ANÁLISIS DE ESTRATEGIAS DE ACTUALIZACIÓN DE REPOSITORIOS DIGITALES A DSPACE 7

Pablo César de Albuquerque[†]
PREBI-SEDICI, CESGI, Universidad Nacional de La Plata, Comisión de Investigaciones Científicas.
Santiago Tettamanti
PREBI-SEDICI, CESGI, Universidad Nacional de La Plata, Comisión de Investigaciones Científicas.
Gonzalo Luján Villarreal
PREBI-SEDICI, CESGI, Universidad Nacional de La Plata, Comisión de Investigaciones Científicas.
Ariel Jorge Lira
PREBI-SEDICI, CESGI, Universidad Nacional de La Plata, Comisión de Investigaciones Científicas.
Marisa Raquel De Giusti
PREBI-SEDICI, CESGI, Universidad Nacional de La Plata, Comisión de Investigaciones Científicas.

Resumen: En este trabajo se analizan distintas estrategias que se pueden aplicar a la hora de actualizar un repositorio digital desde DSpace 6 o anterior hacia DSpace 7. Se realiza un análisis de dicho software, con énfasis en las diferencias arquitecturales y funcionales de las distintas versiones. Luego se describe la experiencia obtenida a partir de la actualización a DSpace 7 del repositorio CIC DIGITAL, para lo cual se detalla el camino seguido a lo largo del proceso y se remarcan los conflictos que surgieron durante este proceso, vinculados tanto a la actualización del código fuente y de la plataforma de software, así como también a la migración de los datos hacia el nuevo esquema propuesto por DSpace 7. Finalmente, se reflexiona sobre las distintas estrategias de adopción de este software para los distintos escenarios en los que puede encontrarse un repositorio digital que pretenda encarar una migración de características similares.
Palabras clave: DSpace 7; migración de software; repositorios digitales.

Title: EXPERIENCES IN MIGRATING AN INSTITUTIONAL REPOSITORY TO DSPACE 7.
Abstract: This paper analyzes different strategies that can be applied when upgrading a digital repository from DSpace 6 or earlier to DSpace 7. An analysis of this software is made, with emphasis on the architectural and functional changes of the different versions. Then, it is described the experience obtained from the upgrade to DSpace 7 of the CIC DIGITAL repository, detailing the path followed through the process and highlighting the conflicts that arose during this process, related to the upgrade of the source code and the software platform as well as to the migration of data to the new schema proposed by DSpace 7. Finally, it is reflected on the different strategies of adoption of this software for the different scenarios in which a digital repository that intends to face a migration of similar characteristics may find itself.
Keywords: Software migration; digital repositories.

1 INTRODUCCIÓN

DSpace es el software para repositorios más utilizado en la actualidad: es utilizado por el 39% de los repositorios (OpenDOAR Statistics - v2.sherpa, 2022). Es un software de código abierto que permite capturar, almacenar, indizar, preservar y distribuir material digital como texto, datos, audio y video. Desde su creación, en el año 2002, DSpace ha evolucionado para incorporar nuevas funcionalidades y herramientas, y hasta la versión 6 ha mantenido una arquitectura interna sin grandes cambios.

En 2021 se liberó la versión 7 de DSpace que difiere de sus antecesoras en varios aspectos: uno de los más importantes es el cambio en su arquitectura y la tecnología utilizada en la interfaz del usuario.

A partir de la versión 7, el software se divide en dos aplicaciones independientes: el *backend* y el *frontend*. El *backend* se refiere a la parte del software encargada de la lógica de negocio, el procesamiento de datos y la gestión de

[†] pablo@sedici.unlp.edu.ar

Recibido: 19-04-2023; 2ª versión: 16-05-2023; 3ª versión 02-08-2023; aceptado: 10-08-2023.

ALBURQUERQUE, P.C.; TETTAMANTI, S.; VILLARREAL, G.L.; LIRA, A.J. y DE GIUSTI, M.R. Análisis de estrategias de actualización de repositiorios digitales a DSpace 7. *Anales de Documentación*, 2023, vol. 26. Disponible en: http://dx.doi.org/10.6018/analesdoc.567401.

la base de datos. Por otro lado, el *frontend* corresponde a la interfaz de usuario, es decir, la parte visual con la que interactúan los usuarios finales.

Estas dos aplicaciones se comunican entre sí a través de una API REST, que es una interfaz de programación de aplicaciones basada en el protocolo HTTP. Esta API REST expone toda la funcionalidad del repositorio, permitiendo que el *frontend* solicite y envíe datos al *backend* de manera eficiente y estructurada. En resumen, el *backend* y el *frontend* trabajan juntos para proporcionar una experiencia completa y funcional a los usuarios del software.

Este cambio arquitectural y la nueva API REST, acercan a DSpace a la propuesta de Next Generation Repositories (NGR) de COAR (COAR Next Generation Repositories: Vision and Objectives, 2020), la cual hace foco en la interoperabilidad entre repositorios, el acceso abierto y la incorporación de servicios de valor agregado para los investigadores.

Si bien la nueva versión incorpora ventajas y facilita el desarrollo de nuevos servicios sobre la API REST, cabe considerar que la adopción de esta nueva versión puede ser dificultosa para las instituciones, dependiendo de la versión que utilicen, del grado de personalización sobre el software y de si toda la funcionalidad necesaria está implementada (ya que algunas funciones de versiones previas aún no se implementaron). Es por eso que no existe una receta o un paso a paso definitivo que permita migrar a la versión 7, manteniendo todas las personalizaciones realizadas.

El repositorio institucional de la Comisión de Investigaciones Científicas de la provincia de Buenos Aires, denominado CIC-DIGITAL, fue creado a finales de 2014. Además de cumplir el rol de archivo digital institucional, el repositorio asiste a los investigadores, personal de apoyo y becarios de la CIC para que puedan disponer en Internet de su producción, maximizando su difusión e impacto, a la vez que protege a largo plazo sus obras. CIC-DIGITAL funciona bajo la dirección <https://digital.cic.gba.gob.ar> y a la fecha cuenta con casi 10.000 publicaciones en acceso abierto. Desde 2015 está adherido al Sistema Nacional de Repositorios Digitales (SNRD) y es cosechado en el portal nacional de repositorios digitales.

En este trabajo se busca compartir la experiencia a partir de la migración a DSpace 7 y exponer las dificultades encontradas, para que otras instituciones, interesadas en utilizar esta versión de DSpace, cuenten con más herramientas a la hora de definir una estrategia de adopción de este software, ya sea a partir de una migración desde una versión anterior o de la implementación desde cero.

2 METODOLOGÍA

Este estudio fue realizado en una primera instancia siguiendo un enfoque cualitativo, basado en el relevamiento del estado del desarrollo de DSpace 7. Se utilizaron múltiples fuentes de información a lo largo este proceso, entre ellas la documentación oficial de DSpace, los repositorios Github vinculados a los proyectos que conforman esta versión de DSpace, y la participación en un *testathon*, donde se hicieron pruebas sobre este software a partir de un un conjunto de casos de uso diseñado para cubrir la mayor cantidad de aspectos de la aplicación completa y, en caso que corresponda, reportar errores al equipo de desarrolladores. El objetivo de este estudio fue comprender la nueva arquitectura de DSpace 7, las tecnologías involucradas y el estado del desarrollo del proyecto. Cabe aquí destacar que, al momento de realizar dicha investigación, parte de la funcionalidad desarrollada en versiones previas de DSpace aún no había sido incorporada en la versión 7.0.

Una vez analizado el estado del desarrollo de DSpace 7, se procedió a realizar un análisis del estado actual del repositorio CIC-DIGITAL. El objetivo aquí era dimensionar el volumen de información que gestiona dicho repositorio y su organización interna. Aquí se evaluaron aspectos tales como la cantidad de recursos almacenados en el repositorio, la estructura de comunidades y colecciones, los flujos de carga definidos y el grado de personalización realizado sobre el software DSpace. A partir de este relevamiento fue posible identificar cuáles eran las personalizaciones que debían ser migradas y cuáles no, ya sea porque dicha funcionalidad ya había sido incorporada en DSpace 7.0 o porque el costo de migración superaba su aporte a los servicios del repositorio.

La siguiente etapa de este proyecto consistió en la formalización de un plan de migración de CIC-DIGITAL, que permita contar con las funcionalidades más importantes del repositorio en una nueva instancia basada en DSpace 7, descartando aquellas que en la etapa anterior habían sido identificadas como “no migrables”.

