las características generales de una aplicación basada en microservicios son las siguientes [73]:

1. *Domain Driver Design* - DDD (*Domain Driver Design*) es un enfoque de desarrollo de software basados en tres premisas:
   - Centrar los esfuerzos en el núcleo y en la lógica del dominio.
   - Basar los diseños complejos en un modelo.
   - Interacción muy estrecha entre el equipo de desarrollo del proyecto y los expertos en el dominio con el fin de identificar claramente los conceptos fundamentales de dicho dominio.
2. Principio de Responsabilidad Única - El Principio de Responsabilidad Única establece que cada microservicio es responsable de una funcionalidad. En general, y más aún en un enfoque monolítico, resulta muy sencillo que el grado de especialización de los componentes de la aplicación

sea bajo y que haya componentes que abarquen demasiadas funcionalidades. En estas condiciones, es más sencillo que haya motivos para migrar o cambiar uno de estos componentes. Al limitar el ámbito de los microservicios, este peligro de inestabilidad se reduce.

3. Interfaces públicas - La interfaz de cada microservicio (productor) es bien conocida y pública y es sobre la que se basa el cliente (consumidor) para hacer uso de la funcionalidad soportada por dicho microservicio. De este modo, es posible evolucionar la implementación del microservicio siempre y cuando la interfaz se conserve. Esta concepción hace que los cambios en la implementación de un microservicio tengan un impacto mínimo en el resto de la aplicación o en otras aplicaciones que lo consuman.

4. Principio de independencia DURS - La independencia DURS (*Deploy, Update, Replace, Scale*) establece que cada microservicio puede desplegarse, actualizarse, reemplazarse y escalarse de manera autónoma al resto.

Esta división modular, por otra parte, exige también un cambio en la concepción de la organización del desarrollo del software. Según la Ley de Conway, cualquier organización producirá sistemas cuya estructura es un fiel reflejo de la estructura de comunicación de dicha organización.

En un paradigma monolítico tradicional, los grupos de desarrollo suelen ser muy horizontales, es decir, hay un grupo encargado en la interfaz de usuario, un grupo encargado de la lógica de negocio, un grupo encargado del acceso a los datos, etc. De esta manera, cambio sencillo puede requerir la comunicación de todos los grupos de desarrollo y la toma de decisiones conjunta, con la consiguiente inversión en tiempo y recursos que esto supone. El resultado es un sistema con una arquitectura rígida.

Por el contrario, en un entorno de microservicios en los que un equipo de desarrollo transversal se ocupa de todo el ciclo del microservicio, desde su desarrollo hasta su puesta en producción y gestión, los equipos son interfuncionales y más reducidos y, por tanto, la toma de decisiones es más ágil.

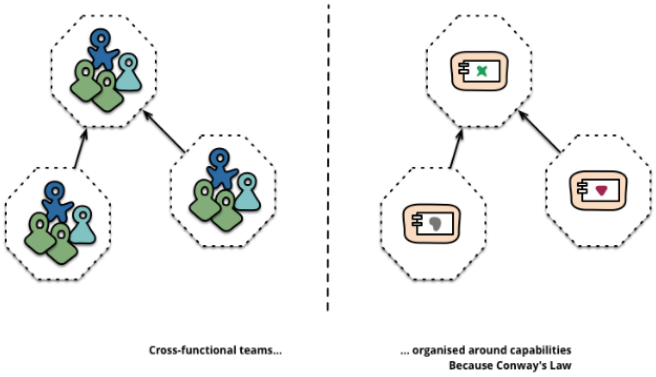


Fig. 7 Fronteras entre servicios reforzados por las fronteras entre equipos [71]

5. Puntos finales inteligentes - En un paradigma de microservicios el grueso del esfuerzo de desarrollo no se centra en la comunicación entre microservicios sino en optimizar la implementación

de cada microservicio, que es lo que añade valor y se centra en la resolución del problema del dominio que aborda la aplicación. Esto es lo que se denomina "tuberías tontas" (*dumb pipes*).

6. Gestión de datos descentralizada - Cada microservicio es responsable de sus datos, ya sea accediendo a distintas instancias de la misma fuente de datos o a otra fuente de datos, con la misma tecnología o una diferentes. Este enfoque recibe el nombre de Persistencia Políglota (*Polyglot Persistence*).

7. Automatización - La entrega continua es una práctica de desarrollo de software mediante la cual se crean, prueban y preparan automáticamente los cambios en el código y se entregan para la fase de producción. Se amplía la integración continua al implementar todos los cambios en el código en un entorno de pruebas y/o de producción después de la fase de creación. Cuando la entrega continua se implementa de manera adecuada, los desarrolladores disponen siempre de un artefacto listo para su implementación que se ha sometido a un proceso de pruebas estandarizado.

Con la entrega continua, todos los cambios en el código se crean, se prueban y se envían a un entorno de almacenamiento o pruebas de no producción. Pueden efectuarse varias pruebas al mismo tiempo antes de la implementación en producción. En el último paso, el desarrollador aprueba la actualización para su envío a producción cuando está listo. El proceso se diferencia de la implementación continua en que en el segundo caso el envío a producción se efectúa automáticamente, sin aprobación explícita. Ver Fig. 8.

La entrega continua permite a los desarrolladores automatizar las pruebas más allá de las pruebas de unidades, por lo que pueden verificar actualizaciones en las aplicaciones en varias dimensiones antes de enviarlas a los clientes. Las pruebas pueden incluir pruebas de la Interfaz de Usuario, de carga, de integración, de fiabilidad de la API, etc. De este modo, los desarrolladores pueden validar las actualizaciones de forma más exhaustiva y descubrir problemas por anticipado. Con la nube, resulta sencillo y rentable automatizar la creación y replicación de varios entornos de pruebas, algo que anteriormente era complicado en las instalaciones.

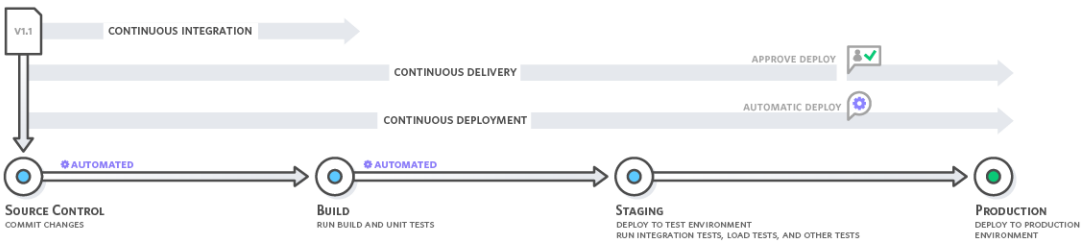


Fig. 8 Entrega continua frente a integración continua y despliegue continuo

La entrega continua automatiza todo el proceso de publicación de software. Cada revisión efectuada activa un proceso automatizado que crea, prueba y almacena la actualización. La decisión definitiva de implementarla en un entorno de producción en vivo la toma el desarrollador. Con la implementación continua el despliegue continuo, las revisiones se implementan en un entorno

de producción automáticamente sin la aprobación explícita del desarrollador, con lo que se automatiza todo el proceso de publicación de software [74] [75] [76].

El desarrollo de software a medida es un esfuerzo intensivo en recursos que exige la colaboración de un grupo heterogéneo de expertos en varias disciplinas, ya que deben participar programadores y analistas, pero también biólogos, médicos, genetistas, etc. Otra opción es adquirir software comercial, que suele ser caro y demasiado general.

El empleo de soluciones basadas en contenedores disminuye las barreras de adopción, incrementa la eficiencia del desarrollo, simplifica la reutilización de módulos de código probados y simplifica la verificación de los algoritmos. Estas ventajas todavía se potencian más si se opta por soluciones *open source*, en las que el código fuente está disponible para observarlo, aprender, mejorarlo y contribuirlo. Por este motivo, entre otros específicos del dominio en cuestión [22], en este TFM nos centramos exclusivamente en soluciones *open source*.

II.2.2 Contenedores

Un requisito básico de las aplicaciones de microservicios es que el despliegue tiene que ser necesariamente ágil. A pesar de que los microservicios podrían gestionarse con máquinas virtuales (VM, *Virtual Machine*) tradicionales, se prefieren tecnologías de contenedores como medio para simplificar el empaquetado y el despliegue tanto de aplicaciones monolíticas como de aplicaciones basadas en microservicios.

Los desarrolladores empaquetan el sistema operativo de la aplicación, las dependencias y los ficheros de configuración en un único paquete (o contenedor) que puede accederse desde un repositorio y desplegarse en casi cualquier servidor. De esta manera, se centran mucho más en la lógica de la aplicación y mucho en menos en cómo se ejecutará. Además, los contenedores son más rápidos de gestionar que las máquinas virtuales y soportan una mayor carga de trabajo.

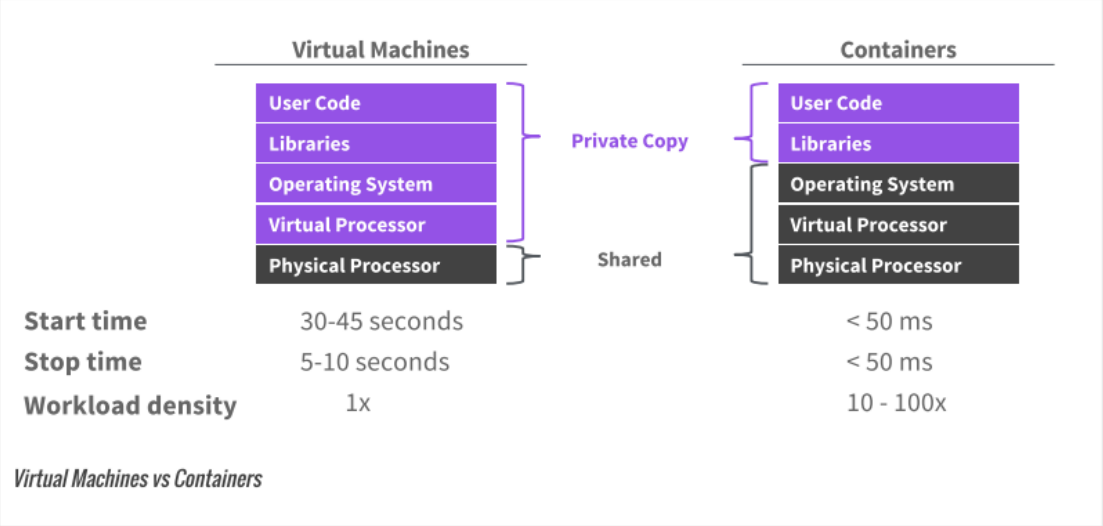


Fig. 9 Máquinas virtuales vs Contenedores

Los contenedores son especialmente importantes en entornos de computación distribuida en la nube debido a su portabilidad (son independientes de la plataforma y se puede lanzar tanto en un entorno Linux como en un entorno Windows, por ejemplo). Las mayores cargas de trabajo y la menor sobrecarga de computación se traducen en un menor gasto a la hora de distribuir los sistemas en la nube.

