

# Experiencias en la migración de un repositorio institucional a DSpace 7

## **PABLO CÉSAR DE ALBUQUERQUE**

PREBI-SEDICI, Universidad Nacional de La Plata (UNLP) (Argentina)

CESGI-Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC) (Argentina)

[pablo@sedici.unlp.edu.ar](mailto:pablo@sedici.unlp.edu.ar)

## **SANTIAGO TETTAMANTI**

PREBI-SEDICI, Universidad Nacional de La Plata (UNLP) (Argentina)

CESGI-Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC) (Argentina)

[santit@sedici.unlp.edu.ar](mailto:santit@sedici.unlp.edu.ar)

## **GONZALO LUJÁN VILLARREAL**

PREBI-SEDICI, Universidad Nacional de La Plata (UNLP) (Argentina)

CESGI-Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC) (Argentina)

[gonzalo@prebi.unlp.edu.ar](mailto:gonzalo@prebi.unlp.edu.ar)

## **ARIEL JORGE LIRA**

PREBI-SEDICI, Universidad Nacional de La Plata (UNLP) (Argentina)

CESGI-Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC) (Argentina)

[alira@sedici.unlp.edu.ar](mailto:alira@sedici.unlp.edu.ar)

## **MARISA RAQUEL DE GIUSTI**

PREBI-SEDICI, Universidad Nacional de La Plata (UNLP) (Argentina)

CESGI-Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC) (Argentina)

[marisa.degiusti@sedici.unlp.edu.ar](mailto:marisa.degiusti@sedici.unlp.edu.ar)

## **RESUMEN**

En este trabajo se cuenta la experiencia obtenida a partir de la migración del repositorio institucional de la Comisión de Investigaciones Científicas (CIC), denominado CIC-DIGITAL, basado en

el software de repositorios DSpace a su última versión (DSpace 7). Se realiza un análisis de este software, mencionando las diferencias arquitecturales y funcionales con respecto a sus versiones anteriores. Luego se explica la estrategia seguida a lo largo del proceso de migración junto con los desafíos encontrados durante el mismo. Finalmente, se reflexiona sobre las distintas estrategias de adopción de este software para los distintos escenarios en los que puede encontrarse un repositorio institucional que pretende encarar una migración de características similares.

#### **PALABRAS CLAVE**

Repositorios digitales; DSpace 7; migración de software.

*Digital repositories, DSpace 7; software migration.*

## **Introducción**

La versión 7 de DSpace<sup>1</sup> difiere de sus antecesoras en varios aspectos, siendo uno de los más importantes el cambio en su arquitectura y la tecnología usada en la interfaz del usuario. A partir de la versión 7, el software se divide en dos proyectos independientes, un backend y un frontend que se comunican a través de una nueva API REST que expone toda la funcionalidad del repositorio. Si bien esta versión trae nuevas ventajas y facilita el desarrollo de nuevos servicios gracias al desacoplamiento del backend y la funcionalidad expuesta por la API REST, al día de hoy no toda la funcionalidad de las versiones anteriores está implementada en DSpace 7. Es por esto último que la adopción de esta nueva versión puede ser más o menos dificultosa, dependiendo del grado de desarrollo de los repositorios digitales en cada institución.

---

<sup>1</sup> DSpace 7: <https://duraspace.org/dspace/dspace-7/>

En este trabajo se busca compartir la experiencia obtenida por el equipo de desarrollo del repositorio CIC-DIGITAL, responsable del desarrollo del repositorio SEDICI de la Universidad Nacional de La Plata (UNLP), a partir de la migración del repositorio a DSpace 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 o de la implementación desde cero.