Una vez elaborado el plan de migración, se procedió con la propia migración. Para ello, se creó un entorno local de desarrollo utilizando la técnica de virtualización con Docker, lo que permitió llevar a cabo múltiples pruebas de forma segura sin afectar los servicios del repositorio.

Docker es una tecnología de virtualización que posibilita la creación y ejecución de entornos aislados y autónomos llamados contenedores, donde es posible empaquetar aplicaciones junto con todas sus dependencias. Estos contenedores son livianos, se configuran rápidamente y son portátiles, lo que significa que se pueden ejecutar sin problemas en diferentes sistemas operativos.

En nuestro caso, aprovechamos Docker para crear un entorno de desarrollo local que replica todas las aplicaciones utilizadas por DSpace. Esto incluye el contenedor web de aplicaciones Apache Tomcat, la base de datos relacional Postgres y el motor de búsqueda Apache Solr. El uso de la virtualización con Docker en este entorno de desarrollo permitió una fácil reproducción y facilitó las pruebas de migración sin afectar los servicios del repositorio en producción.

En la última etapa, una vez completada la migración en el entorno de desarrollo, pero antes de instalar esta nueva versión en el servidor de producción, se realizaron pruebas de carga y análisis de rendimiento, a fin de asegurar un adecuado desempeño del repositorio ante picos de demanda. En este sentido, se realizó un análisis cuantitativo del rendimiento simulando el tráfico que podría sufrir el repositorio en distintos escenarios. Los resultados de estas simulaciones permitieron definir muchos parámetros de rendimiento en el entorno de producción. Para definir los escenarios a simular se tomaron datos reales de uso del repositorio obtenidos de Google Analytics, lo que permitió identificar cargas máximas, mínimas y promedio en distintos períodos.

3 CAMBIOS DE DSPACE

3.1 DSpace 6 y versiones anteriores

La versión 6 de DSpace y las anteriores están compuestas de varios módulos Maven que combinados permiten generar seis aplicaciones, de las cuales dos son específicas para *frontend*: JSPUI y XMLUI. La primera está desarrollada con Java Servlets y la segunda con el *framework* Apache Cocoon. En la Imagen 1 se puede apreciar, dentro de la línea punteada roja, las seis aplicaciones correspondientes a la capa de aplicación de DSpace. Es importante observar que el desarrollo del *frontend* y del *backend* se encuentra dentro de una misma aplicación.

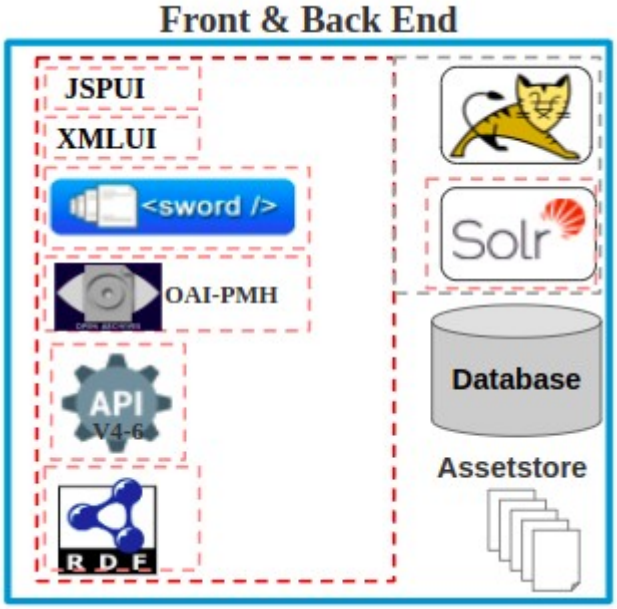

Imagen 1. Arquitectura de DSpace 6 (Donohue *et al*., 2021).

En este punto, cabe remarcar que todas las tecnologías usadas por JSPUI y por XMLUI, están, o bien en desuso (servlet, JSP, Java 8) o bien abandonadas (Apache Cocoon). Tal es así, que el propio equipo de desarrollo de DSpace decidió discontinuar el soporte de ambas herramientas.

3.2 DSpace 7

DSpace lanzó la versión 7.0 en agosto del 2021, pero es un desarrollo que comenzó en 2015. Esta nueva versión introdujo un cambio de tecnologías y de la estructura general del proyecto, separándolo en dos proyectos independientes, DSpace Backend y DSpace Frontend. DSpace Frontend implementa la interfaz de usuario en una aplicación web basada en Angular[1], mientras que DSpace Backend implementa la lógica de negocio y expone toda la funcionalidad del repositorio a través de una API REST que fue reescrita desde cero y es el principal punto de comunicación entre el repositorio y otros sistemas.

Si bien en las versiones 6 y anteriores existe una API REST que permite una interacción con el repositorio, no toda la funcionalidad podía ser usada desde esta API. Esta carencia hacía que la implementación de un *frontend*, que pudiese reemplazar a las interfaces de usuario XMLUI y JSPUI, fuese inviable.

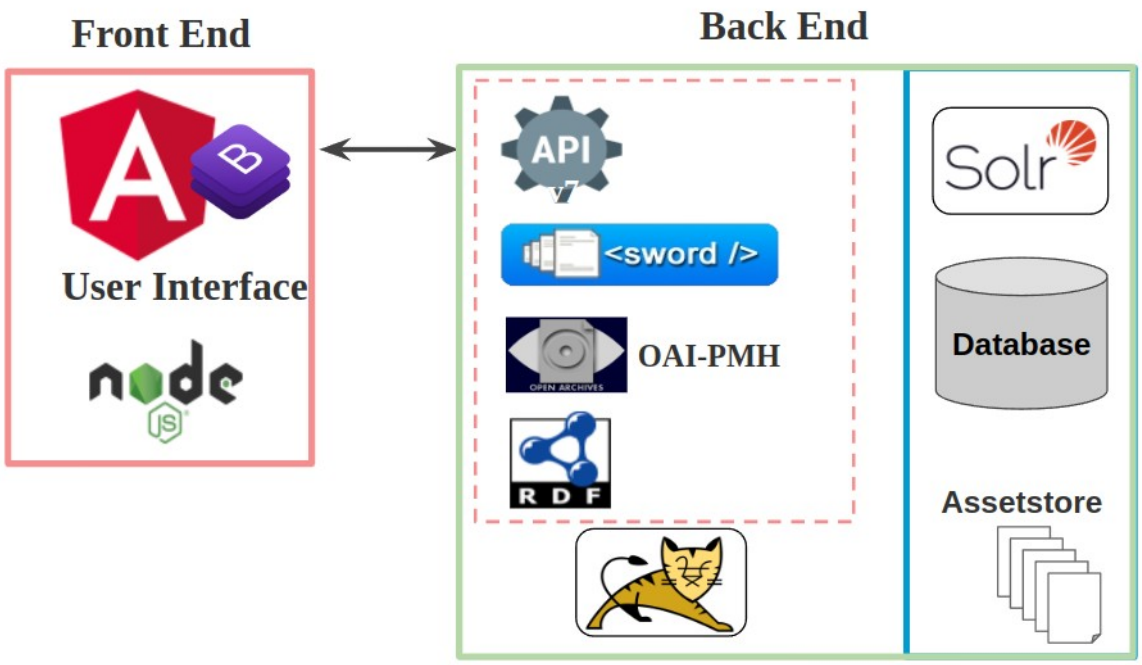

Imagen 2. Comunicación entre componentes de la arquitectura de DSpace 7 (basado en Donohue *et al*., 2021).

3.3 Arquitectura

En esta nueva versión de DSpace, se prescinden de algunos módulos, como es el caso de los proyectos que implementan la interfaz del usuario “dspace-xmlui”, “dspace-xmlui-mirage2” y “dspace-jspui” que ya no son compatibles ni se distribuyen con DSpace.

El módulo “dspace-solr” tampoco forma parte ya de DSpace 7, y la instalación del indexador Solr debe hacerse por fuera de la aplicación.

Otro módulo que se encuentra obsoleto es “dspace-rest”, reemplazado por la API definida en la aplicación web “dspace-server-webapp”. Esta es una aplicación basada en Spring Boot, la cual constituye el punto de acceso al resto de los módulos que conforman el *backend* (REST API, OAI-PMH, SWORD, SWORDv2, RDF), facilitando la instalación del repositorio, debido a que ahora solo se debe instalar una única aplicación, en vez de una por módulo.