La Tabla 1 resume las principales tecnologías de contenedores disponibles actualmente:

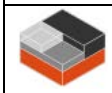
	*Docker* es una herramienta *open source* que automatiza el despliegue de aplicaciones dentro de contenedores software proporcionando una capa adicional de abstracción y automatización de la virtualización a nivel de sistema operativo en Linux
	*Apache Mesos* es un núcleo de sistemas distribuidos *open source* que abstrae el clúster completo en un único recurso de computación, utilizando grupos de contenedores Linux para aislar el consumo de recursos.
	*Rocket* (o rkt) desarrollado por CoreOS, es un gestor de contenedores para clúster Linux que pone especial hincapié en la seguridad, simplicidad y flexibilidad en arquitecturas en clúster.
	App Container (aapc) es una especificación *open source* que define cómo tienen que ejecutarse las aplicaciones en contenedores: un formato de imagen, entorno de ejecución y protocolo de descubrimiento.
	Linuxcontainers.org es un proyecto paraguas de LXC, LXD, LXCFS y CGManager. Su objetivo es ofrecer un entorno independiente del vendedor para el desarrollo de contenedores Linux
	OpenVX es entorno de virtualización basado en contenedores Linux que crea múltiples contenedores seguros y aislados en único servidor físico

Tabla 1 – Principales tecnologías de contenedores

La tecnología de contenedores ha conseguido que se puedan levantar miles de instancias o servicios de una aplicación en cuestión de minutos e incluso segundos. Sin embargo, esta aceleración de la capacidad de desarrollo también plantea nuevos retos.

Es cierto que los contenedores individuales son relativamente pequeños en tamaño y complejidad comparados con las máquinas virtuales, pero gestionar un gran número de contenedores supone una complejidad operacional tremenda y el paso de desarrollo a producción puede convertirse en muy complicado. En estas condiciones, resulta indispensable un medio de planificar, gestionar, autenticar y descubrir microservicios y esto es lo que se conoce como orquestación de contenedores.

II.2.3 Plataformas de gestión y orquestación de microservicios

La orquestación automatiza el despliegue y la gestión de contenedor y microservicio a escala, creando así las bases para nuevos *frameworks* de servicios útiles en entornos de *Big Data*, Internet de las Cosas y otras aplicaciones de nueva generación. Por otra parte, uno de los defectos más comunes de las plataformas de orquestación de microservicios es la gestión de los servicios sin estado, o lo que es lo mismo, la fiabilidad de ejecutar las bases de datos y los sistemas de almacenamiento en una infraestructura compartida.

Al abstraerse de la plataforma, las herramientas de orquestación permiten a los usuarios tratar un clúster completo como si fuera una única unidad de despliegue. Además, en general, se caracterizan por los siguientes aspectos:

- Características de línea base: el proceso de orquestación implica algún tipo de herramienta que automatice todos los aspectos de gestión de la aplicación y debe formar parte, sin duda, del núcleo de la herramienta de orquestación de contenedores.
- Configuración declarativa: las herramientas de orquestación proporcionan una opción a los equipos *DevOps* de declarar unas guías para la gestión de la carga de trabajo y su configuración partiendo de un esquema estándar en formato YAML o JSON. Estas definiciones contienen información crítica sobre los repositorios, la red, el almacenamiento y los ficheros de log. Esta aproximación permite a las herramientas de orquestación aplicar la misma configuración varias veces y conseguir siempre el mismo resultado en el sistema destino. También permite establecer distintas configuraciones para la misma aplicación durante las etapas de desarrollo, pruebas y producción en diferentes entornos.
- Reglas y restricciones: las cargas de trabajo suelen requerir políticas especiales en aspectos como el rendimiento o la alta disponibilidad.
- Aprovisionamiento: se trata de la negociación de la capacidad asignada a los contenedores dentro de la disponible en el sistema.
- Descubrimiento: en un despliegue distribuido compuesto por varios contenedores ejecutándose en diferentes máquinas, el descubrimiento del contenedor es crucial. Los servidores web necesitan descubrir dinámicamente a los servidores de bases de datos. De la misma manera, los balanceadores de carga deben descubrir y registrar los servidores web.
- Monitorización de la salud: puesto que las herramientas de orquestación están al tanto de la configuración deseada por el sistema, debe ser capaz de realizar un seguimiento de los contenedores y de las máquinas en que se ejecutan. Igualmente, cuando un contenedor cae, la herramienta de orquestación puede lanzar un respaldo asegurando que el despliegue siempre alcanza el estado deseado por el desarrollador o el operador.

Dependiendo de las consideraciones de cada caso concreto es posible escoger una única herramienta de orquestación o una solución más integrada. En la Tabla 2 se recogen las plataformas más empleadas en la actualidad, cada una con unas funcionalidades distintas para la planificación, gestión de recursos y gestión del servicio.

Orquestación de contenedores • Planificación • Gestión de recursos • Descubrimiento de servicios		***Docker Swarm*** proporciona capacidades de clustering nativas para convertir un grupo de máquinas *Docker* en una única, máquina *Docker* virtual
		***Kontena*** es una plataforma de contenedores y microservicios construida sobre *Docker*

		***Nomad*** es un planificador distribuido, de alta disponibilidad, organizado en centros de datos y regiones, que gestiona clústeres de máquinas y corre aplicaciones sobre ellas. También gestiona máquinas virtuales.
Plataformas de Contenedores • Gestión del ciclo de vida de contenedores		***Kubernetes*** es un sistema de código abierto para automatizar el despliegue, escalado y gestión de aplicaciones contenerizadas.
		*Rancher* de forma nativa da soporte y gestiona clústeres *Kubernetes*, *Swarm* y *Mesos*, proporciona más que solo orquestación
Sistema operativo distribuido • Operación de contenedores • Servicios de sistema distribuido • Catálogo de servicios + Ecosistema de App		*DC/OS* combina la orquestación de contenedores y la plataforma de sistema en una única plataforma. Basado en *Mesos*, incluye todo lo necesario para correr microservicios elásticamente en producción, incluyendo contenedores y servicios de datos con estado.

Tabla 2 - Principales plataformas de orquestación y gestión de microservicios

Llegados a este punto, nos parece muy conveniente subrayar el hecho de que las tecnologías y herramientas citadas en los apartados anteriores pueden colaborar entre ellas, esto es, no es necesario decantarse por una sola de ellas si no que, en ocasiones, será necesario utilizar funcionalidades complementarias de varias de ellas. Por ejemplo, un clúster de *Kubernetes* puede gestionar contenedor *Docker* o *rkt* en paralelo [77] [78] [79].

II.3 MICROSERVICIOS PARA BIG DATA GENÓMICO EN LA NUBE

II.3.1 Computación en la nube para Big Data Genómico

Gracias a los avances en la tecnología de NGS de secuenciación del genoma, el volumen de datos genómicos transferido a archivos públicos es actualmente superior al rango de multi-pentabytes (1 pentabyte = 10^{15} bytes). En la *International Cancer Genome Consortium* (ICGC), por ejemplo, grupos de 17 países ya habían manejado en el 2015 un volumen de datos de 2 pentabytes –equivalentes a 500.000 DVDs- en solo cinco años. Utilizando una conexión típica a internet de un entorno universitario, costaría más de 15 meses transferir un repositorio de este tamaño a la red local de un investigador. Además, el coste del hardware necesario para almacenarlo y procesar estos datos, costaría más de 1 millón de dólares [64].

Los servicios de computación en la nube proporcionan “elasticidad”, es decir, que un investigador puede utilizar tantos recursos de computación como necesite para completar un análisis rápido, y pagar únicamente por el tiempo de cálculo consumido. Realizando sus análisis en la nube, controlando desde sus escritorios computadores virtuales en la nube, múltiples investigadores pueden trabajar en paralelo, compartiendo sus datos y métodos con facilidad. Además, el análisis de grandes volúmenes de datos genómicos que antes llevaban meses, en la nube pueden realizarse en días o semanas.

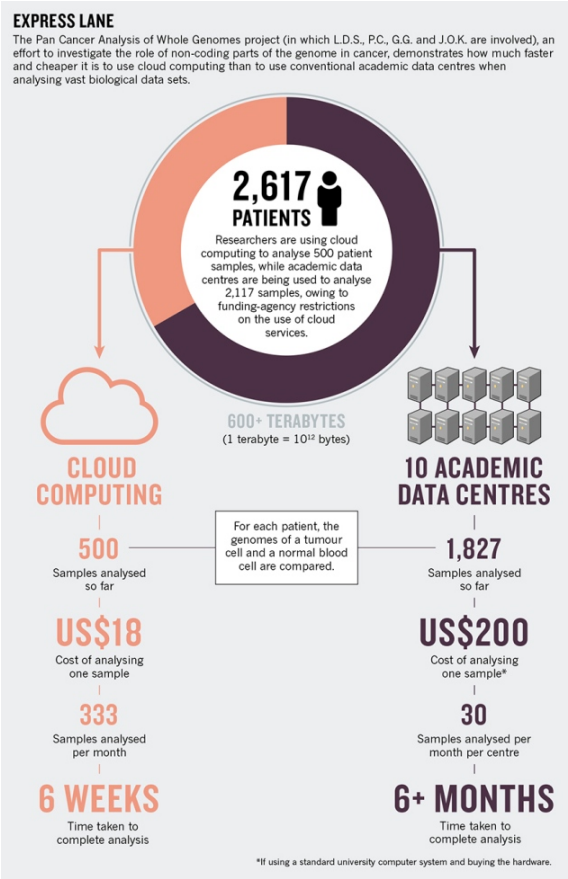


Fig. 10 Ahorro de tiempo y dinero gracias al uso de la Computación en la nube en un proyecto genómico [64]

En la actualidad, los servicios en la Nube son tan seguros como los de la mayoría de centros de datos académicos, y con frecuencia incluso más. Las principales compañías comerciales, incluyendo Amazon, Google y Microsoft, al igual que pequeñas compañías focalizadas en la investigación genómica, ofrecen estos servicios de forma segura. Estos proveedores utilizan encriptación fuerte para los datos, tienen sistemas avanzados de control de acceso, para garantizar el acceso restringido a los datos, y proporcionan herramientas para que los propietarios de los datos puedan monitorizar el uso de una forma cerrada.