## Arquitectura de DSpace

### DSpace 6 y versiones anteriores

DSpace es el software para repositorios más utilizado en la actualidad: es utilizado por aproximadamente el 39 % de los repositorios (OpenDOAR Statistics - v2.sherpa, 2022). La versión 6 y las anteriores estaban compuestas de varios proyectos, gestionados a través de la herramienta Maven<sup>2</sup>, entre los cuales dos de ellos eran utilizados para implementar la interfaz del usuario. Estos proyectos son dspace-jspui y dspace-xmlui, desarrollados con Java Servlets y el framework Apache Cocoon, respectivamente. En la IMAGEN 1 se puede apreciar, dentro de la línea punteada roja, las seis aplicaciones correspondientes a la capa de aplicación de DSpace, entre las cuales figuran JSPUI y XMLUI. Es importante observar que el desarrollo del frontend y del backend se encuentra dentro de un mismo proyecto.

---

<sup>2</sup> Maven: <https://maven.apache.org/>

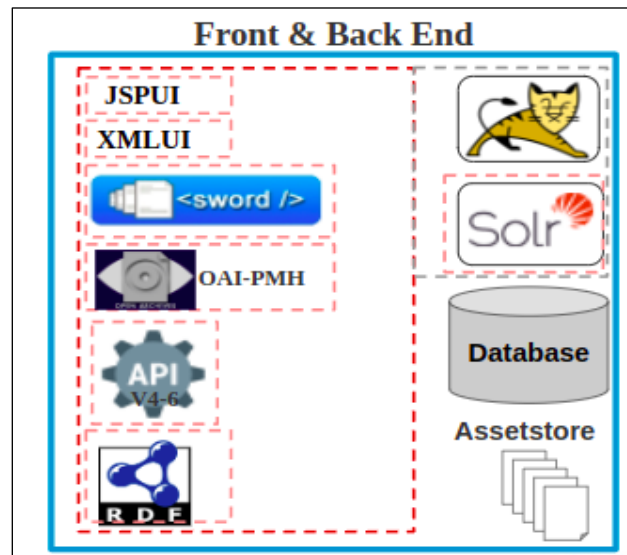


IMAGEN 1. Arquitectura de DSpace 6 (DONOHUE et al., 2021)

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.

### DSpace 7

DSpace lanzó la versión 7.0 en agosto del 2021. Esta versión promete mantener la misma funcionalidad que las versiones anteriores y además agregar nuevas características como, por ejemplo, el manejo de entidades configurables y las relaciones entre ellas. Además de la ampliación de su funcionalidad, la publicación de esta nueva versión introdujo también un cambio de tecnologías y de la estructura general del proyecto. Por el lado de la interfaz de usuario o frontend, se desarrolló una aplicación web en Angular<sup>3</sup> y de manera desacoplada del resto del sistema; esta decisión implicó la creación de una nueva interfaz de usuario, así como también la eliminación de los módulos de frontend de las anteriores versiones. Por el lado del sistema de gestión o backend, se dejó de lado el uso del software Cocoon (ya deprecado)

<sup>3</sup> Angular: <https://angular.io/>

para pasar a utilizar Spring de forma más integral, y se construyó una nueva API REST que brinda acceso a toda la funcionalidad del repositorio<sup>4</sup>.

El salto hacia nuevas tecnologías propuesto en esta versión de DSpace, sumado a la incorporación de nueva funcionalidad, a una API REST que brinda acceso a toda funcionalidad del sistema, y a la posibilidad de implementar una interfaz de usuario más amigable, 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.