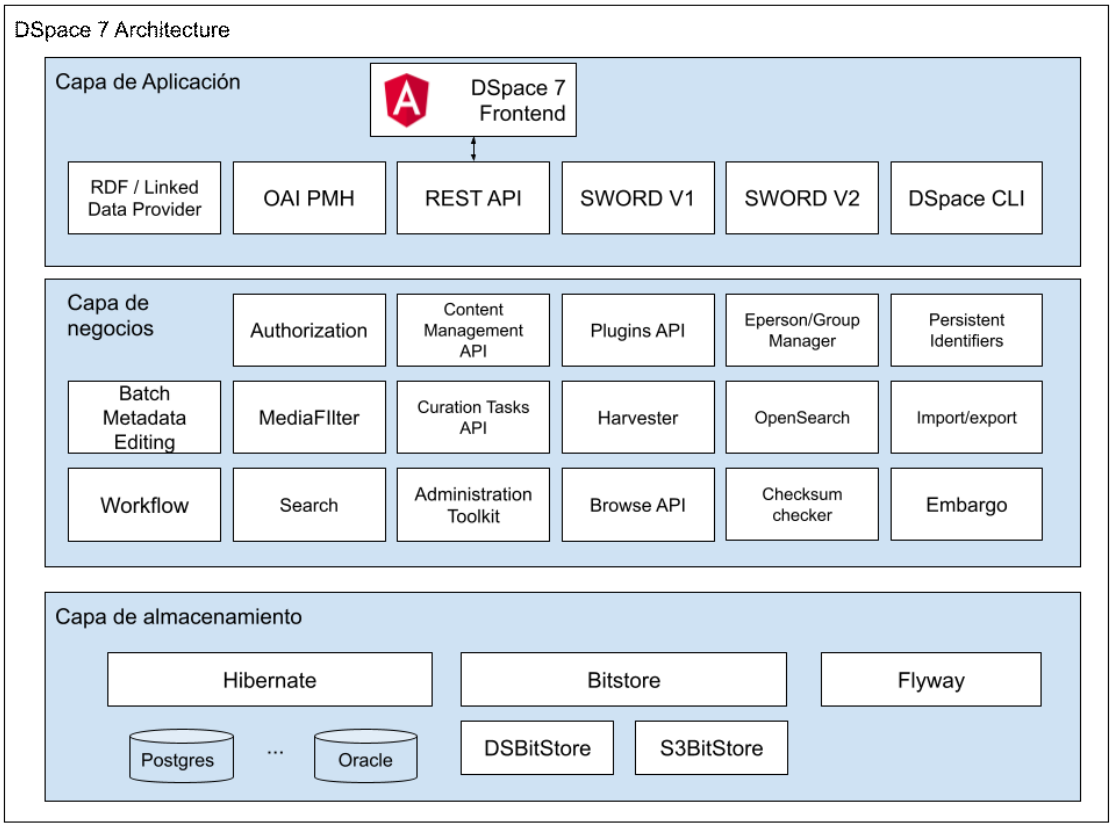

Imagen 3: Organización de componentes de la arquitectura de DSpace 7. Elaboración propia.

3.4 Entidades configurables

La versión 7 de DSpace ofrece también la posibilidad de expandir el modelo subyacente para poder representar no solo recursos, como artículos, tesis o presentaciones en congresos, sino que ahora es posible representar entidades que forman parte de un modelo CRIS (Current Research Information System), como lo son las personas, organizaciones, proyectos, revistas, etc.

Un modelo CRIS (Sistema de Información de Investigación Actual) es un sistema que permite la gestión de información relacionada con la investigación académica. Proporciona una estructura para almacenar y organizar datos sobre investigadores, instituciones, proyectos de investigación, publicaciones, entre otros elementos relevantes en el ámbito científico. En el contexto de DSpace, la versión 7 ha ampliado sus capacidades para incluir este tipo de entidades dentro del repositorio, brindando un enfoque más completo y robusto para la gestión de información académica y científica.

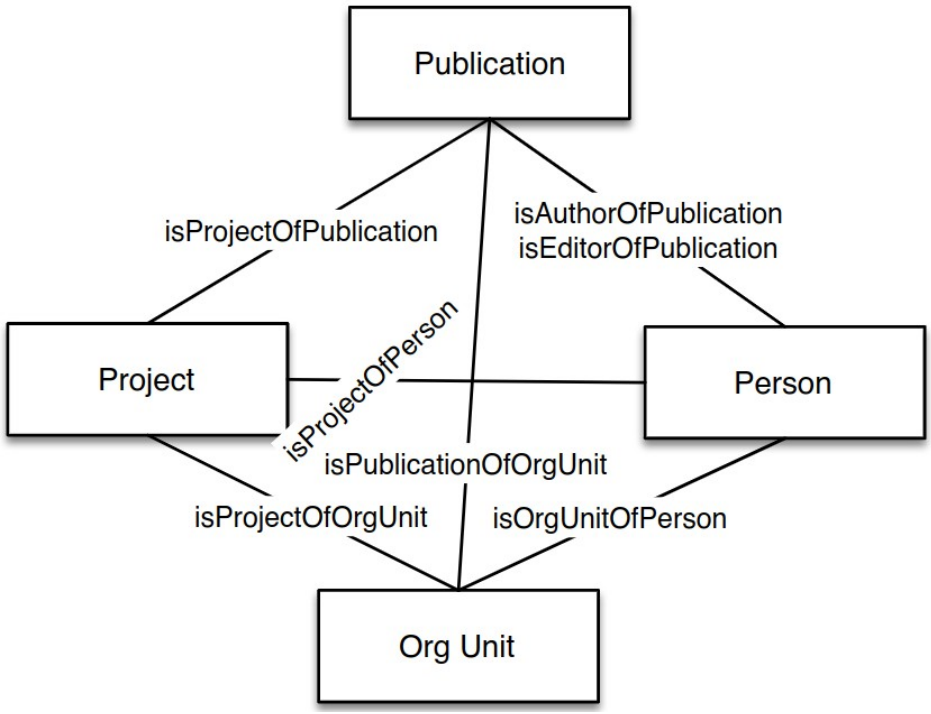

Imagen 4. (Bosman *et al*., 2021).

La adición de estas nuevas entidades permite gestionar nuevos tipos de recursos y crear páginas para cada una de las instancias de estas entidades, así como se hace con los ítems. Un ejemplo del potencial de esta nueva funcionalidad es que es posible crear perfiles de autor a partir de los valores almacenados en los metadatos que describen quién es el creador de un determinado recurso, y gestionarlos dentro de DSpace y no a través de un sistema de gestión de vocabularios controlados externo como se hace en CIC-DIGITAL (de Albuquerque, 2018).

3.5 *Frontend* Angular

Para la interfaz de usuario, DSpace 7 ofrece un *frontend* basado en Angular que reemplaza a las interfaces XMLUI y JSPUI. Angular es un *framework* de código abierto orientado al desarrollo de aplicaciones *frontend* basado en TypeScript. Este *framework* permite definir componentes reutilizables, a los que se puede dotar de funcionalidades que enriquezcan la experiencia del usuario, logrando así interfaces potentes y flexibles. Además de basarse en el lenguaje TypeScript, mucho más expresivo y seguro que Javascript, Angular se destaca por su alto grado de adopción y por una activa y creciente comunidad de desarrolladores.

El proyecto DSpace Angular provee tres directorios de temas en los que se definen los componentes y estilos que se van a aplicar en la interfaz del usuario. Estos temas son:

- Base Theme: capa que define los estilos, estructura y componentes principales de DSpace y que serán reutilizados por el resto de los temas.
- DSpace Theme: tema y estilos por defecto de DSpace basado en el tema base.
- Custom Theme: su función es ser una base para crear temas personalizados. Contiene componentes vacíos que extienden aquellos definidos en el tema base y están pensados para ser personalizados de manera sencilla.

Por defecto se utiliza el tema DSpace Theme, mientras que para poder llevar a cabo las personalizaciones y desarrollar un tema propio, a partir del Custom Theme, es necesario modificar ciertos archivos de configuración. En estos archivos es posible definir temas distintos para distintas partes de la aplicación, es decir que sería posible que todas las páginas que están bajo la ruta /items utilice el tema Custom, mientras que todo lo que está bajo /collections utilice el tema DSpace.

4 MIGRACIÓN A DSPACE 7

4.1 Beneficios

Entre los beneficios que trae DSpace 7 se puede mencionar una mayor capacidad de integración con nuevos servicios y un aumento en el grado de interoperabilidad del repositorio con otros sistemas, a partir de la

implementación de un nuevo módulo en el *backend*, que expone toda su funcionalidad a través de la nueva API REST, la cual brinda acceso a toda la funcionalidad del sistema.