Unos pocos organismos financiadores de investigación del genoma humano todavía son cautos con este tema, - por ejemplo, algunas agencias financiadoras europeas recomiendan a los investigadores almacenar sus datos dentro de la jurisdicción de sus agencias con el fin de cumplir con la legislación europea de protección de datos. Pero el ahorro económico, la flexibilidad, fiabilidad y seguridad de la computación en la nube hace prever un giro radical en el uso de estas tecnologías para el ámbito de la genómica. La decisión tomada en el 2007 por el *National Institute of Health* (NIH) de los EEUU en esta dirección, levantando las restricciones del uso de la computación en la nube para almacenar y analizar sus cientos de miles de genomas, hace esperar que esta medida se extienda en otras agencias internacionales y, de esta forma, se extienda el uso de la Nube en el ámbito del *Big Data Genómico*.

La adopción de esta visión todavía requiere trabajo desde el punto de vista técnico y legal. Por un lado, se necesitarán protocolos para autorizar el acceso a los datos sensibles en la nube, así como mecanismos para revocar el acceso. En la parte legal, se tendrán que establecer normas para establecer los roles y responsabilidades de las agencias financiadoras, los custodios de los datos, los proveedores de servicios en la nube y de los investigadores que hagan uso de los datos genómicos en la nube. Para ello la *Global Alliance for Genomics and Health* (GA4GH) [23] trabaja en proporcionar un marco político y estándares técnicos para hacer posible la compartición responsable de datos genómicos en el marco de los derechos humanos.

II.3.2 Microservicios para Big Data Genómico

El desarrollo de software a medida en el ámbito genómico es un esfuerzo intensivo en recursos que exige la colaboración de un grupo heterogéneo de expertos en varias disciplinas, ya que deben participar programadores y analistas, pero también biólogos, médicos, genetistas, etc. Otra opción es adquirir software comercial, que suele ser caro y demasiado general.

El empleo de soluciones basadas en contenedores disminuye las barreras de adopción, incrementa la eficiencia del desarrollo, simplifica la reutilización de módulos de código probados y simplifica la verificación de los algoritmos [80].

También el uso de arquitecturas de microservicios supone un incremento de la seguridad y estabilidad de las aplicaciones. Esto se consigue al eliminar la interdependencia entre componentes, y al mejorar la gestión de fallos, pues desaparece la necesidad de detener la aplicación completa en caso de fallo de un único microservicio.

Estas ventajas todavía se potencian más cuando se opta por soluciones *open source*, en las que el código fuente está disponible para observarlo, aprender, mejorarlo y poder contribuir.

En [81] además se destaca cómo el uso de contenedores, junto con *workflows* supone un incremento en la eficiencia en el procesado de los datos genómicos, pues las herramientas bioinformáticas tienen complejas dependencias y son difíciles de construir desde cero; por ello la habilidad del despliegue que aportan los contenedores en diferentes sistemas operativos y versiones disminuye el esfuerzo necesario para el análisis.

II.3.3 Proyectos genómicos de referencia

Proyectos en abierto para datamining genómico

Hemos visto que el volumen de datos disponibles para la investigación está creciendo a una tasa exponencial. En el sector que nos ocupa es relativamente novedoso, sin embargo, ya existen publicados datos en abiertos que permiten complementar los análisis de datos secundarios en los procesos de datamining genómico (apartado II.1.3). Algunos ejemplos son el proyecto YODA (*Yale Open Data Access*) a través del cual es posible acceder a datos de pruebas clínicas [37] o el proyecto

TCGA (*The Cancer Genome Atlas*) que publica datos genómicos sobre el cáncer una vez que se han hallado resultados iniciales en investigaciones relacionadas [38].

API genómicas GA4GH

La complejidad de los datos genómico, debido a su volumen y diversidad, junto con la necesidad de compartir datos, ha tenido como consecuencia la creación de numerosas API (*Application Programming Interface*) que ofrecen acceso seguro, modular e interoperable a distintas plataformas y aplicaciones [24]. En su forma más sencilla, una API es un conjunto de protocolos e instrucciones que permite que las aplicaciones se comuniquen entre sí (ver Fig. 11).

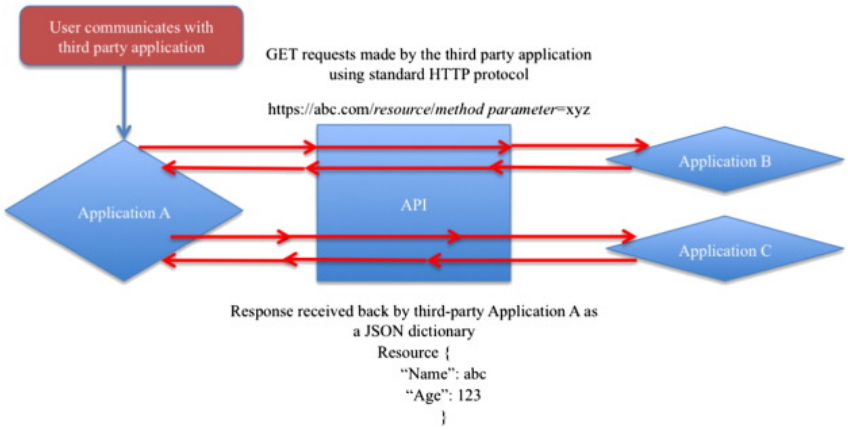


Fig. 11 Comunicación entre dos aplicaciones a través de un API [24]

Las API genómicas facilitan tanto el desarrollo de aplicaciones basadas en *Big Data Genómico* que *Global Alliance for Genomics and Health* [23] ha creado un grupo de trabajo específico para las tareas de integración y compartición de datos genómicos que ha establecido en formato estándar para la representación de los mismos así como las recomendaciones para proveer API genómicas interoperables. El *framework* proporcionado por el GA4GH permite acceder a datos de diversas fuentes utilizando una llamada a la misma API [24]. Algunos ejemplos de implementaciones de API genómicas son *Google Genomics API* (2018) [25], *SMART Genomics API* [26] o *23andMe Genomic API* [27].

Herramientas de análisis genómicos (*workflows/pipelines*)

Junto con este crecimiento masivo en volumen y disponibilidad, el análisis de los mismos datos genómicos suele exigir importantes recursos de procesamiento. Por ejemplo, resulta muy habitual que los datos producidos por las técnicas de secuenciación NGS (*Next Generation Sequencing*) necesiten de varias etapas de preprocesado como paso previo a un análisis detallado de los mismos (ver Fig. 12). Además de esta intensidad en recursos, la reproducibilidad de experimentos computacionales con estos datos es limitada debido a la complejidad del sistema y la configuración del software [40] [41]. La disponibilidad de numerosas herramientas en distintas plataformas y de bases de datos heterogéneas con estructuras diversas, dificultan sobre manera el trabajo de los profesionales de las Ciencias de la Vida. Por tanto, existe una necesidad urgente de desarrollar

soluciones de integración que ayuden a estos profesionales a ejecutar aplicaciones de manera transparente a la plataforma de utilicen.

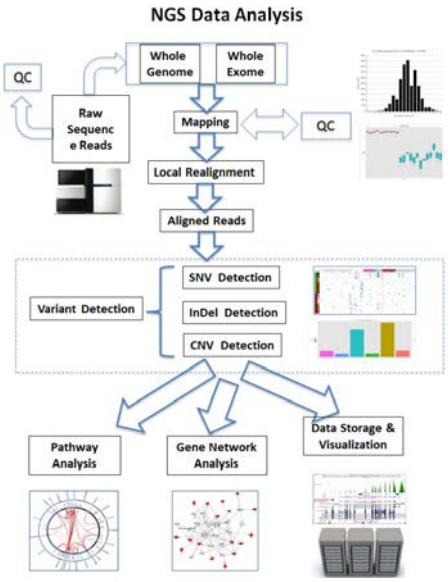


Fig. 12 *Pipeline* de análisis de datos NGS [82]

En los últimos años se han hecho algunos intentos de proporcionar frameworks que mejoraran la reproducibilidad de aplicaciones individuales y *pipelines* de análisis, como las herramientas de análisis genómico *Galaxy* [28] o SUSHI [61], pero todavía presentan restricciones de fiabilidad, especialmente en experimentos realizados en entornos con recursos limitados o en clústeres de computación [42]. Los *pipelines* o flujos de trabajo genómicos consisten en varias piezas de software de distinta procedencia (ya sea diferente proveedor o tecnología de construcción) que se conectan entre sí con un propósito determinado. Suelen ser prototipos académicos con propósitos muy concretos y frecuentemente difíciles de instalar, configurar y desplegar. En efecto, un programa implementado en un entorno dado, en general, presenta dependencias con librerías y otros componentes de dicho entorno. Como consecuencia, un *pipeline* diseñado en un entorno es poco probable que funcione sin problemas en otro distinto sin realizar un esfuerzo considerable para adaptarlo a las nuevas condiciones [43].

Para resolver estos problemas de portabilidad la mejor solución son los contenedores frente a las máquinas virtuales, como ya justificamos en la sección II.2.2, ya que permiten ejecutar paquetes aislados y autocontenidos en gran cantidad de plataformas [44] [45]. La ventaja más obvia de esta aproximación es que evita la instalación tediosa de una pléyade de programas con complejas dependencias entre sí por la descarga de una imagen preconstruida y lista para ser ejecutada que contiene toda la configuración del software necesaria. Otra ventaja de los contenedores es que ejecutan cada proceso en un contenedor aislado, evitando conflictos con otros programas instalados en la misma máquina y garantizando, por otra parte, que cada proceso se ejecuta bajo unas condiciones de configuración predecibles y que no cambian con el tiempo debido a software mal

configurado, actualizaciones del sistema o errores de programación. Por otra parte, hay estudios [46] [65] que indican que una aplicación que se ejecuta en un contenedor presenta un rendimiento igual o mejor comparadas con su ejecución en una máquina virtual tradicional. Por ejemplo, en [42] se describe cómo utilizar contenedores *Docker* para reproducir un *pipeline* de análisis de datos procedentes del TGCA para investigar AML (*Acute Myeloid Leukemia*), un tipo de cáncer que afecta a las células sanguíneas y que se caracteriza, entre otras cosas, por el crecimiento anormal de dichas células y su interferencia con las células sanas y que resulta mortal en cuestión de semanas. El *pipeline* consistía en acceder a los datos del TGCA (aproximadamente, 12 TB con los datos de secuenciación del genoma más otros 12 TB con el exoma secuenciado), identificar las variantes somáticas y, a través de varias herramientas de biosoftware, predecir la heterogeneidad del tumor. En la parte derecha de la Fig. 13, se muestra la aplicación desplegada en un servidor virtual, con el *pipeline* de análisis instalado en un *CentOS* 6.7 y ejecución manual debido a las limitaciones del software para multitarea y computación paralela, así como las limitaciones de recursos locales. Sin embargo, cuando se despliega la aplicación con *Docker* (parte derecha de la Fig. 13), el *pipeline* se lanza automáticamente y es posible escalar en función de los recursos del sistema disponibles. Además, el aislamiento de los contenedores evita problemas de dependencias y librerías.