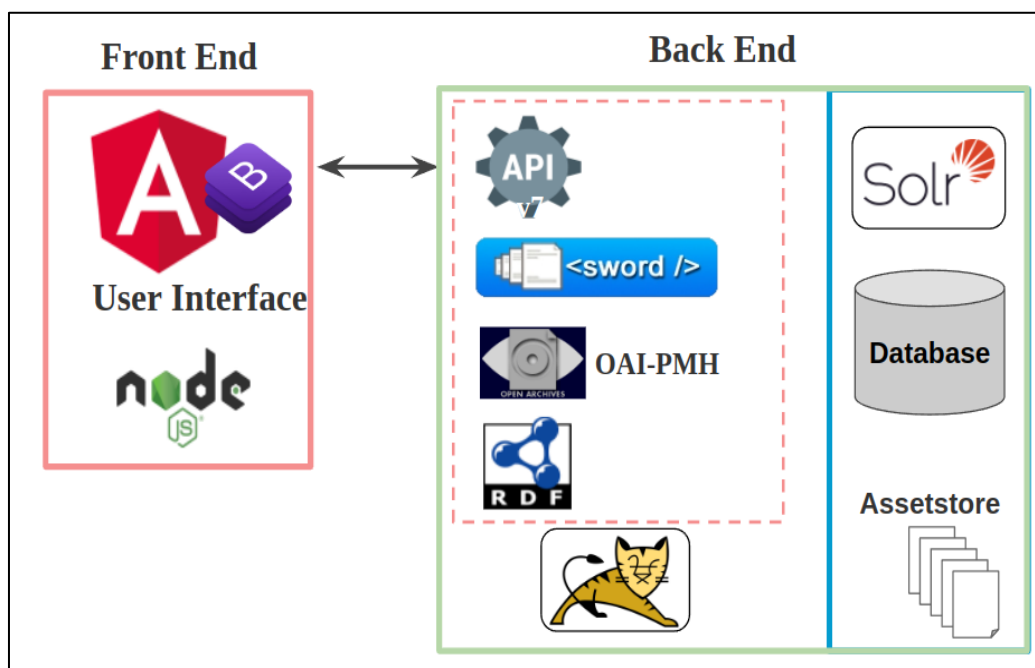


IMAGEN 2. Comunicación entre componentes de la arquitectura de DSpace 7 (basado en DONOHUE et al., 2021)

<sup>4</sup> Si bien las versiones anteriores de DSpace ya disponían de una API REST, esta brindaba acceso a un conjunto muy acotado de funcionalidades del repositorio, y su desarrollo estaba enfocado en la integración de los repositorios con sistemas de terceros o herramientas externas.