La integración de estos nuevos servicios va de la mano con la propuesta de la iniciativa Next Generation Repositories (NGR) de COAR de modificar el actual modelo de repositorios centrados en los metadatos hacia un modelo centrado en los recursos (COAR Next Generation Repositories: Vision and Objectives, 2020). Esta propuesta también es impulsada por OpenAIRE a través de su programa OpenAIRE-Advance (OpenAIRE Advance Open Call Final, 2019), cuyas iniciativas promocionan el desarrollo de mecanismos que fomenten la ciencia abierta y que permitan acercar la producción académica, tanto a usuarios de la comunidad científica como también a la población general; más aún, que habiliten una participación ciudadana más activa y un mejor enfoque de los problemas sociales de naturaleza multidisciplinar.

Con el desacoplamiento del *frontend*, DSpace 7 otorga la posibilidad de implementar una interfaz de usuario más amigable, moderna y basada en componentes reutilizables, dejando de lado tecnologías que han quedado sin soporte ni actualizaciones, como ya se mencionó. Otro beneficio se encuentra en la posibilidad de extender el modelo subyacente para orientarlo hacia un sistema CRIS (Asserson y Jeffery, 2010), que gestione otros tipos de entidades, como “Personas”, “Organizaciones”, “Proyectos”, etcétera.

El salto hacia nuevas tecnologías propuesto en esta versión de DSpace, sumado a la incorporación de nuevas funcionalidades, y una nueva y moderna interfaz de usuario, hace que sea muy conveniente para un repositorio desarrollado en DSpace 6 o en versiones anteriores migrar hacia DSpace 7. Sin embargo, esta migración no siempre puede realizarse de manera directa: si bien gran parte de la lógica de negocios se mantuvo entre versiones, se deben readaptar las personalizaciones realizadas a la nueva estructura de API REST y *frontend* desacoplado, y se deben trasladar las personalizaciones en la interfaz de usuario hacia una nueva tecnología o bien realizar su rediseño.

4.2 Análisis y plan de migración

La migración de DSpace versión 6 o anterior a la versión 7 se compone de una migración de código, una migración del servidor y una migración de los datos del repositorio.

Algunas cuestiones a considerar al momento de definir un plan de migración a DSpace 7 son:

- Cantidad de recursos almacenados en el repositorio.
- Estructura de comunidades y colecciones.
- Flujos de carga definidos.
- Complejidad de los formularios de carga.
- Versión actual del repositorio.
- Personalizaciones realizadas. No solo la cantidad, sino donde fueron hechas estas modificaciones.
- Visibilidad del repositorio.
- Versionado del código. Si se realiza o no versionado y con qué herramienta se realiza.

En la siguiente imagen se propone una serie de preguntas a hacerse al momento de definir las estrategias de migración y las posibles acciones a seguir, dependiendo de cada caso:

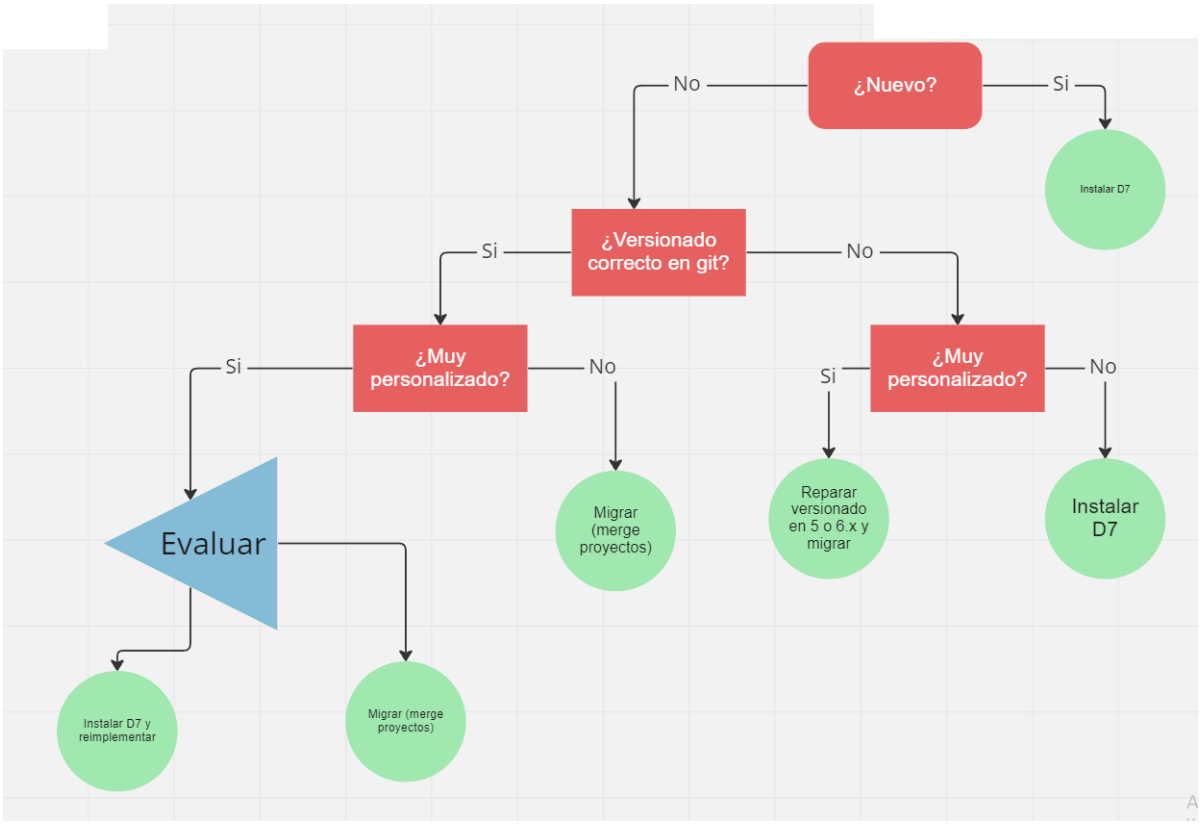

Imagen 5. Árbol de decisión para definir la estrategia de migración a DSpace 7. Elaboración propia.

Lo primero que una institución debería tener en cuenta es si su repositorio es nuevo o si ya posee uno. En el caso de la primera opción, la instalación de una nueva instancia de DSpace 7 es el camino más viable; a menos que se quiera utilizar un DSpace 6 con la motivación de utilizar alguna funcionalidad específica, lo cual no valdría la pena ya que obligaría a una migración a la versión 7 en el corto plazo por la falta de soporte que sufrirá la versión 6. El segundo punto a tener en cuenta es el versionado del proyecto: ¿está el proyecto versionado utilizando git u otra herramienta adecuada? ¿Es el proyecto una bifurcación del proyecto DSpace/DSpace? ¿Se mantiene un correcto historial de cambios? Si la respuesta a alguna de estas preguntas es negativa, quizás antes de realizar la migración se deberá corregir el versionado del código para alinearlo con el proyecto de DSpace/DSpace. Por último, para decidir la estrategia de migración de código se debe tener en cuenta el grado de personalización del repositorio y, dependiendo de ello, evaluar la estrategia de migración a seguir.

Una vez realizada la migración de todo el sistema es necesario adecuar la arquitectura sobre la que se va a hacer el despliegue tanto del *frontend* como del *backend*, teniendo en cuenta que se debe mantener la visibilidad del repositorio en la web.

4.3 Migración de código

La migración de código tiene como objetivo incorporar el código de DSpace 7 al código del repositorio, manteniendo las personalizaciones realizadas. Para esto se cuenta con dos posibilidades. La primera es realizar un *merge* entre el código de DSpace 7 y el del repositorio, mientras que la segunda se trata de implementar las personalizaciones directamente en una instalación limpia. Para poder realizar la fusión del código (*merge*), es necesario que el código del repositorio a migrar sea una bifurcación (*fork*) del proyecto DSpace/DSpace, con el código debidamente versionado. Un "*fork*" en este contexto significa que se ha creado una copia independiente del repositorio original de DSpace/DSpace, permitiendo trabajar en él de forma separada.

El hecho de tener un *fork* del proyecto implica que se han realizado modificaciones específicas en el código. Estas modificaciones pueden incluir cambios, mejoras o adiciones de funcionalidades según las necesidades del proyecto. Además, es importante que el código esté adecuadamente versionado, lo que implica llevar un registro de los cambios realizados a lo largo del tiempo.

Si el código del repositorio a migrar no es un *fork* de DSpace/DSpace con un adecuado control de versiones, no será posible realizar la fusión directa con el repositorio original. En ese caso, los cambios realizados deberán ser replicados manualmente en el repositorio adecuado.