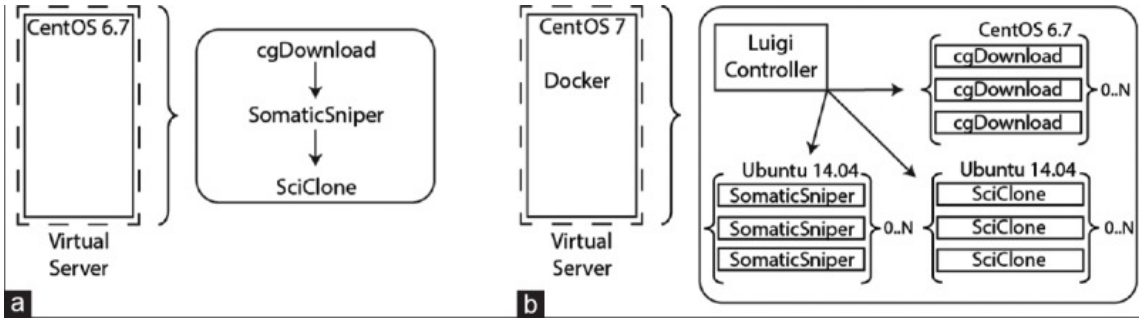


Fig. 13 Ejemplo de utilización de *Docker* para desplegar una aplicación

Otra aproximación, no excluyente de la anterior, a la automatización de tareas es el empleo de un *workflow* genómico o gestor de flujo. Consiste en la integración de herramientas bioinformáticas con bases de datos genómicas con el fin automatizar el análisis y el almacenamiento de secuencias. Algunos de los más extendidos son *Taverna* [48], *Galaxy* [28], *SnakeMake* [84] o *NextFlow* [85]. Todos ellos garantizan la trazabilidad y la reproducibilidad de los procesos, aunque continúan exigiendo una elevada curva de aprendizaje. La utilización de contenedores en este entorno puede reportar importantes beneficios, ya que simplifican el proceso de ejecución y despliegue.

Uno de los *workflows* más utilizados es *Galaxy* [39] que proporciona, además, un conjunto de utilidades adicionales que permite diseñar *pipelines* completos y compartirlos. Existen últimamente experiencias en la integración de *Docker* y *Galaxy* [29] [86] [87], con el objetivo de distribuir imágenes *Docker* de los *pipelines*. Concretamente, en [29] se integra, además, un registro para almacenar y organizar todas las herramientas de *Galaxy* empleadas (ver Fig. 14), además de gestionar el control de versiones de la misma.

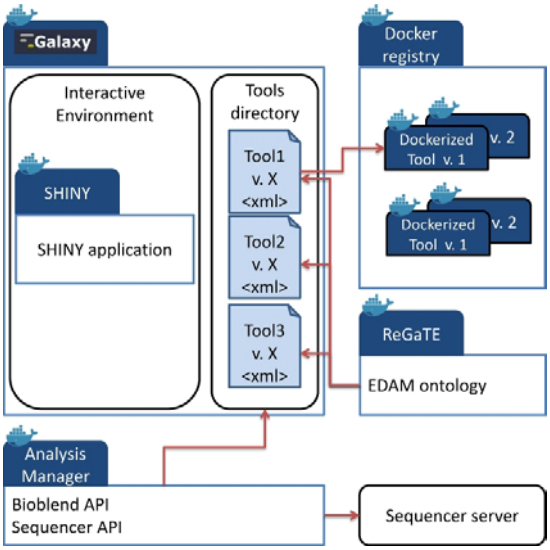


Fig. 14 Integración de *Docker* y *Galaxy* [29]

Es decir, parece que las herramientas bioinformáticas se pueden mezclar con *Docker* para construir *workflows* reproducibles y reutilizables a través de la creación de imágenes *Docker*. Tanto es así, que están proliferando repositorios de imágenes *Docker* de los que cualquier investigador puede obtener *workflows* listo para ejecutar. Algunos ejemplos son *BGDMDocker* [88], *BioContainers* [30], *CyVerse* [89], *BioShaDock* [90]. De la misma manera, existen herramientas que proporcionan imágenes *Docker* públicas como *Perl* y *BioPerl* [31] [32], *python* y *biopython* [33] [34] y *R* y *Bioconductor* [35] [36].

Por otro lado, conviene citar dos repositorios públicos de imágenes *Docker*, que son bioboxes (http://bioboxes.org) and *BioDocker* (http://bio*Docker*.org). Ambos ofrecen ejemplos explícitos de cómo incorporar en contenedores herramientas bioinformáticas desarrolladas a medida junto con otras utilidades y herramientas mantenidas por la comunidad.

Finalmente, cabe destacar como ejemplos de proyectos abiertos que tomamos como referencia para este TFM por hacer uso de *Docker* para ayudar en la gestión de *pipelines*: la plataforma *AWS PanCancer Workflow Launcher* [91], que permite lanzar *workflows* con *Amazon Web Services* utilizando *Docker*; *BaseSpace* [92], que es una plataforma de análisis y almacenamiento en la nube para datos genómicos generados por secuenciadores Illumina; y el ya mencionado *NextFlow* [85] [93], que permite generar *workflows* escalables y reproducibles mediante el uso de contenedores *Docker*.

II.3.4 Comparativa y Solución propuesta

Selección de tecnologías de contenedores:

Fruto de todo el análisis previo realizado y la consulta de las fuentes primarias y secundarias, ha sido posible valorar el nivel de cumplimiento de los distintos requisitos establecidos en la sección II.1.4,

obteniendo como resultado de dicho análisis la siguiente tabla comparativa de tecnologías de contenedores.

Id.	Requisito	***Docker***	***Mesos***	***Roquet***	**Appc**	**Linux containers**	**Open container**	**OpenVZ**
								
R1	*Open Source*[1]	+++	+++	+++	+++	++	+++	+
R2	Uso generalizado	+++	+	NO	NO	NO	NO	NO
R3	Seguridad y privacidad	++	++	Descartado por falta de utilización en el sector (no cumple R2)	Descartado por falta de utilización en el sector (no cumple R2)	Descartado por falta de utilización en el sector (no cumple R2)	Descartado por falta de utilización en el sector (no cumple R2)	Descartado por falta de utilización en el sector (no cumple R2)
R4	Gestión grandes volúmenes de datos	+++	++					
R5	Reutilización	+++	++					
R6	Trabajo colaborativo	+	+					
R7	Reproducibilidad	+++	+					
R8	Rendimiento	++	++					
R9	Escalabilidad	+++	++					
R10	Estandarización	+++	+					
R11	Usabilidad	-	-					
R12	Inmediatez	+	+					

Tabla 3 - Tabla resumen selección Tecnología de Contenedores

(Impacto: rojo = Alto, amarillo = Medio, verde = Bajo)

En esta tabla, se puede observar que se han descartado cinco tecnologías por su poca representatividad en las fuentes consultadas. Entre las dos tecnologías restantes, *Docker* y *Mesos*, el hecho de que *Docker* sea la más utilizada, y que se haya convertido en estándar de facto como tecnología de contenedores de referencia, marca la diferencia y hace recomendar su uso en proyectos *Big Data Genómico*s.

El formato del fichero *Docker* ha convertido en un estándar de facto y los principales proveedores de contenedor (entre los que se encuentran *Docker Inc.*, *Google*, *Pivotal*, *Mesosphere* y otros) formaron la CNCF (*Cloud Native Computing Foundation*) y la OCI (*Open Container Initiative*). Hoy en día, tanto la CNCF como la OCI tienen como objetivo garantizar la interoperabilidad y la estandarización de interfaces entre distintas tecnologías de contenedores con el fin de asegurar que cualquier contenedor *Docker*, construido con cualquier herramienta, se puede ejecutar en cualquier entorno o infraestructura.

[1] Apache License 2.0: Docker, Mesos, Roquet, Open Containrer; GNU LGPLv2.1+: Linux Container; GNU GPL version1: OpenVZ

Selección de plataforma de gestión y orquestación de microservicios

Id.	Requisito	***Docker Swarm***	***Kontena***	***Nomad***	***Kubernetes***	***Rancher***	***DC/OS***
							
R1	*Open Source*[2]	++++	+++	+++	+++	+++	+++
R2	Uso generalizado	+	NO	NO	+++	NO	++
R3	Seguridad y privacidad	NA	Descartado por falta de utilización en el sector (no cumple R2)	Descartado por falta de utilización en el sector (no cumple R2)	NA	Descartado por falta de utilización en el sector (no cumple R2)	NA
R4	Gestión grandes volúmenes de datos	++			+++		+++
R5	Reutilización	++			+++		+++
R6	Trabajo colaborativo	+			+++		+
R7	Reproducibilidad	++			++		++
R8	Rendimiento	+			+++		++
R9	Escalabilidad	-			++		+++
R10	Estandarización	+			++		+
R11	Usabilidad	+++			+		+
R12	Inmediatez	+			++		+

Tabla 4 - Selección de plataforma de gestión y orquestación de microservicios
(Impacto: rojo = Alto, amarillo = Medio, verde = Bajo)

Kubernetes es una solución muy atractiva porque es *open source* y el código está bajo la tutela de la CNCF, algo que contrasta, por ejemplo, con *Docker Swarm* que, a pesar de ser también *open source*, está demasiado controlado por *Docker Inc.*

Se han descartado *Kontena*, *Nomad* y *Rancher* por la poca aceptación en el sector según las fuentes consultadas, lo que implica no cumplir uno de los requisitos de más alto nivel (R2), con esto la comparativa se ha centrado en estudiar *Swarm*, *Kubernetes* y *DC/OS*. De esta comparativa destaca como principal diferencia la simplicidad de uso de *Swarm*, la escalabilidad de *Kubernetes* y *DC/OS* y la capacidad de estas dos plataformas para gestionar entornos complejos, y en casos de muchos nodos y los que la escalabilidad es crítica *DC/OS* sería la mejor solución.