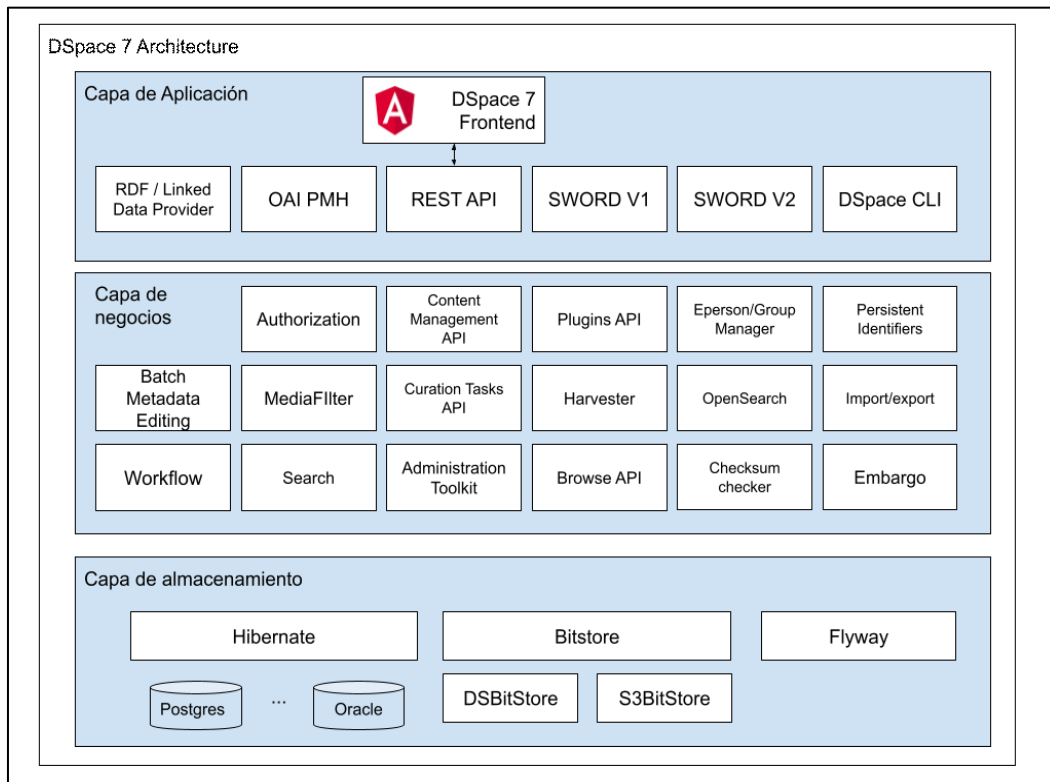


IMAGEN 3. Organización de componentes de la arquitectura de DSpace 7

### Beneficios de la nueva versión de DSpace

- Desacoplamiento del backend en un nuevo módulo, que expone toda su funcionalidad a través de una API REST, que a su vez permite integrar más fácilmente nuevos servicios.
- Implementación de una interfaz moderna basada en componentes reutilizables, dejando de lado tecnologías que han quedado sin soporte ni actualizaciones.
- Facilidad para integrar nuevos servicios, gracias al desacoplamiento antes mencionado. Esta mejora aumenta el grado de interoperabilidad del repositorio con otros sistemas a partir del uso de una nueva API REST más completa que la existente en versiones anteriores de Dspace.
- Posibilidad de extender el modelo para orientarlo hacia un sistema CRIS (ASSERSON & JEFFERY, 2010) que gestione otros tipos de entidades, como “Personas”, “Organizaciones”, “Proyectos”, etcétera.

## Funcionalidad nueva y pendiente a la versión 7.2.1

### Configurable Entities

Una de las nuevas funcionalidades de la versión 7 de DSpace recae en 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, como lo son las personas, organizaciones, proyectos, revistas, etc. A estas nuevas entidades se las llama *Configurable Entities* (DSpace 7 - Configurable Entities, n.d.) y no sólo modelan entidades, sino también las relaciones que pueden existir entre ellas. En la IMAGEN 4 se pueden observar cuatro entidades (Publicación, Proyecto, Persona y Organización) así como las relaciones que se definieron entre ellas.

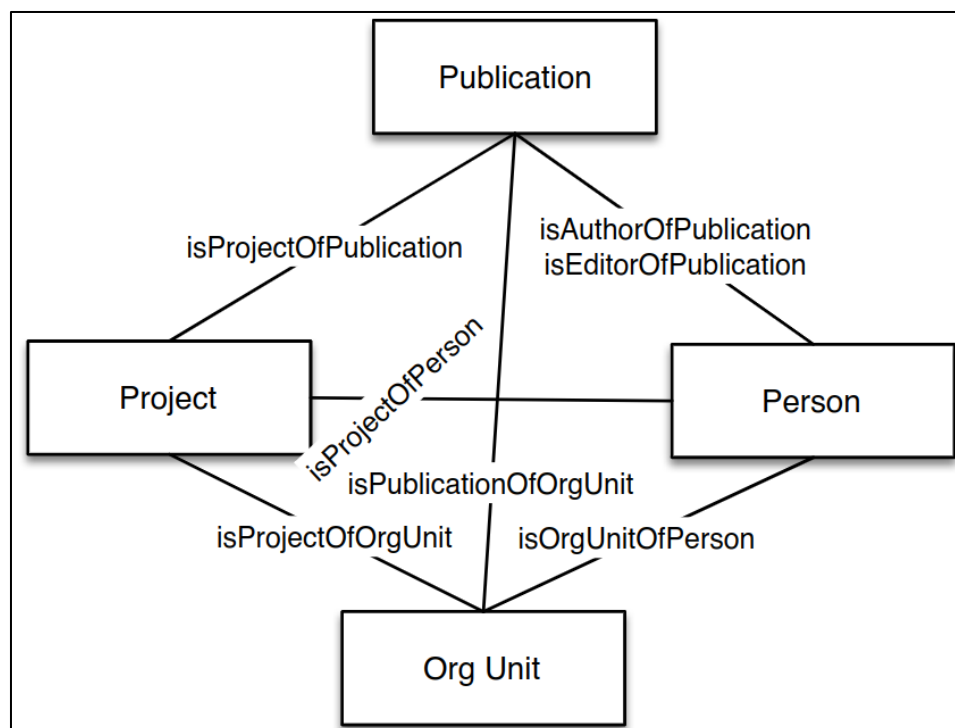


IMAGEN 4. Tomada de BOSMAN et al., 2021

La adición de las Configurable Entities 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).