En el caso de la aplicación *frontend*, al haberse eliminado los módulos XMLUI y JSPUI que implementan la interfaz del usuario en DSpace 6, las personalizaciones hechas deberán ser desarrolladas desde cero en Angular, o bien dejadas de lado.

Para la migración del *backend*, una de las opciones es realizar un *merge* con el código de DSpace, ya que de esta forma se puede preservar el historial de cambios realizados y mantener una continuidad en el desarrollo del proyecto. Esta tarea no es siempre viable y su dificultad va a depender de la cantidad de personalizaciones realizadas, de su complejidad y de la documentación que respalde los cambios realizados. Puede ocurrir que algunas personalizaciones hayan sido realizadas por personas que ya no se encuentren en el equipo de desarrollo del repositorio y, si no se realizó la documentación de manera clara, el entendimiento y la relevancia de los cambios realizados puede llegar a ser dificultoso.

Para realizar el *merge*, lo primero que se debería hacer es identificar las personalizaciones ya que una vez fusionado el código de ambos proyectos es probable que se generen conflictos. Para solucionar los conflictos es necesario comprender que no todas las personalizaciones tienen el mismo grado de importancia, y definir la prioridad de cada una va a permitir ayudar a resolverlos, ya sea en favor del código personalizado, a favor del código de DSpace o introduciendo una corrección manual para que coexistan ambas modificaciones.

Es posible que algunas de las personalizaciones sean incluidas en futuras versiones de DSpace, por lo que es importante estar al tanto del desarrollo del proyecto para evitar conflictos a futuro. En dicho caso, se podría descartar la personalización y esperar a que sea implementada por DSpace. En caso de que sea necesaria la implementación de una personalización prioritaria, y esta se encuentre planificada para versiones futuras de DSpace, es posible implementar dicha solución y hacer un *pull request* para que sea incorporada al código del proyecto DSpace/DSpace. En caso de que la personalización ya sea parte del código de DSpace, pero no forme parte de una versión estable, se podría hacer un *cherry pick* de los *commits* que interesa preservar.

Para esto es posible consultar el Github de DSpace, donde se puede ver no solo los *commits* agregados a partir de la última versión estable, sino que es posible ver los *issues* reportados, su estado, en qué versión se planean resolver, entre otros.

Ante la imposibilidad de resolver todos los conflictos, o en caso de que se dificulte demasiado su resolución, se podría recurrir a una estrategia de nuevo comienzo, que implica la instalación de una instancia de DSpace 7 limpia y reimplementar aquellas personalizaciones que sean vitales para la institución.

Como resultado de la migración de código se espera un proyecto DSpace que pueda ser compilado de manera correcta, al que, en este punto, le restaría incorporar los datos (recursos, comunidades, colecciones, índices de estadísticas, etc.).

4.4 Migración de datos

Para hacer la migración de datos existen dos posibilidades[2]. La primera posibilidad consiste en utilizar una herramienta que posee DSpace para realizar migraciones automáticas entre las distintas versiones. Esta herramienta se utiliza desde la interfaz CLI de DSpace y hace uso del software Flyway, muy utilizado para realizar control de versiones en bases de datos. Para ejecutarla se debe hacerlo sobre una instancia de DSpace en funcionamiento y su ejecución modifica el modelo de la base de datos relacional para adaptarlo a los cambios introducidos en la nueva versión. De esta manera, se puede continuar utilizando la misma base de datos que se estaba usando hasta entonces, pero con una estructura actualizada y sin perder ningún dato. La segunda posibilidad consiste en hacer una exportación de archivos AIP de la vieja instalación, y luego en la nueva instalación en una nueva base de datos, realizar la importación de estos archivos. Esta exportación e importación permite migrar ítems, comunidades y colecciones, pero no así otros datos como los envíos que aún se encuentran en el flujo de trabajo (*workflow*) donde los revisores realizan la curaduría de nuevos recursos.

4.5 Adecuación del entorno de producción

Un aspecto importante a tener en cuenta a lo largo de la migración es que no debería verse afectada la visibilidad web del repositorio. Para eso es necesario garantizar que los identificadores persistentes definidos hagan las

redirecciones esperadas, que los tiempos de respuesta del repositorio sean los esperados, que no surjan errores, que se mantenga la interoperabilidad con otros sistemas como OAI, y mucho más.

Al cambiar las tecnologías sobre las que se desarrolla DSpace, es necesario adaptar la arquitectura sobre la que funciona el repositorio. Anteriormente, DSpace contaba con un conjunto de aplicaciones web que funcionaban dentro de un contenedor web como Apache Tomcat. Ahora, al usar un *frontend* basado en Angular, se debe agregar al menos un *process manager*[3] que sea capaz de gestionar proyectos Node.js[4] en un escenario de producción. Según el escenario de funcionamiento, también puede ser necesario incluir otros servicios de *proxy* reverso y/o *caching*.

5 MIGRACIÓN DE CIC-DIGITAL

5.1 Características

CIC-DIGITAL fue originalmente desarrollado sobre DSpace 4 y actualizado en repetidas ocasiones hasta llegar a la versión 6.4-snapshot, es decir, la última versión disponible en la rama 6, previa a DSpace 7. Si bien 6.4-snapshot no es un *release* formal y estable, esta versión contiene modificaciones, parches de seguridad y arreglo de errores de importancia para el correcto funcionamiento del repositorio y es casi idéntica a la versión 6.4 finalmente publicada en julio de 2022.

La interfaz de usuario de dicha versión está desarrollada íntegramente con XMLUI, y para la personalización de dicha interfaz se creó un tema propio, denominado cicba[5], el cual es una extensión de Mirage, uno de los temas incluidos en DSpace. Cuenta también con un sistema de autoridades implementado en el sistema de gestión de contenidos Drupal[6] en versión 9. La interconexión entre Drupal y DSpace se realiza mediante una API REST y una serie de conectores a medida desarrollados sobre CIC-DIGITAL para tal fin.

CIC-DIGITAL posee además una serie de personalizaciones sobre DSpace realizadas a partir de los requerimientos de la institución, entre las que se pueden mencionar la implementación de conectores con vocabularios controlados para el control de autores, instituciones y taxonomías, personalización del *workflow*, mapeos de diversos *metadatos* a OAI para compatibilidad con el SNRD, personalización de los formularios de carga de acuerdo al esquema de *metadatos* y a los tipos de recursos del repositorio, y modificaciones a los módulos de OpenSearch y Request a Copy.

Para la puesta en producción se contaba con tres máquinas virtuales, una para la base de datos en Postgres, otra para el sistema de vocabularios controlados basado en Drupal y la última contenía el *assetstore*, el índice de Solr de DSpace y un Tomcat que desplegaba el código del repositorio. En la imagen 6 se muestra cómo era el conjunto de tecnologías y máquinas virtuales involucradas antes de la migración.

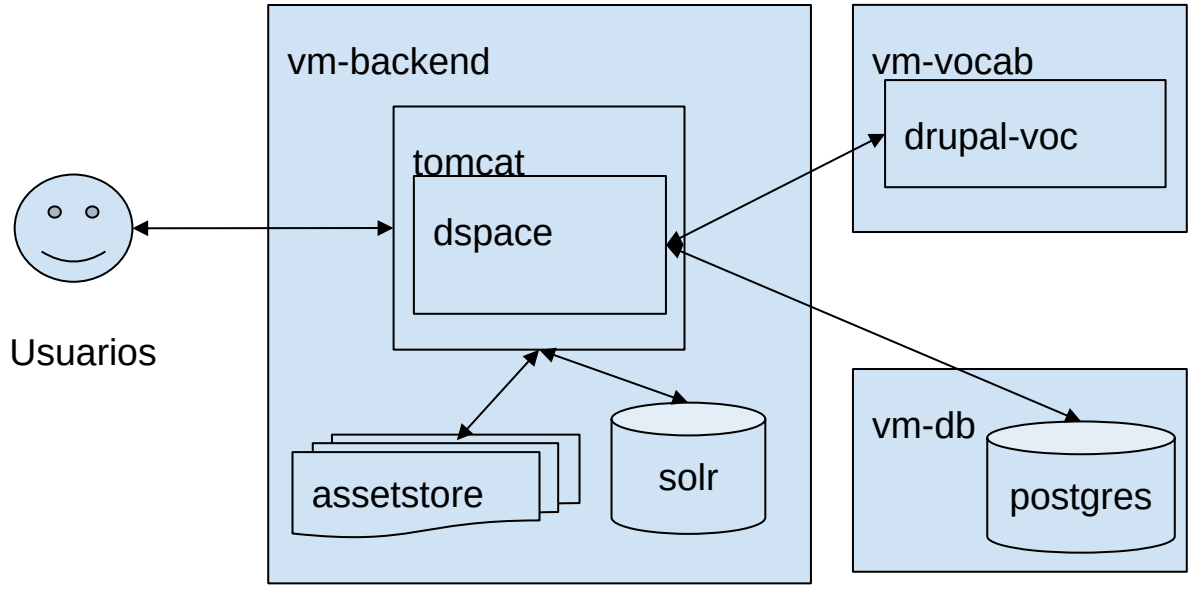

Imagen 6. *Stack* de tecnologías en CIC-DIGITAL 6. Elaboración propia.