Teniendo en cuenta lo anteriormente expuesto, para cumplir los requisitos anteriores, para proyectos pequeños, en los que prime la simplicidad, la mejor opción será elegir *Docker* con *Docker Swarm*. En el caso de proyectos mayores, o proyectos pequeños con expectativas de crecimiento, en los que la escalabilidad vaya a ser importante, se mantendrá ***Docker*** como tecnología de contedores, pero es optará por ***Kubernetes* o *DC/OS*** (*Docker* con *Apache Mesos* y *Marathon*) como tecnologías de gestión y orquestación de microservicios.

[2] Licencia MIT: Docker Swarm; Apache 2.0: Kontena, Nomad, Kubernetes, Rancher, DC/OS

III. CONCLUSIONES Y PROPUESTA DE TRABAJO FUTURO.

III.1 CONCLUSIONES

Las técnicas de secuenciación de nueva generación, NGS (Next Generation Sequencing), han tenido como principal consecuencia la denominada explosión de datos Genómicos gracias a la cual la Comunidad Científica dispone de una cantidad ingente de información genómica. La principal consecuencia es la necesidad de aplicar técnicas de *Big Data* y de computación de distribuida en la nube.

Por otra parte, la Bioinformática actual se caracteriza por la compartición de información y la colaboración entre científicos e instituciones dispersas por todo el mundo en proyectos de investigación que cada son más complejos. Se requiere, por tanto, que cada experimento sea "empaquetado" de manera que permita reproducirlo en las mismas condiciones. Ese empaquetado de los experimentos y *pipelines* en los llamados contenedores cumple con las exigencias de portabilidad y reproducibilidad citadas anteriormente.

Finalmente, existe una marcada tendencia en el desarrollo de software a diseñar aplicaciones siguiendo un paradigma de microservicios basado, fundamentalmente, en la división funcional especializada del dominio de la aplicación, tal y como se ha expuesto a lo largo del presente trabajo.

Con estas consideraciones, este TFM ha buscado identificar la mejor solución tecnológica que resuelva los problemas identificados en el *Big Data Genómico* haciendo uso de las tecnologías empleadas en la gestión de microservicios, permitiendo la provisión de servicios genómicos de alta disponibilidad, así como la integración continua en desarrollo, pruebas y puesta en producción de los mismos.

Tras un estudio exhaustivo del estado del arte del software en este dominio, se ha identificado una serie requisitos de selección de las herramientas componentes del framework objetivo y son los siguientes:

R1. Open source

R2. Tecnologías de uso más generalizado en la comunidad bioinformática

R3. Seguridad y privacidad

R4. Gestión de grandes volúmenes de datos

R5. Reutilización

R6. Trabajo colaborativo

R7. Reproducibilidad

R8. Eficiencia/rendimiento

R9. Escalabilidad

R10. Estandarización

R11. Usabilidad / facilidad de uso

R12. Inmediatez

Considerando estos requisitos queda en evidencia que será necesario extender el uso las tecnologías de computación en la Nube en el ámbito genómico para cumplir especialmente con los requisitos R4, R6 y R8, pero también R3, R9 y R12.

Por otro lado, las arquitecturas de microservicios, se ofrecen como solución más ventajosa para el *Big Data Genómico*, y que permitirán en primer lugar cubrir los requisitos R5, R7 y R9 y en segundo lugar R3, R8, R10, R11 y R12.

Una vez justificada la necesidad de utilizar arquitecturas de microservicios y computación en la nube para el *Big Data Genómico*, se ha realizado una selección de las tecnologías de microservicios más apropiadas.

Teniendo en cuenta lo anteriormente expuesto, para cumplir los requisitos identificados en este TFM, se ha llegado a la conclusión de que, en las condiciones actuales, los microservicios genómicos se empaquetarán en imágenes ***Docker*** y que las opciones más adecuadas para la gestión y orquestación de los contenedores en sistemas complejos son ***Kubernetes*** y ***DC/OS*** (*Apache Mesos* y *Marathon*).

III.2 PROPUESTAS DE TRABAJO FUTURO.

El grupo de Sistemas Información Genómicos del Centro PROS de la UPV, en el marco del proyecto de Excelencia DataMe (ref: TIN2016-80811-P, duración 2017-2020), está desarrollando un método de producción de software para el desarrollo de aplicaciones *Big Data* y va a llevar a cabo un caso de estudio en el dominio genómico. Los resultados de este proyecto son de utilidad en la primera fase de desarrollo del proyecto, y, las conclusiones de este TFM, con la selección de la mejor solución tecnológica, permitirá plantearse el uso de microservicios para el desarrollo del caso de estudio genómico.

Asimismo, este TFM ha puesto en evidencia las ventajas del uso de microservicios en la nube para resolver los retos del desarrollo de software en el ámbito genómico y ofrece evidencias para optar por el diseño de aplicaciones siguiendo un paradigma de microservicios en el desarrollo de nuevos servicios bioinformáticos. También servirá para valorar la migración a microservicios de la herramienta VarSearch, herramienta de búsqueda de variaciones del genoma humano desarrollada en el PROS, utilizando la plataforma de microservicios propuesta.

IV.BIBLIOGRAFÍA

[1] Z. Stephens, S. Lee, F. Faghri, R. Campbell, C. Zhai, M. Efron y et al, «Big Data: Astronomical or Genomical?,» *PLOS Biology*, vol. 13, nº 7, 7 Julio 2015.

[2] E. Hayden, «Genome researchers raise alarm over big data,» *Nature*, p. 312–314, July 2015.

[3] J. Rothberg y et al, «An integrated semiconductor device enabling non-optical genome sequencing,» *Nature*, vol. 475, nº 75356, pp. 348-352, 21 July 2011.

[4] J. Clarke, H. Wu, L. Jayasinghe y et al., «Continuous base identification for single-molecule nanopore DNA sequencing,» *Nature Nanotechnology*, vol. 4, nº 4, pp. 265-70, 2009.

[5] Q. Huang, S. Jing, J. Yi y W. Zhen, Innovative Testing and Measurement Solutions for Smart Grid, Singapore: John Wiley & Sons, 2015.

[6] W. Raghupathi y V. Raghupathi, «Big data analytics in healthcare: Promise and potential,» *Health Inf. Sci. Syst.*, vol. 2, nº 3, 2014.

[7] R. Chen y et al., «Personal omics profiling reveals dynamic molecular and medical phenotypes,» *Cell*, vol. 148, nº 6, pp. 1293-1307, 16 Marzo 2012.

[8] H. Rehm, J. Berg, L. Brooks y et al., «ClinGen - The clinical genome resource,» *The New England Journal of Medicine*, vol. 372, 2015.

[9] Y. Karen, G. Dongliang y M. Max, «Big Data Analytics for Genomic Medicine,» *International Journal of Molecular Sciences*, vol. 18, nº 2, 2017.

[10] J. M. Abuín, J. C. Pichel, T. F. Pena y J. Amigo, «BigBWA: approaching the Burrows–Wheeler aligner to Big Data technologies,» *Bioinformatics*, vol. 31, nº 24, p. 4003–4005, 2015.

[11] B. Kayyali, D. Knott y S. Van Kuiken, «The big-data revolution in US health care: Accelerating value and innovation,» 2013.

[12] J. Sun y C. Reddy, «Big Data Analytics for Healthcare,» de *ACM SIAM International Conference on Data Mining*, Austin, TX, 2013.

[13] Research and Markets, «Big Data in the Healthcare & Pharmaceutical Industry: 2017 – 2030 – Opportunities, Challenges, Strategies & Forecasts,» Research and Markets, 2017.

[14] Boehringer Ingelheim, «Pradaxa® (dabigatran etexilate) 150mg bid continues to be the only oral anticoagulant which showed superior ischaemic stroke reduction vs. warfarin in its pivotal study RE-LY® – Results of the ENGAGE AF-TIMI 48 trial published for edoxaban vs. warfarin do,» 2013.

[15] Bayer, «Los datos de vida real en diferentes países confirman el balance beneficio-riesgo positivo del anticoagulante oral rivaroxaban en la práctica clínica habitual,» 29 8 2016. [En línea]. Available: https://www.bayer.es/medios/noticias/pharmaceuticals/2016_08_29_los-datos-de-vida-real-en-diferentes-paises-confirman-el-balance-beneficio-riesgo-positivo-del-anticoagulante-oral-rivaroxaban.php. [Último acceso: 26 marzo 2018].

[16] Roche, «Roche to acquire Flatiron Health to accelerate industry-wide development and delivery of breakthrough medicines for patients with cancer,» Basel, 2018.

[17] SEIS, «Informes SEIS (Sociedad Española de Informática de la Salud),» 2016. [En línea]. Available: http://www.seis.es/Informes.html. [Último acceso: 15 mayo 2018].

[18] Inst. Inf. Sanitaria, «El Sistema de Historia Clínica Digital del SNS,» Ministerio de Sanidad y Política Social, Madrid, 2009.

[19] E. Jamoom, N. Yang y E. Hing, «Adoption of certified electronic health record systems and electronic information sharing in physician offices: United States, 2013 and 2014,» Hyattsville, 2016.

[20] CRUE-TIC, «Promoción de software útil y económico en Universidades y Centros de Enseñanza Superior e Investigación,» [En línea]. Available:

https://cruetic.ugr.es/pages/comunicaciones/plumecrue_tic/%21. [Último acceso: 27 junio 2018].

[21] Wikipedia, «Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities,» [En línea]. Available: https://en.wikipedia.org/wiki/Berlin_Declaration_on_Open_Access_to_Knowledge_in_the_Sciences_and_Humanities. [Último acceso: 25 junio 2018].

[22] J. Quackenbush, «Open-source software accelerates bioinformatics,» *Genome Biology*, vol. 4, nº 336, 2003.

[23] «Global Alliance for Genomics and Health,» [En línea]. Available: https://www.ga4gh.org/. [Último acceso: 27 junio 2018].

[24] R. Swaminathana, Y. Huanga, S. Moosavinasaba, R. Buckleya, C. W. Bartlettb y S. M. Lin, «A Review on Genomics APIs,» *Computational and Structural Biotechnology Journal*, vol. 14, pp. 8-15, 2016.

[25] Google Cloud, «Google Genomics,» [En línea]. Available: https://cloud.google.com/genomics/. [Último acceso: 27 junio 2018].

[26] Harvard University, «Smart Genomics,» [En línea]. Available: https://projects.iq.harvard.edu/smartgenomics. [Último acceso: 20 mayo 2018].

[27] 23andMe, «What is the 23andMe API?,» [En línea]. Available: https://api.23andme.com/overview/. [Último acceso: 10 abril 2018].

[28] D. Blankenberg, G. Von Kuster y N. Coraor, «*Galaxy*: A Web-Based Genome Analysis Tool for Experimentalists,» *Current Protocols in Molecular Biology*, vol. 89, nº 1, pp. 19.10.1-19.10.21, 2010.