### **DSpace 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 con 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 la activa y creciente comunidad de desarrolladores.

El proyecto DSpace Angular provee tres directorios de temas, entre los cuales uno de ellos define los componentes principales del frontend, mientras que los otros dos permiten su personalización. Estos temas son:

- Base Theme
- DSpace Theme
- Custom Theme

La definición de estos temas es de suma importancia ya que lo que se busca es una forma de personalizar el frontend sin tocar los componentes principales y evitar conflictos con el desarrollo principal del proyecto.



## Migración

### Características de CIC-DIGITAL

El repositorio institucional de la Comisión de Investigaciones Científicas de la provincia de Buenos Aires (CIC), 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 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 11.000 publicaciones en acceso abierto. Desde 2015 está adherido al SNRD<sup>5</sup> y es cosechado en el portal nacional de repositorios digitales.

CIC-DIGITAL fue originalmente desarrollado sobre DSpace 4 y actualizado en repetidas ocasiones hasta llegar a la versión 6.4-snapshot, es decir, a la última versión disponible en la rama 6 y 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. CIC-DIGITAL posee además una serie de personalizaciones sobre Dspace, realizadas a partir de distintos requerimientos que surgieron en algunos casos por necesidades detectadas por los administradores durante el uso del repositorio a lo largo de los años, y en otros casos a partir de las políticas propias de la institución. El stack de tecnologías utilizado en CIC-DIGITAL es el estándar de DSpace, es decir, se utiliza Solr como indexador de texto para búsquedas y estadísticas y PostgreSQL como base de datos. La interfaz de usuario está desarrollada íntegramente con XMLUI, y para la personalización de dicha interfaz se creó un tema propio, denominado cicba<sup>6</sup>, el cual es una extensión al tema Mirage que es el tema por defecto de DSpace 6.

---

<sup>5</sup> Sistema Nacional de Repositorios Digitales, Ministerio de Ciencia, Tecnología e Innovación: <https://repositoriosdigitales.mincyt.gob.ar/>

<sup>6</sup> Github CIC-DIGITAL: <https://github.com/CICBA/DSpace/releases/tag/dspace-cic-6.4.3>

Además de las adaptaciones realizadas sobre el código de DSpace, CIC-DIGITAL cuenta también con sistema de autoridades implementado en el sistema de gestión de contenidos Drupal<sup>7</sup> 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.

### **Definición de un plan de migración**

El proceso de migración se puede dividir en dos: la migración del frontend y la migración del backend. Por el lado del backend, la migración<sup>8</sup> consistió en una serie de pasos que implican la actualización del código de CIC-DIGITAL hacia la nueva versión. Esto implicó incorporar los cambios realizados por DSpace en su proyecto principal de desarrollo de la versión 7 con el proyecto de CIC-DIGITAL. El frontend, por su parte, requirió de una iteración continua de diseño de la nueva interfaz<sup>9</sup>, 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.

### **Migración de backend**

La migración del backend consistió en lo siguiente:

1. Como primer paso, antes de realizar la incorporación (*merge*) de la versión a actualizar de DSpace, se realizaron análisis y tareas de preparación para que el proceso de merge sea lo más sencillo posible:
  - Se realizó un relevamiento de cambios en la nueva versión de DSpace. Esto sirvió, además de para conocer la funcionalidad disponible en la nueva versión, para saber qué cambios iba a sufrir el código proyecto de CIC Digital.