5.2 Proceso de migración para CIC-DIGITAL

5.2.1 Migración de código

En primera instancia, se relevó la funcionalidad agregada y eliminada en la nueva versión para así entender qué cambios debía sufrir el código de CIC-Digital.

El primer paso fue borrar los proyectos que ya no forman parte del desarrollo como dspace-jspui, dspace-xmlui y dspace-solr, con el fin de evitar conflictos. Luego se realizó el *merge*[7] del código de DSpace correspondiente a la última versión estable de la rama 7.x, que al momento de hacer la migración era la 7.2.1.

El resultado inicial del *merge* fue trescientos conflictos en archivos modificados, por lo cual se decidió categorizarlos con el uso de *git*, de forma tal que permitiera resolver la mayor cantidad de manera genérica, dejando para el final aquellos conflictos que resultaran más complejos.

Los conflictos generados a partir de modificaciones realizadas por el propio desarrollo de DSpace 7, es decir por las diferencias entre las versiones 6 y 7, se resolvieron de manera genérica en favor de los cambios en DSpace 7. En cambio, los conflictos que surgieron por las personalizaciones realizadas por CIC Digital, se analizaron de manera individual.

Luego, se adaptaron las extensiones propias CIC-DIGITAL para que funcionasen en esta nueva versión y el código pudiese compilar en DSpace 7. Por ejemplo, dado que en DSpace 7 se cambió el mecanismo de selección de una autoridad fue necesario adaptar los conectores[8] del módulo de autoridades externas de CIC-DIGITAL.

5.3 Implementación del *frontend*

El *frontend*, por su parte, requirió de una iteración continua de diseño de la nueva interfaz[9], implementación de lo propuesto y pruebas (*testing*) de lo realizado, hasta llegar al punto de tener una interfaz de usuario usable y funcional, apta para ser utilizada por los usuarios del repositorio.

Para el *frontend* se debió rehacer la interfaz de usuario, lo que requirió de numerosas acciones de diseño, implementación y pruebas hasta obtener la interfaz de usuario final. A modo descriptivo, se enumeran algunas de estas acciones:

- Armado de *mockups* (maquetas) y prototipos de todo el sitio: se priorizó la página de inicio, la vista de un ítem, la página de búsqueda y la visualización de los centros CIC. Se eligieron estas páginas en particular por ser las más visitadas por los usuarios y las que definen la identidad estética del sitio.
- Implementación de lo diseñado sobre DSpace.
- Pruebas: el desarrollo en progreso se puso en funcionamiento en un servidor de pruebas para que los administradores del repositorio testearan y reportaran los cambios y nuevas funcionalidades.

El proceso fue iterativo, y en cada ciclo se fue refinando cada vez más el detalle del diseño. Las primeras iteraciones se enfocaron más en la estructura general, colores y tipografía de las páginas; las últimas se enfocaron en el detalle de diseño y funcionalidad de cada sección y elemento de cada página en particular.

5.4 Funcionalidad faltante

También se realizó un análisis funcional para detectar la funcionalidad faltante que existe en CIC-DIGITAL pero que no está implementada todavía en DSpace 7. Se encontró que algunas funcionalidades de la versión 6 no se migraron a las primeras versiones de DSpace 7. Entre ellas, podemos mencionar el módulo de Request a Copy[10], y personalizaciones realizadas en Opensearch por el lado del *backend*, y la funcionalidad de *type-bind* de *metadatos* en el *frontend*. Estas funcionalidades no fueron implementadas por la versión 7.0 de DSpace, pero sí se encuentran presentes en versiones siguientes. En algunos casos, se optó por esperar a la migración a esas versiones para poder hacer uso de esas características, pero en los casos donde la funcionalidad era primordial para el funcionamiento del repositorio, como es el caso del *type-bind*, se implementaron soluciones temporales.

5.5 Migración de datos

Para la migración de datos se utilizó la herramienta disponible desde la interfaz CLI de DSpace que hace uso del software Flyway-db, descrita previamente, a través del comando dspace database migrate. Dado que el esquema original de la base de datos de DSpace 6 no había sido alterado, la ejecución del comando no generó inconvenientes, y a su término se generó la nueva estructura de la base de datos relacional para DSpace 7. En caso de haber realizado cambios mayores en la estructura de la base de datos, como por ejemplo cambios en tipos de datos de columnas, en las

reglas de validación o en las relaciones entre tablas, esta etapa podría no resultar tan directa, y quizás sea necesario llevar la base de datos hacia el esquema original antes de utilizar Flyway-db.

6 ARMADO DEL ENTORNO DE PRODUCCIÓN

Una vez terminada la migración del código y testeada la aplicación, hubo que preparar los servidores para la salida a producción de la nueva versión de CIC Digital. En ese sentido, DSpace provee de algunas recomendaciones para la puesta a punto del sistema en un entorno productivo:

- Para el *Backend* se debe tener corriendo el servidor en HTTPS; de otra manera fallarán los *logins*. También se recomienda la instalación de un Apache o un Nginx para HTTPS y como *proxy* reverso el Tomcat.
- Para el *Frontend*, DSpace recomienda utilizar un gestor de procesos Node.js para la interfaz en producción, por ejemplo, PM2 (Production process manager). Así como se hizo para el *backend*, se recomienda la configuración de un *proxy* reverso Apache o Nginx con HTTPS.

Es posible tener el *frontend* y el *backend* en la misma máquina o en máquinas separadas, con un mismo *proxy* para las dos aplicaciones o uno para cada una. Si se tienen el *backend* y el *frontend* en la misma máquina, se puede directamente configurar un solo Apache o Nginx en esa máquina como *proxy* tanto para el *backend* como para el *frontend*.

En CIC Digital se utilizó una estrategia similar a la sugerida por DSpace: se utilizaron dos servidores, uno para el *frontend* y otro para el *backend.* Si bien en un principio se intentó utilizar un solo *proxy* reverso Apache para las dos aplicaciones, esto resultó en una sobrecarga de *request* para el servidor, el cual no podía soportar todo el tráfico de las dos aplicaciones con los recursos que se le habían asignado. Por lo que luego se optó por configurar un *proxy* reverso propio para cada aplicación. Para el *backend* se mantuvo el servidor Apache como reverso hacia la API y al OAI, para el *frontend* se eligió utilizar un servidor Nginx. En la Imagen 7 se muestra cómo quedó conformado el *stack* de tecnologías que conforman el repositorio.