[29] W. Digan y et al, «An architecture for genomics analysis in a clinical setting using *Galaxy* and *Docker*,» *GigaScience*, vol. 6, nº 11, pp. 1-9, 2017.

[30] F. da Veiga Leprevost y et al, «BioContainers: an open-source and community-driven framework for software standardization,» *Bioinformatics*, vol. 33, nº 16, pp. 2580-2582, 2017.

[31] *Docker* Hub, «Perl,» [En línea]. Available: https://hub.*Docker*.com/_/perl/. [Último acceso: 27 junio 2018].

[32] *Docker* hub, «Bioperl,» [En línea]. Available: https://hub.*Docker*.com/r/bioperl/bioperl/. [Último acceso: 27 junio 2018].

[33] *Docker* hub, «Python,» [En línea]. Available: https://hub.*Docker*.com/_/python/. [Último acceso: 27 junio 2018].

[34] *Docker* hub, «Biopython,» [En línea]. Available: https://hub.*Docker*.com/r/biopython/biopython/. [Último acceso: 27 junio 2018].

[35] *Docker* hub, «R-base,» [En línea]. Available: https://hub.*Docker*.com/_/r-base/. [Último acceso: 27 junio 2018].

[36] *Docker* hub, «Bioconductor,» [En línea]. Available: https://hub.*Docker*.com/r/bioconductor/release_base/. [Último acceso: 6 junio 2018].

[37] H. M. Krumholz y J. Waldstreicher, «The Yale Open Data Access (YODA) Project — A Mechanism for Data Sharing,» *The New England Journal of Medicine*, vol. 375, pp. 403-405, 4 Agosto 2016.

[38] F. Collins y A. Barker, «Mapping the cancer genome. Pinpointing the genes involved in cancer will help chart a new course across the complex landscape of human malignancies,» *Scientific American*, vol. 296, nº 3, 2007.

[39] D. Roldán Martínez, Bioinformática: el ADN a un solo clic, Madrid: RA-MA, 2015.

[40] A. Nekrutenko y J. Taylor, «Next-generation sequencing data interpretation: enhancing reproducibility and accessibility,» *Nature Reviews Genetics*, vol. 13, nº 667, pp. 667-672, 17 Agosto 2012.

[41] B. M. Kuenzi, A. L. Borne, J. Li, E. B. Haura, S. A. Eschrich, J. M. Koomen, U. Rix y P. A. Stewart, «APOSTL: An Interactive *Galaxy Pipeline* for Reproducible Analysis of Affinity Proteomics Data,» *Journal of Proteome Research,* vol. 15, nº 12, pp. 4747-4754, 2016.

[42] W. Schulz, T. Durant, A. J. Siddon y R. Torres, «Use of application containers and *workflows* for genomic data analysis,» *Journal of Pathology Informatics,* vol. 7, nº 53, 2016.

[43] D. Garijo, S. Kinnings, L. Xie, L. Xie, Y. Zhang, Y. Bourn y et al, «Quantifying Reproducibility in Computational Biology: The Case of the Tuberculosis Drugome,» *PLOS ONE,* vol. 8, nº 11, 2013.

[44] W. Gerlach y e. at., «Skyport – Container-Based Execution EnvironmentManagement for Multi-Cloud Scientific *Workflows*,» de *Data-Intensive Computing in the Clouds (DataCloud), 2014 5th International Workshop on*, New Orleans, LA, USA, 2014.

[45] C. Boettiger, «An introduction to *Docker* for reproducible research,» *ACM SIGOPS Operating Systems Review - Special Issue on Repeatability and Sharing of Experimental Artifacts,* vol. 49, nº 1, pp. 17-79, 2015.

[46] W. Felter, A. Ferreira, R. Rajamony y J. Rubio, «An updated performance comparison of virtual machines and linux contain.,» IBM Research, 2014.

[47] W. Digan, H. Countouris, M. Barritault, D. Baudoin, P. Laurent-Puig, H. Blons, A. Burgun y B. Rance, «An architecture for genomics analysis in a clinical setting using *Galaxy* and *Docker*,» *Gigascience,* vol. 6, nº 11, pp. 1-9, 2017.

[48] K. Wolstencroft y et al., «The Taverna *workflow* suite: designing and executing *workflows* of Web Services on the desktop, web or in the cloud,» *Nucleic Acids Research,* vol. 41, nº W1, pp. 557-561, 2013.

[49] E. Ayday, E. De Cristofaro, J. P. Hubaux y G. Tsudik, «Whole Genome Sequencing: Revolutionary Medicine or Privacy Nightmare?,» *Computer,* vol. 48, nº 2, pp. 58-66, Febrero 2015.

[50] J. W. Bos, K. Lauter y M. Naehrig, «Private predictive analysis on encrypted medical data,» *Journal of Biomedical Informatics,* vol. 50, pp. 234-243, 2014.

[51] S. D. Constable, Y. Tang, S. Wang, X. Jiang y S. Chapin, «Privacy-preserving GWAS analysis on federated genomic datasets,» *BMC Med. Inform. Decis. Mak.,* vol. 15, nº 5, 2015.

[52] F. Chen y e. al., «PRINCESS: Privacy-protecting Rare disease International Network Collaboration via Encryption through Software guard extensionS,» *Bioinformatics,* vol. 33, nº 6, pp. 871-878, 2017.

[53] L. Kamm, D. Bogdanov, S. Laur y J. A. Vilo, «A new way to protect privacy in large-scale genome-wide association studies,» *Bioinformatics,* vol. 29, nº 7, p. 886–893, 2013.

[54] X. Jiang, Y. Zhao y et al., «A community assessment of privacy preserving techniques for human genomes,» *BMC. Med. Inform. Decis. Mak.,* vol. 14, 2014.

[55] M. Naveed y et al, «Privacy and security in the genomic era,» *ACM Comput. Surv.,* vol. 48, nº 1, 2015.

[56] H. Tang y et al, «Protecting genomic data analytics in the cloud: state of the art and opportunities,» *BMC Med. Genomics,* vol. 9, nº 63, 2016.

[57] A. Schlosberg, «Data security in genomics: A review of Australian privacy requirements and their relation to cryptography in data storage,» *J Pathol Inform,* vol. 7, nº 6, 2016.

[58] S. Wang y et al., «A community effort to protect genomic data sharing, collaboration and outsourcing,» *npj Genomic Medicine,* vol. 2, nº 1, 2017.

[59] S. Wang y et al., «HEALER: homomorphic computation of ExAct Logistic rEgRession for secure rare disease variants analysis in GWAS,» *Bioinformatics,* vol. 32, nº 2, pp. 211-218, 2016.

[60] Y. Zhang y et al., «FORESEE: Fully Outsourced secuRe gEnome Study basEd on homomorphic Encryption,» *BMC Medical Informatics and Decision Making,* vol. 15, nº 5, 2015.

[61] M. Hatakeyama, L. Opitz y G. Russo, «SUSHI: an exquisite recipe for fully documented, reproducible and reusable NGS data analysis,» *BMC Bioinformatics,* vol. 17, nº 228, 2016.

[62] P. Romano, R. Giugno y A. Pulvirenti, «Tools and collaborative environments for bioinformatics research,» *Briefings in Bioinformatics,* vol. 12, nº 6, pp. 549-561, 2011.

[63] B. Howe, «Virtual Appliances, *Cloud computing,* and Reproducible Research,» *Computing in Science and Engineering Magazine,* vol. 14, nº 4, pp. 36-41, 2012.

[64] L. D. Stein, B. M. Knoppers, P. Campbell, G. Getz y J. Korbel, «Data analysis: Create a cloud commons,» *Nature,* vol. 523, pp. 149-151, 9 Julio 2015.

[65] Di Tommasso y et al., «The impact of *Docker* containers on the performance of genomic *pipelines*. 3:e1273,» *PeerJ,* vol. 3, 2015.

[66] F. Versaci, L. Pireddu y G. Zanetti, «Scalable genomics: From raw data to aligned reads on *Apache* YARN,» de *2016 IEEE International Conference on Big Data (Big Data),* Washington, 2016.

[67] S. Moorthie, A. Hall y C. F. Wright, «Informatics and clinical genome sequencing: opening the black box,» *Genetics in Medicine,* vol. 15, pp. 165-171, 13 Septiembre 2013.

[68] D. Bolchini y et al, «Better bioinformatics through usability analysis,» *Bioinformatics,* vol. 33, nº 16, pp. 2580-2582, 2009.

[69] A. Tempest y et at., «The need to redefine genomic data sharing: A focus on data accessibility,» *Applied & Translational Genomics,* vol. 3, nº 4, pp. 100-104, 2014.

[70] W. Lathe, J. Williams, M. Mangan y D. Karolchik, «Genomic Data Resources: Challenges and Promises,» *Nature Education* , vol. 1, nº 3, 2008.

[71] M. Fowler, «Microservices a definition of this new architectural term,» 25 Marzo 2014. [En línea]. Available: http://martinfowler.com/articles/microservices.html. [Último acceso: 23 abril 2018].

[72] M. Fowler, «Tolerant Reader,» 9 Mayo 2011. [En línea]. Available: http://www.martinfowler.com/bliki/TolerantReader.html. [Último acceso: 23 abril 2018].

[73] V. Reynolds, Getting started with Microservices, DZone REFCARD #215, 2016.

[74] V. Reynolds, «Getting started with Microservices,» *DZone REFCARDZ,* nº 215, p. 6, 2016.

[75] I. Robinson, «Consumer-Driven Contracts: A Service Evolution Pattern,» Martin Fowler, 16 Junio 2006. [En línea]. Available: http://www.martinfowler.com/articles/consumerDrivenContracts.html. [Último acceso: 4 febrero 2018].

[76] E. Wolff, Microservices Flexible Software Architecture, LeanPub, 2016.

[77] Stackshare, «*Kubernetes* vs. *Rancher* vs. *Docker Swarm*,» [En línea]. Available: https://stackshare.io/stackups/*Docker-Swarm*-vs-*Kubernetes*-vs-*Rancher*. [Último acceso: 2 mayo 2018].

[78] C. Tozzi, «Container Orchestrators You Might Have Missed,» *Container Journal,* 26 Mayo 2017.

[79] Platform9, «*Kubernetes* vs *Docker Swarm*,» 22 Junio 2017. [En línea]. Available: https://platform9.com/blog/*Kubernetes-Docker-Swarm*-compared/. [Último acceso: 27 junio 2018].

[80] C. Williams, J. Sica, R. Killen y U. Balis, «The growing need for microservices in bioinformatics,» *Journal of Pathology Informatics,* vol. 7, nº 45, 2016.