---

<sup>7</sup> Drupal: <https://www.drupal.org/>

<sup>8</sup> Migración a DSpace7: <https://github.com/CICBA/DSpace/pull/55>

<sup>9</sup> CICBA/DSpace-Angular: <https://github.com/CICBA/DSpace-angular>

- Se analizaron los *fixes* (correcciones) superpuestos con los resueltos en CIC-Digital. Al detectar correcciones de errores realizados en paralelo en el desarrollo de DSpace 7 y en CIC-Digital (sean correcciones propias o incorporadas desde la versión 6), se pudieron prever potenciales puntos de conflicto al realizar el *merge*.
- Se borraron proyectos que no ya no forman parte del desarrollo de DSpace 7. Con el cambio de arquitectura en la aplicación que propone DSpace 7 se descontinuaron varios de los proyectos que se mantienen en uso hasta la versión 6.
  - I. dspace-jspui
  - II. dspace-xmlui
  - III. dspace-solr, ya que en la nueva versión el indexador se instala de manera desacoplada al resto de la aplicación
- 2. Se incorporó (*merge*) el código de DSpace correspondiente a la última versión estable de la rama 7.x (en el caso de CIC-DIGITAL, la versión estable al momento de hacer la migración era la *dspace-7.1*<sup>10</sup>). En este punto se identificaron algunos conflictos entre el nuevo código de DSpace y el código de CIC-DIGITAL debido a algunas personalizaciones desarrolladas.
- 3. Luego se buscó diferenciar los conflictos a partir del reporte de git. Con esta herramienta fue posible saber si el conflicto fue originado por código proveniente de CIC-DIGITAL o código proveniente del desarrollo de DSpace. Luego, se repasaron los conflictos resultantes de la integración, se analizaron los cambios implementados en CIC-DIGITAL y se definieron qué modificaciones deben conservarse.
- 4. Una vez analizados los conflictos, se procedió a resolverlos a partir de la diferenciación propuesta por git. La resolución de cada conflicto fue específica para cada caso en particular, aunque algunos se pudieron solucionar de forma genérica. Por ejemplo, un caso de resolución

---

<sup>10</sup> DSpace 7.1: <https://github.com/DSpace/DSpace/releases/tag/dspace-7.1>

genérica de conflictos se dió en archivos modificados tanto por DSpace como por CIC-DIGITAL, pero que no tenían en su contenido personalizaciones propias de CIC-DIGITAL, sino que contenían sólo modificaciones propias de DSpace. En esos archivos el conflicto surgía porque se habían realizado cambios en la versión 6 que no se incorporaron en la versión 7 de DSpace, en esos casos se determinó quedarse con los cambios realizados en la versión 7.

5. Una vez que se resolvieron los conflictos, hubo que adaptar las personalizaciones realizadas en CIC-DIGITAL para que sean consistentes con los cambios de esta nueva versión y el código pudiese compilar en DSpace 7. Por ejemplo, en el módulo de autoridades, el mecanismo de selección de una autoridad fue modificado, esto hizo necesario adaptar los conectores<sup>11</sup> definidos en CIC-DIGITAL a estas modificaciones y así para consumir del sistema de gestión de autoridades externo que se utiliza para gestionar investigadores de la CIC, instituciones y taxonomías usadas para describir items.

### **Migración de datos**

DSpace posee una herramienta 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 que haya sido actualizada de una versión de DSpace a otra más nueva; su ejecución modifica el modelo de la base de datos relacional para adaptarlo a los cambios introducidos en la nueva versión. Para la migración de la base de datos de CIC-DIGITAL, una vez que se migró todo el código se realizó una copia de la base de datos del repositorio, para no utilizar la base de datos original, a modo de prueba, y luego se utilizó la herramienta de migración para llevar la base de datos a la última versión

---

<sup>11</sup> <https://github.com/CICBA/DSpace/commit/8acf8c8c52e9748f816b0d2ed0f5ae26da9ca0a9>

disponible en DSpace 7. La ejecución de la herramienta resultó exitosa, sin mayores inconvenientes. Una vez realizadas las pruebas pertinentes en el sistema migrado, se procedió a realizar la migración de la base de datos en producción.

### **Funcionalidad faltante en *backend***

Una de las funcionalidades que no se implementaron en DSpace 7.0 fue la de Request a Copy, que permite a un usuario contactar directamente al autor de un ítem en estado embargado con el fin de solicitar su acceso. Esta funcionalidad forma parte de la versión 6 de CIC-DIGITAL, pero al no ser un comportamiento bloqueante para su uso no se volvió a implementar. Afortunadamente, este módulo sí se encuentra presente en la versión 7.1<sup>12</sup>, por lo que su configuración queda planteada para más adelante.