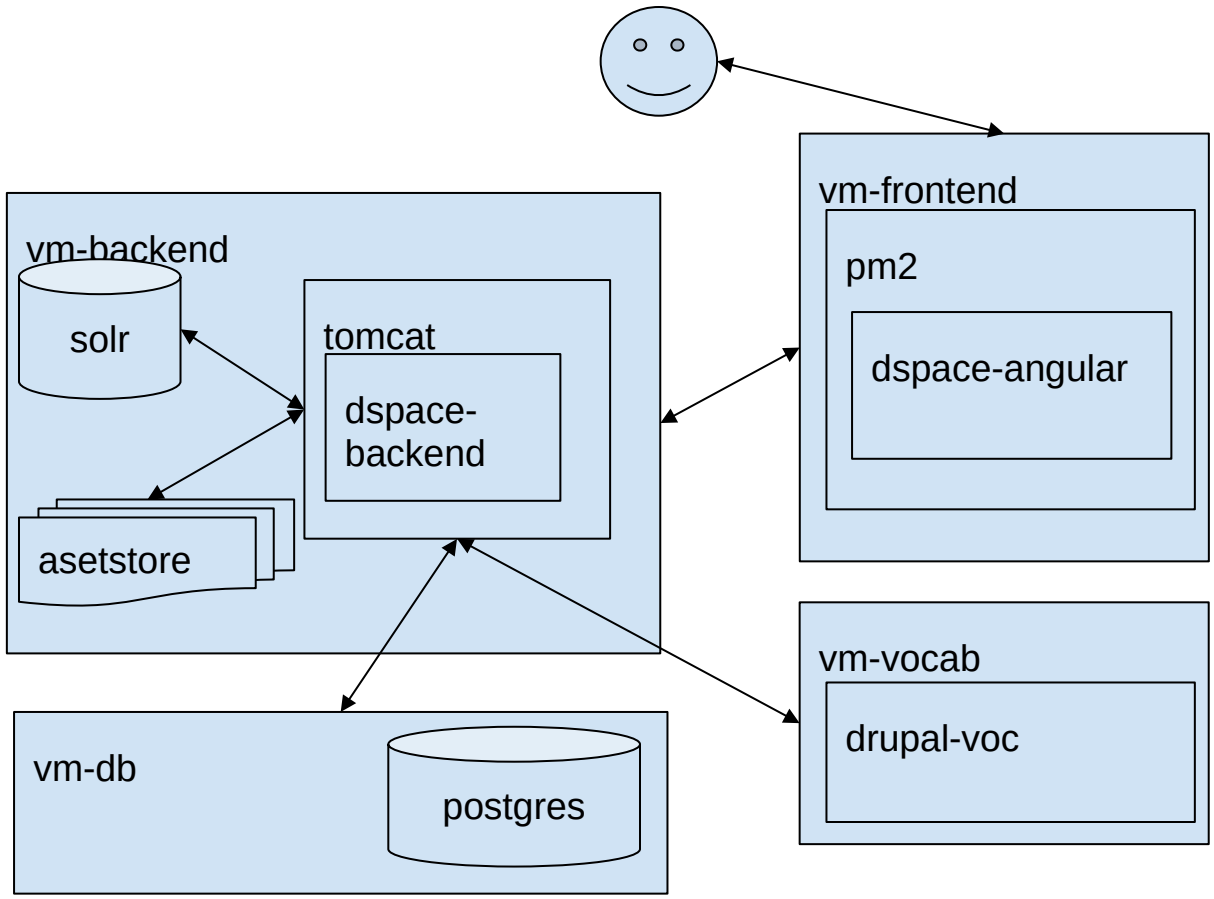

Imagen 7. Arquitectura de CIC-DIGITAL 7. Elaboración propia.

Una vez que se adecuó la arquitectura del repositorio, se realizaron pruebas de estrés al servidor del *frontend* para simular un uso intensivo de la aplicación y realizar un análisis de los tipos y tiempos de respuesta de las dos aplicaciones. Para realizar estas pruebas, se relevó el tráfico de CIC digital, registrado en una cuenta de Google Analytics[11], haciendo énfasis en la cantidad de sesiones en un día normal, en un momento de mucho tráfico y en los momentos donde hay menos tráfico. A partir de esos datos, se realizaron distintos tests de estrés con el uso del software Artillery[12], teniendo en cuenta esos tres posibles casos. También se testearon casos atípicos en donde la cantidad de sesiones y usuarios fuera mayor que un día de mucho tráfico, para poder obtener datos sobre cómo se comportaba la aplicación y el uso de los recursos en un caso de uso extremo.

Con los resultados obtenidos en los tests (tiempos de respuestas de la página, cantidad de sesiones exitosas, conexiones perdidas debido a tiempo de espera agotado) se realizaron cambios de configuración en los servidores, en especial a los *proxy* Nginx y Apache y a PM2, para mejorar el rendimiento del sistema. Una vez cambiada la configuración se volvió a testear, y se repitió este ciclo de pruebas y re-configuración hasta que los servidores pudieron soportar la cantidad de *requests* y sesiones definidas en los distintos tests de manera satisfactoria.

7 REFLEXIONES SOBRE EL PROCESO DE MIGRACIÓN

7.1 Estrategia para nuevos repositorios

Como se ha visto anteriormente, la migración de DSpace 6 (o versiones anteriores) a DSpace 7 puede requerir una gran inversión de tiempo y recursos técnicos, que se acentúa cuando se trata de implementaciones con un gran número de personalizaciones. Es por eso que la adopción desde el inicio de la versión 7 sea tal vez la aproximación más indicada. Con esta adopción temprana una institución no solo se asegura de tener un *stack* de tecnologías actualizado, sino que también tiene la posibilidad de integrarse al desarrollo del código de DSpace, aportando en base a la experiencia que se va generando a lo largo del desarrollo.

Es cierto que no todas las funcionalidades de las versiones anteriores están disponibles al día de hoy (en la versión 7.3), y que quizás un repositorio quiera hacer uso de alguna de ellas. Pero el costo de utilizar la versión 6 de DSpace para hacer uso de alguna funcionalidad específica, para luego migrar a 7, es mayor que el de instalar DSpace 7 desde el inicio y luego esperar a que esa funcionalidad esté disponible en versiones posteriores o bien implementarla por cuenta propia.

7.2 Posibles estrategias para repositorios existentes

En caso de que se tenga un DSpace 6 y una institución quiera dar el salto a DSpace 7, la estrategia que se puede utilizar para migrar puede ser similar a la usada en CIC-DIGITAL. En primer lugar, habría que identificar las personalizaciones realizadas y de ser posible priorizarlas. Una vez que se incorpore el código de la versión 7 a una rama de desarrollo, los conflictos resultantes van a indicar la cantidad de trabajo que puede conllevar la migración.

7.3 Posibles estrategias para repositorios muy personalizados

En caso de que los conflictos sean numerosos y afecten el código de las personalizaciones incorporadas a lo largo del desarrollo, una buena opción es repensar el desarrollo para arrancar desde una instalación limpia. Si bien esta opción parece demandar un gran esfuerzo, si se tiene en cuenta que como mínimo será necesario desarrollar todo el *frontend* desde cero, quizás pensar en un desarrollo integral que incluya tanto *frontend* como *backend* puede ser una buena opción.

7.4 Trabajo a futuro

A partir de la migración de CIC-DIGITAL a DSpace 7 es posible extender el modelo para definir nuevas entidades como personas y organizaciones. De esta forma es posible armar perfiles de autores que agrupen su producción científica en una página, como si se tratase de un recurso más del repositorio, y ofrecer servicios como estadísticas de uso de los recursos del investigador, agrupar métricas de impacto, crear grafos a partir de las relaciones entre distintos investigadores, etc. Para esto es necesario recopilar información de los investigadores de la CIC para armar los perfiles lo más completos posible. Una posible fuente de información para realizar esta tarea es el sistema de vocabularios controlados que utiliza CIC-DIGITAL, que actualmente controla los valores que se almacenan en los metadatos dedicados a representar investigadores en un ítem. También es posible cruzar información de los investigadores a partir del uso de identificadores persistentes provenientes de diversas infraestructuras académicas globales destinadas a la identificación de obras, datos de investigación, personas, instituciones, organismos de financiamiento, proyectos, y más.

Otro asunto a tratar es que el buscador académico Google Scholar indexa los recursos del repositorio a partir de la URL del *backend* y no la del *frontend*, lo que puede afectar la visibilidad del repositorio o al menos la generación de reportes en base a la URL del *frontend*.

Por otro lado, en DSpace 6 y versiones anteriores, se asigna un *handle* a cada uno de los recursos, y ese identificador funciona también como su URL. A partir de la versión 7, la URL de los recursos cambió y está formada por un patrón basado en el tipo de DSpace Object (ya sea ítem, comunidad, colección, etc.) y por su identificador de objeto en DSpace, el UUID. Esta modificación impactó negativamente en la trazabilidad de las estadísticas de uso en *dashboards* y reportes que utilizaban la URL con el *handle* de los recursos. Las herramientas de estadísticas no tienen manera automática de conectar una URL formada por un UUID con su correspondiente *handle* y así unir las estadísticas de uso para ambas URL. En la Imagen 8 se puede apreciar un reporte basado en la URL de los recursos a partir de su handle.