[81] T. D. A. J. S. R. T. Wade L Schulz, «Use of application containers and *workflows* for genomic data analysis,» *Journal of Pathology Informatics,* 2016.

[82] GenomeHub - Genomics made easy, «Next Generation Sequencing Data Analysis,» [En línea]. Available: https://genomehubcam.wordpress.com/ngs-data-2/. [Último acceso: 27 junio 2018].

[83] A. Sharma y A. L. S. Rai, «*Workflow* management systems for gene sequence analysis and evolutionary studies – A Review,» *Bioinformation,* vol. 9, nº 13, pp. 663-672, 2013.

[84] J. Köster y S. Rahmann, «Snakemake— a scalable bioinformatics *workflow* engine,» *Bioinformatics,* vol. 28, nº 19, pp. 2520-2522, 2012.

[85] P. Di Tommaso y et al., «*NextFlow* enables reproducible computational *workflows*,» *Nature Biotechnology*, vol. 35, nº 4, pp. 316-319, 2017.

[86] B. Grüning, «*Docker-Galaxy* 2016,» [En línea]. Available: https://github.com/bgruening/*Docker-Galaxy*-stable. [Último acceso: 23 junio 2018].

[87] F. Ramírez y et al., «deepTools2: a next generation web server for deep-sequencing data analysis,» *Nucleic Acids Research,* vol. 8, nº 44, pp. W160-W165, 2016.

[88] G. Cheng, Q. Lu, L. Ma, G. Zhang, L. Xu y Z. Zhou, «BGDM*Docker*: a *Docker workflow* for data mining and visualization of bacterial pan-genomes and biosynthetic gene clusters,» *PeerJ,* vol. 5, p. e3948, 2017.

[89] U. Devisetty, K. Kennedy, P. Sarando y et al., «Bringing your tools to cyverse discovery environment using *Docker*,» *F1000Research,* vol. 5, nº 1442, 2016.

[90] S. O. M. H. e. a. Moreews F, «BioShaDock: a community driven bioinformatics shared *Docker*-based tools registry,» *F1000Research,* vol. 4, nº 1443, 2015.

[91] International Cancer Genome Consortium (ICGC), «ICGC-TCGA-PanCancer,» [En línea]. Available: https://github.com/ICGC-TCGA-PanCancer/cli/wiki. [Último acceso: 24 junio 2018].

[92] Illumina, «BaseSpace Sequence Hub,» [En línea]. Available: https://basespace.illumina.com/home/index. [Último acceso: 23 junio 2018].

[93] Center for Genomic Regulation (CRG), «*NextFlow*,» [En línea]. Available: https://www.*NextFlow*.io/. [Último acceso: 26 junio 2018].

[94] J. Baier, Getting Started with *Kubernetes*, Packt Publishing Ltd, 2015.

[95] D. Kakadia, *Apache Mesos* Essentials, Packt Publishing, 2015.

[96] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. D Joseph, R. Katz, S. Shenker y I. Stoica, «*Mesos*: A Platform for Fine-Grained Resources Sharing in the Data Center,» University of California, Berkeley.

[97] «Azure Microsoft,» [En línea]. Available: https://azure.microsoft.com/mediahandler/files/resourcefiles/249aaf61-9b90-40f9-a217-41b6d23643f7/Deploying-Microservices-and-Containers-with-Azure-Container-Service-and-DC-OS.pdf. [Último acceso: 25 junio 2018].

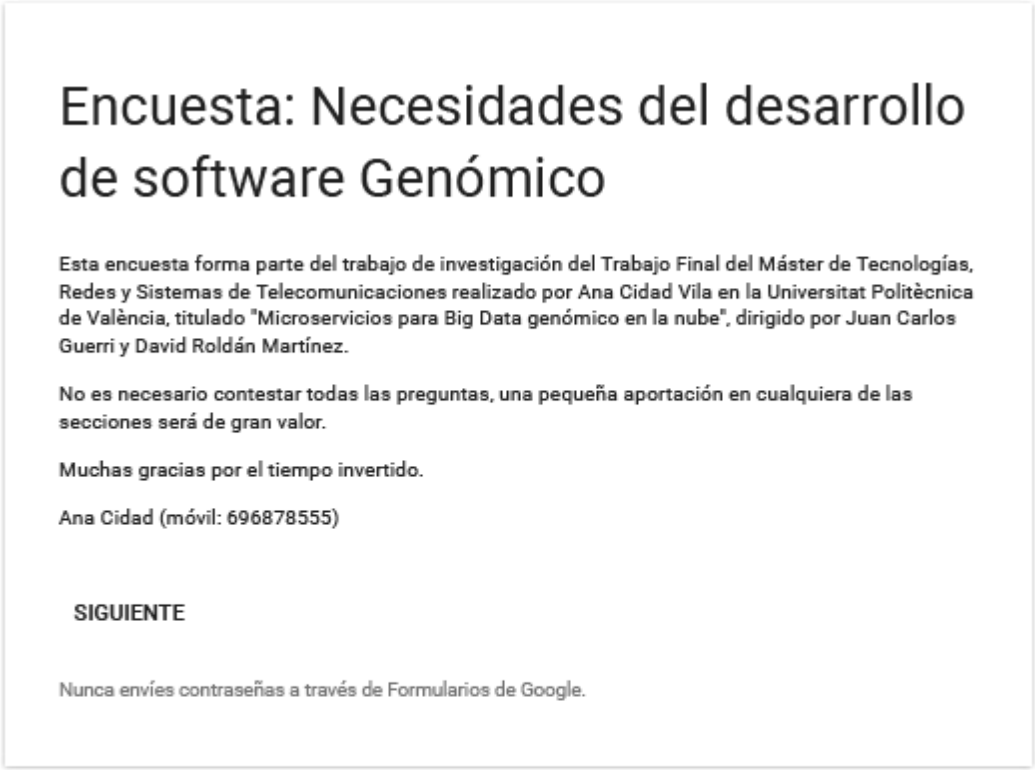
[98] «*Docker Swarm*,» [En línea]. Available: http://searchitoperations.techtarget.com/definition/*Docker-Swarm*. [Último acceso: 27 junio 2017].

[99] *Docker*, «*Docker Swarm* overview,» [En línea]. Available: https://docs.*Docker*.com/*Swarm*/overview/. [Último acceso: 27 junio 2018].

ANEXO 1 – ENCUESTA: NECESIDADES DEL DESARROLLO DE SOFTWARE GENÓMICO

https://goo.gl/forms/jywKXjM6l7qDNpiF2

Encuesta publicada en *Google Forms* y dirigida a expertos del sector genómico:



Encuesta: Necesidades del desarrollo de software Genómico

Esta encuesta forma parte del trabajo de investigación del Trabajo Final del Máster de Tecnologías, Redes y Sistemas de Telecomunicaciones realizado por Ana Cidad Vila en la Universitat Politècnica de València, titulado "Microservicios para Big Data genómico en la nube", dirigido por Juan Carlos Guerri y David Roldán Martínez.

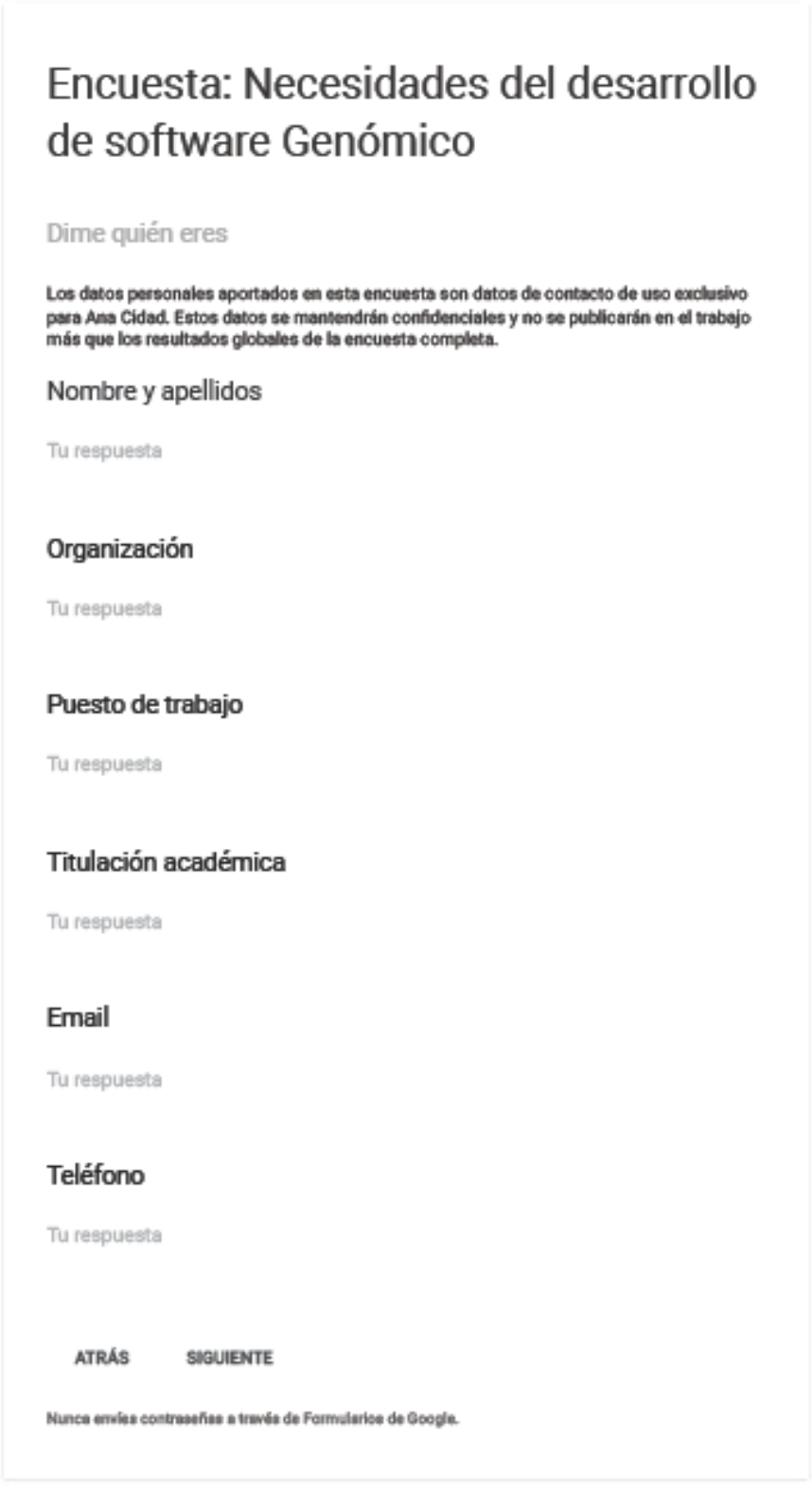
No es necesario contestar todas las preguntas, una pequeña aportación en cualquiera de las secciones será de gran valor.