### **Migración *frontend***

#### **Implementación de tema propio en Frontend Angular**

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 finalizada. A modo descriptivo se enumeran algunas de estas acciones:

- Armado de *mockups* (maquetas) y prototipos de todo el sitio: priorizando página del inicio, vista de un ítem, 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 usuarios administradores del repositorio testeen y reporten los cambios y nuevas funcionalidades.

---

<sup>12</sup> <https://wiki.lyrasis.org/display/DSDOC7x/Request+a+Copy>

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.

Además de realizar el rediseño de la interfaz de usuario, también se realizó un análisis funcional del *frontend*, detectando tanto funcionalidad faltante que existe en CIC-DIGITAL pero que no está implementada por parte de DSpace, como errores en el software que no permitía un correcto funcionamiento. Para el caso de la funcionalidad propia de CIC-DIGITAL, algunos cambios se decidieron implementar en la nueva versión, a otros se los pospuso para versiones futuras, y a otros se los descartó directamente. Estas decisiones dependieron de la importancia y el uso que tienen en el repositorio las funcionalidades implementadas por esas personalizaciones. Para el caso de los errores en el software, se procedió a arreglarlos y reportarlos a DSpace junto con la propuesta de resolución del mismo.

### **Funcionalidad faltante en *frontend***

Una de las principales funcionalidades ausentes en la versión 7.0 fue la del *type-bind*. Este funcionalidad asocia un conjunto de metadatos a un tipo de recurso en particular, de esta forma se pueden crear formularios más específicos que contengan sólo la información relevante para un tipo de recurso en particular, como puede ser el metadato que representa el director de una tesis (que sólo debería aplicar para el tipo de recurso de tesis). Para poder seguir con la migración se desarrolló una solución temporal, sin embargo al día de la fecha ya existe una solución<sup>13</sup>, que aún no ha sido incorporada al proyecto, por parte de otros usuarios que crearon una solución más general del problema.

---

<sup>13</sup> <https://github.com/DSpace/dspace-angular/pull/1529>

## Desafíos encontrados en la migración de D6 a D7

### Frontend

Angular es un *framework* que permite definir componentes reutilizables en toda la aplicación; sin embargo, no tiene definido nativamente un procedimiento para definir temas que hereden funcionalidad de otros temas. Por ende, el primer desafío encontrado fue identificar en DSpace la manera en que debía definirse un tema principal para toda la aplicación a partir del cual se puedan personalizar los componentes que conforman el frontend. A lo largo de las versiones beta de DSpace 7, se ofrecía un método de adaptación del *frontend* a partir de un tema llamado mantis, pero dicho método no terminó de ofrecer una solución suficientemente genérica. En la versión 7.0 de DSpace, se ofrece una solución que reemplaza mantis<sup>14</sup> en la que se define un conjunto de componentes sin funcionalidad, que sólo hacen referencia a los componentes principales de DSpace, y que son fácilmente extensibles. Sin embargo esta solución tampoco resultó lo suficientemente genérica como para personalizar todos los componentes sin modificar aspectos centrales de la aplicación. Este tipo de modificaciones deberían ser evitables, ya que al modificar estos componentes del núcleo de la aplicación pueden llevar a posibles conflictos con el desarrollo principal de DSpace.

## Reflexiones sobre el proceso de migración

### Posibles estrategias para nuevos repositorios

Como se ha visto anteriormente, la migración de DSpace 6 a 7 puede requerir de una gran inversión de tiempo y recursos técnicos, que se acentúa más cuando se trata de implementaciones con un gran número de personalizaciones. Si bien es verdad que no toda la funcionalidad de las versiones anteriores están disponibles al día de hoy (en la versión 7.2.1), y es

---

<sup>14</sup> <https://github.com/DSpace/dspace-angular/pull/1063>

tentador para una institución que no tiene un repositorio aún funcionando hacer uso de esas versiones, a medida que se vaya personalizando la instalación, el salto a la versión 7 va a ser cada vez más costoso. Esto, sumado a que es poco probable que un repositorio nuevo utilice todas las funcionalidades de DSpace (ya que la adaptación al uso del software y su posterior personalización es un proceso que se realiza a lo largo del tiempo), podría indicar que la adopción de la versión 7 sea tal vez la aproximación indicada. Con esta adopción temprana una institución no sólo se asegura 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.