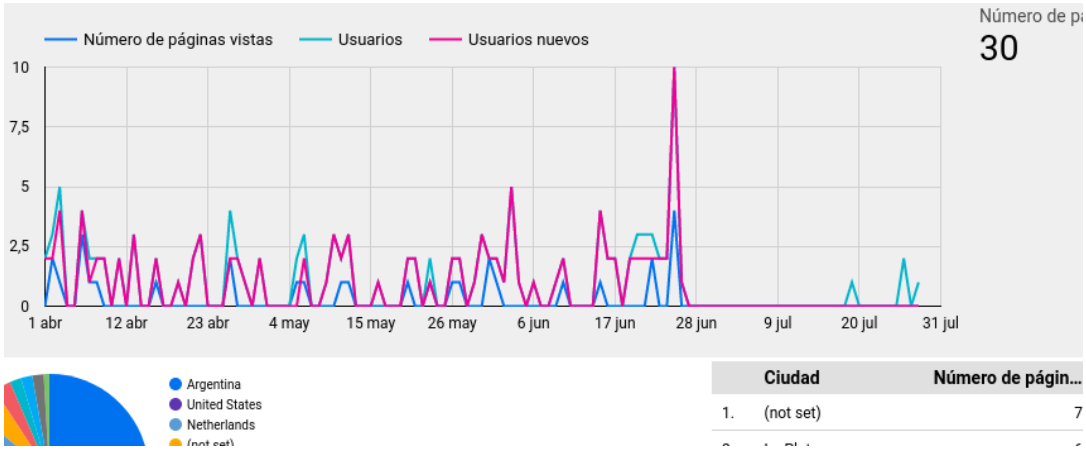

Imagen 8 - Reporte de web *analytics* basado en *handle*. (Google Analytics, 2023).

Para solucionar este problema sería deseable que la URL primaria del recurso fuese configurable y no dependiera solo del UUID.

8 CONCLUSIONES

La nueva API REST de DSpace 7 no solo hace que finalmente sea posible implementar un *frontend* desacoplado de la capa de negocios, como es el caso del proyecto DSpace Angular, sino que permite incorporar nuevos servicios o desarrollar nuevas aplicaciones que interactúen con los repositorios. Esta característica es un punto muy importante como base para el desarrollo de Next Generation Repositories (NGR) (Confederation of Open Access Repositories *et al*., 2017), una iniciativa que busca posicionar a los repositorios digitales como proveedores de servicios más que como un depósito de producción científica.

El desacoplamiento de la API REST permite implementar nuevos sistemas que no solo consuman los datos del repositorio, sino que también permite realizar tareas de gestión de usuarios, permisos y otras funcionalidades del repositorio. Con esto en mente es posible crear nuevas aplicaciones, como *frontends* dedicados a usuarios o tareas más específicas, como un sistema que gestione importaciones en masa provenientes de distintas fuentes, que implique realizar transformaciones en los datos que deban ser supervisadas por un humano.

La posibilidad de extender el modelo subyacente de DSpace es sin dudas uno de los puntos más interesantes de esta última versión. Con esta funcionalidad es posible gestionar entidades propias de un modelo CRIS como personas, proyectos, organizaciones, etc., que tienen *metadatos* acordes a cada entidad. Por ejemplo, los *metadatos* asociados a las entidades *person* se basan por defecto en el esquema Person de Schema.org (schema-person-types.xml). De la misma forma que ocurre cuando un ítem es creado en DSpace, al momento de crear una entidad Person, se creará también una página que agrupe sus metadatos de manera tal que sea posible crear perfiles de autores para aquellos que sean considerados más relevantes para la institución, ofreciéndoles un servicio de suma utilidad en los tiempos que corren (Manzur y Tettamanti, 2021).

En resumen, DSpace 7 ofrece un avance tecnológico notable en comparación con sus versiones anteriores, y los beneficios que aporta su adopción van a incrementarse a medida que su desarrollo continúe, se termine de

implementar la funcionalidad faltante y de depurar los errores propios de un software que realizó un cambio arquitectural importante e incorporó nuevas tecnologías. Sin embargo, esta adopción puede resultar compleja y requerir gran dedicación de tiempo, tanto en la implementación del software, como en la capacitación para su uso y su posterior personalización.

De acuerdo a la experiencia obtenida durante el proceso de migración de CIC-Digital, antes de tomar la decisión de migrar un repositorio estable en DSpace 6 o en DSpace 5 a la nueva versión se debe analizar detenidamente la situación de ese repositorio en lo que respecta al grado de personalización de su instalación de DSpace y en particular al personal disponible y sus capacidades técnicas. Un repositorio con un alto grado de personalización implica un proceso de migración muy complejo, debido a la necesidad de adaptar los cambios realizados a la nueva versión. En ese caso se debería evaluar el costo de migrar las personalizaciones sobre implementarlas nuevamente en una instalación de DSpace 7 “limpia”. Si, en cambio, el grado de personalización es menor o nulo, la incorporación de los cambios de DSpace 7 al proyecto del repositorio pareciera ser la opción correcta.

En ambos casos se recomienda encarar el proceso de migración solo si se cuenta con personal con las capacidades técnicas necesarias para afrontar el desarrollo. La migración podría implicar varios meses de trabajo, y si no se cuenta con un equipo técnico apto para realizarla podría ser inviable. En caso de no disponerlo y de que sea necesario capacitar al equipo de desarrollo, se debe tener en cuenta que el costo de aprendizaje de las tecnologías usadas (Docker, Angular, Java, REST, Spring, Hibernate, etc.) puede ser bastante alto. Sin embargo, las instituciones deben poner en la balanza el costo del recurso humano versus el beneficio que se puede lograr usando un software de este tipo como ofrecer mejores servicios para investigadores, la mejora del impacto académico, la adopción del modelo de ciencia abierta y más.

9 BIBLIOGRAFÍA

ASSERSON, A. y JEFFERY, K. CRIS and Institutional Repositories. *Data Science Journal*, 2010, vol. 9, n 0, p. 14-23. Disponible en: https://doi.org/10.2481/dsj.CRIS3.

RODRIGUES, E., & SHEARER, K. (2017). *Next Generation Repositories: Behaviours and Technical Recommendations of the COAR Next Generation Repositories Working Group*. Disponible en: <https://digitalcommons.unl.edu/scholcom/64> [Consulta: 02 deoctubre de 2023]

DE ALBUQUERQUE, P.C. (2018). *Soporte de vocabularios controlados y autoridades en repositorios digitales.* [Tesis, Universidad Nacional de La Plata]. Disponible en: https://digital.cic.gba.gob.ar/handle/11746/8621

DONOHUE, T.; LOWEL, A. y BOLLINI, A. *Getting Started with DSpace 7.0: Basic Training*. Open Repositories 2021 (OR 2021), Virtual. Disponible en: https://doi.org/10.5281/zenodo.4908060.

LYRASIS. (2023) *DSpace 7— Configurable Entities*. Disponible en: <https://wiki.lyrasis.org/display/DSDOC7x/Configurable+Entities>

MANZUR, E. y TETTAMANTI, S. (2021). *Desarrollo de servicios basados en perfiles académicos normalizados para autores de repositorios institucionales*. [Tesis, Facultad de Informática]. Disponible en: http://sedici.unlp.edu.ar/handle/10915/122513

OpenAIRE Advance Open Call Final (2019). Disponible en: <https://www.openaire.eu/openaire-advance-open-call-final-pdf> [Consulta: 15 de noviembre de 2022]

OpenDOAR Statistics—V2.sherpa. (2022). Disponible en: <https://v2.sherpa.ac.uk/view/repository_visualisations/1.html> [Consulta: 15 de febrero de 2023]

[1] **NOTAS**

Web oficial del *framework* Angular: <https://angular.io/>.

[2] Más información sobre ambas posibilidades puede encontrarse en el siguiente enlace: <https://wiki.lyrasis.org/display/DSDOC7x/Migrating+DSpace+to+a+new+server>.

[3] El equipo de desarrollo de DSpace recomienda el uso de PM2 como *process manager* en entornos de producción. Más información: <https://wiki.lyrasis.org/display/DSDOC7x/Installing+DSpace#InstallingDSpace-PM2(oranotherProcessManagerforNode.jsapps)(optional,butrecommendedforProduction)>.

[4] Node.js es un entorno de ejecución para JavaScript <https://nodejs.org/es/>.

[5] Enlace al repositorio Github de CIC-DIGITAL <https://github.com/CICBA/DSpace/releases/tag/dspace-cic-6.4.3>.

[6] Enlace a la web oficial del CMS Drupal <https://www.drupal.org/>.

[7] *Pull request* donde se realizó la migración a DSpace7 en CIC-DIGITAL <https://github.com/CICBA/DSpace/pull/55>.

[8] *Commit* donde se migran los conectores REST de autoridades a la versión 7 de DSpace <https://github.com/CICBA/DSpace/commit/8acf8c8c52e9748f816b0d2ed0f5ae26da9ca0a9>.

[9] Repositorio de código del proyecto CICBA/DSpace-angular <https://github.com/CICBA/DSpace-angular>.

[10] Más información sobre la funcionalidad Request a Copy: <https://wiki.lyrasis.org/display/DSDOC7x/Request+a+Copy>.

[11] Web de Google Analytics: <https://analytics.google.com/analytics/web>.

[12] Web oficial del software Artiollery.io: <https://www.artillery.io/>.