Muchas gracias por el tiempo invertido.

Ana Cidad (móvil: 696878555)

SIGUIENTE

Nunca envíes contraseñas a través de Formularios de Google.



Encuesta: Necesidades del desarrollo de software Genómico

Dime quién eres

Los datos personales aportados en esta encuesta son datos de contacto de uso exclusivo para Ana Cidad. Estos datos se mantendrán confidenciales y no se publicarán en el trabajo más que los resultados globales de la encuesta completa.

Nombre y apellidos

Tu respuesta

Organización

Tu respuesta

Puesto de trabajo

Tu respuesta

Titulación académica

Tu respuesta

Email

Tu respuesta

Teléfono

Tu respuesta

ATRÁS SIGUIENTE

Nunca envíes contraseñas a través de Formularios de Google.

Este contenido no ha sido creado ni aprobado por Google. Informar sobre abusos - Condiciones del servicio - Otros términos

Google Formularios

Encuesta: Necesidades del desarrollo de software Genómico

Valoración del impacto de Requisitos del desarrollo de software en entornos genómicos de investigación

Valorar del 0 al 10 el impacto de cada requisito planteado.
Estos requisitos se centran en el desarrollo de software en el ámbito genómico y bioinformático, con el fin de poder elegir las mejores tecnologías que permitan dar soporte a nuevos desarrollos en entornos principalmente de investigación. No estamos hablando de requisitos para el software de usuario final destinado a médicos o genetistas, sino de las necesidades de aquellos que desarrollan software y utilizan tecnologías en el entorno bioinformático. Esta información servirá para tomar decisiones tecnológicas como valorar la necesidad de trabajar en la nube, o utilizar contenedores, o seleccionar la mejor plataforma de gestión de microservicios.

R1.Open Source

En un entorno universitario y en general de investigación, es una tendencia innegable el uso de soluciones open source. Además de los motivos económicos, la CRUE-TIC y la Declaración de Berlín apoyan este tipo de proyectos porque ofrecen la posibilidad de acceder al código fuente, mejorarlo y participar activamente en una comunidad abierta, algo que encaja con la labor social de fomento del Conocimiento propia de una institución educativa o de investigación en general.

Valora el impacto de R1

	0	1	2	3	4	5	6	7	8	9	10	
Menor impacto	○	○	○	○	○	○	○	○	○	○	○	Mayor impacto

R2. Tecnologías de uso más generalizado en la comunidad bioinformática

Según crece la complejidad de las herramientas bioinformáticas, resulta difícil para los pequeños laboratorios y grupos de investigación mantener en el tiempo técnicos capaces de modificar y desarrollar posteriormente todas las fases de un pipeline bioinformático. Por ello cada vez se hace más necesaria la existencia de plataformas que permitan la experimentación y el uso de una gran variedad de herramientas y algoritmos existentes, con servicios divididos en módulos gestionables e intercambiables. Por este motivo se ha valorado de forma especial, la tendencia general de la comunidad científica en el dominio de la bioinformática a la hora de escoger cualquier tecnología, para asegurar el mantenimiento y evolución de los servicios desarrollados, así como la facilidad a la hora de integrar y hacer uso de servicios existentes

Valora el impacto de R2

	0	1	2	3	4	5	6	7	8	9	10	
Menor impacto	○	○	○	○	○	○	○	○	○	○	○	Mayor impacto

R3. Seguridad y privacidad

El incremento de la disponibilidad de los datos genómicos tiene grandes implicaciones para la privacidad personal. El genoma tiene características esenciales que, entre otras, incluyen la asociación con riesgo de enfermedades, identificación de capacidades y revelación de parentescos, que hacen de los datos genómicos información altamente sensible. Es necesario mantener tanto la autenticidad e integridad de los datos (los datos son correctos y no se pueden alterar), como la privacidad (solo el perfil autorizado puede acceder a estos datos)

Valora el impacto de R3

0 1 2 3 4 5 6 7 8 9 10

Menor impacto ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Mayor impacto

R4. Gestión de grandes volúmenes de datos

En el ámbito genómico, uno de los principales problemas que se encuentran son los grandes volúmenes de información que hay que compartir/transferir y la gran cantidad de recursos de computación necesarios para manejarlos.

Valora el impacto de R4

0 1 2 3 4 5 6 7 8 9 10

Menor impacto ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Mayor impacto

R5. Reutilización

En el ámbito de las ciencias de la vida, además del reto del Big Data, la principal dificultad se encuentra en que la mayoría de investigadores biomédicos no disponen da capacidad para realizar por sí mismos el análisis de los grandes conjuntos de datos actuales utilizando las herramientas e infraestructura computacional adecuada, de forma que pueda ser completamente comprendido y reutilizado por otros. Es un requisito en este ámbito, especialmente por los retos introducidos por la secuenciación NGS y para no "reinventar la rueda", el disponer de plataformas y metodologías que permitan reutilizar herramientas desarrolladas por la comunidad bioinformática

Valora el impacto de R5

0 1 2 3 4 5 6 7 8 9 10

Menor impacto ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Mayor impacto

R6. Trabajo colaborativo

En la investigación avanzada en general, y especialmente en comunidades de software libre e investigación en abierto (open research), se requiere una gran interacción y colaboración. Con el fin de alcanzar objetivos de investigación más ambiciosos y de mayor impacto internacional, además de compartir información, es necesario poder trabajar de forma colaborativa en nuevos algoritmos para su mejora y compartición

Valora el impacto de R6

0 1 2 3 4 5 6 7 8 9 10

Menor impacto ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Mayor impacto

R7. Reproducibilidad

El campo actual del análisis de la secuenciación NGS está en constante fluctuación, y no existe un amplio consenso sobre lo que son las "buenas prácticas". En esta situación es especialmente importante poder reutilizar y adoptar diferentes aproximaciones analíticas publicadas en la literatura, pero en las actuales publicaciones de análisis computacionales no existen suficientes detalles para llevarlos a cabo. Para reproducir el experimento computacional más simple es necesario tener como mínimo acceso a los datos primarios y tener acceso al software y su versión, los parámetros de configuración y el nombre del genoma de referencia, y con frecuencia no toda esa información aparece en las publicaciones científicas de referencia. Es una necesidad disponer de herramientas que permitan reproducir los experimentos de una forma sencilla, fiable y transparente para el investigador

Valora el impacto de R7

0 1 2 3 4 5 6 7 8 9 10

Menor impacto ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Mayor impacto

R8. Eficiencia/rendimiento

Considerando las necesidades de computación necesaria para procesar volúmenes de información tan grande y en la actualidad, los análisis genómicos actuales conllevan periodos de tiempo muy grandes. Es de gran relevancia elegir la plataforma que realice una mejor gestión de los recursos software y hardware. La eficiencia o mejora en el rendimiento se traducirá en un ahorro de costes y tiempos

Valora el impacto de R8

0 1 2 3 4 5 6 7 8 9 10

Menor impacto ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Mayor impacto

R9. Escalabilidad

Uno de los retos que han surgido de la revolución del Big Data genómico, es el de desarrollar herramientas de computación escalables que puedan dar soporte al crecimiento exponencial del flujo de datos masivos generados. Además, con el objetivo de poder hacer un uso versátil, las herramientas deben ser escalables, y ofrecer la capacidad adecuada de dar soporte a configuraciones tanto sencillas y complejas

Valora el impacto de R9

0 1 2 3 4 5 6 7 8 9 10

Menor impacto ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Mayor impacto

R10. Estandarización

La implementación clínica de las tecnologías NGS requiere la estandarización e integración de las bases de datos y "pipelines" de análisis, así como de herramientas informáticas para la toma de decisiones. El uso de herramientas que utilicen estándares ampliamente aceptados por la comunidad, garantiza su compatibilidad con desarrollos existentes y futuros, lo que ayudará en general a impulsar el sector bioinformático

Valora el impacto de R10

0 1 2 3 4 5 6 7 8 9 10

Menor impacto ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Mayor impacto

R11. Usabilidad / facilidad de uso

Mejorar la usabilidad de los recursos bioinformáticos permite a los investigadores encontrar, interactuar, compartir, comparar y manipular información relevante de una forma más eficiente y efectiva. Esto redunda en una mejor percepción de los procesos biológicos con el potencial último de conseguir nuevos resultados de investigación. Las barreras de usabilidad pueden generar importantes obstáculos por no satisfacer la experiencia de usuario y forzar a los investigadores a emplear tiempo y esfuerzos innecesarios para completar sus tareas. En un entorno en que el número de bases de datos crece, así como la diversidad de usuarios, es necesario elegir las soluciones tecnológicas que cumplan los más altos estándares de usabilidad.

Valora el impacto de R11

0 1 2 3 4 5 6 7 8 9 10

Menor impacto ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Mayor impacto

R12. Inmediatez

En situación de urgencia es necesario que la información clínica de un paciente esté disponible en tiempo real. Aunque en los análisis de datos genómicos suele ser un proceso lento y en la actualidad no se espera esta inmediatez, sí que lo es para los sistemas que también traten con información clínica

Valora el impacto de R12

0 1 2 3 4 5 6 7 8 9 10

Menor impacto ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ Mayor impacto

Describe otro requisito que consideres relevante y no aparezca en esta lista

Tu respuesta

ATRÁS SIGUIENTE

Nunca envíes contraseñas a través de Formularios de Google.

Encuesta: Necesidades del desarrollo de software Genómico

Información sobre tecnologías empleadas

Aquí valoraré que me aportes información sobre las tecnologías que actualmente utilizáis en vuestra organización para el desarrollo de software genómico. No es necesario contestar todas las preguntas.

Describe brevemente qué tipo de desarrollo de software genómico se lleva a cabo en tu organización

Tu respuesta

¿Reutilizáis software o utilizáis servicios desarrollados por terceros? ¿cuáles?

Tu respuesta

¿Almacenáis los datos genómicos en local o en la nube? ¿por qué habéis adoptado esa decisión?

Tu respuesta

¿Utilizáis algún sistema de información o base de datos para almacenar la información genómica? ¿puedes describirlo brevemente?

Tu respuesta

¿Utilizas tecnologías de Máquinas virtuales o contenedores? ¿cuáles?

Tu respuesta

¿Quieres contarme algo más sobre vuestra forma de trabajar?

Tu respuesta

ATRÁS ENVIAR