### **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 posibles 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.

### **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.



## Conclusiones

En relación a la nueva arquitectura propuesta, si bien en las versiones 6 y anteriores existe una API REST que permite una interacción con el repositorio, no toda su 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. Sin embargo, los esfuerzos que se han volcado en el desarrollo de la nueva API REST de DSpace 7 no sólo hacen 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 un proveedor de servicios más que como un depósito de producción científica.

En relación a las entidades configurables, cabe resaltar que 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 el servicio (MANZUR & TETTAMANTI, 2021).

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. Con

esto en mente es posible crear nuevas aplicaciones como por ejemplo *frontends* dedicados a usuarios o tareas más específicas, como puede ser un sistema que gestione importaciones de ítems en masa, provenientes de distintas fuentes, que implique realizar transformaciones en los datos que deban ser supervisadas por un humano.

En resumen, DSpace 7 ofrece un avance tecnológico en comparación con sus versiones anteriores, y los beneficios que aporta su adopción van a incrementarse a medida que su desarrollo continúe y 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 una cierta 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 lo recomendable quizás es la implementación de un DSpace 7 “limpio”, y luego desarrollar las personalizaciones sobre esa instalación. 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 un desarrollo basado en las tecnologías sobre las que está implementada esta nueva versión de DSpace. En caso de no disponer de ese personal técnico y 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. La migración podría implicar meses de trabajo, y si no se cuenta con un equipo técnico apto para realizarla podría ser inviable. 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 ser mejores servicios para investigadores, mejora del impacto académico, adopción del modelo de ciencia abierta y más.

## Bibliografía

- ASSERSON, A., & JEFFERY, K. (2010). CRIS and Institutional Repositories. *Data Science Journal*, adpub, 1004290223-1004290223. <https://doi.org/10.2481/dsj.CRIS3>
- CONFEDERATION OF OPEN ACCESS REPOSITORIES, RODRIGUES, E., & SHEARER, K. (2017). Next Generation Repositories: Behaviours and Technical Recommendations of the COAR Next Generation Repositories Working Group. Copyright, Fair Use, Scholarly Communication, etc. <https://digitalcommons.unl.edu/scholcom/64>
- DE ALBUQUERQUE, P. C. (2018). Soporte de vocabularios controlados y autoridades en repositorios digitales [Tesis, Universidad Nacional de La Plata]. <https://digital.cic.gba.gob.ar/handle/11746/8621>
- DONOHUE, T., LOWEL, A., & BOLLINI, A. (2021, junio 7). Getting Started with DSpace 7.0: Basic Training. *Open Repositories 2021 (OR 2021)*, Virtual. <https://doi.org/10.5281/zenodo.4908060>
- DSPACE 7—Configurable Entities. (s. f.). Google Docs. [https://docs.google.com/document/d/1X0XsppZYOtPtbmq7yXwmu7FbMAfLxxOCONbw0\\_rl7jY/edit?usp=embed\\_facebook](https://docs.google.com/document/d/1X0XsppZYOtPtbmq7yXwmu7FbMAfLxxOCONbw0_rl7jY/edit?usp=embed_facebook)
- MANZUR, E., & TETTAMANTI, S. (2021). Desarrollo de servicios basados en perfiles académicos normalizados para autores de repositorios institucionales [Tesis, Facultad de Informática]. <http://sedici.unlp.edu.ar/handle/10915/122513>
- OPENDOAR Statistics—V2.sherpa. (2022). [https://v2.sherpa.ac.uk/view/repository\\_visualisations/1.html](https://v2.sherpa.ac.uk/view/repository_visualisations/1.html)