UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

Titulo: Reportes estadísticos para repositorios digitales desarrollados en DSpace
Autores: Facundo Gabriel Adorno
Director: Dra. Marisa Raquel De Giusti
Asesor profesional: Lic. Ariel Jorge Lira
Carrera: Licenciatura en Sistemas

Resumen

Esta tesina de grado detalla la implementación de una herramienta para facilitar el análisis y comprensión de los datos estadísticos almacenados en un repositorio DSpace, de tal forma que ayude o asista en la toma de decisiones a nivel político, administrativo y operativo de una institución. Para realizar este trabajo se analizó la arquitectura del software para repositorios DSpace, de la herramienta complementaria Solr para la indexación de datos, y de los distintos módulos existentes en DSpace que agregan distintas funcionalidades basados en estos datos, de tal manera de poder abordar en el análisis de las tecnologías que se utilizarían para la implementación del prototipo. Posterior a este análisis, se comenzó con la implementación de la herramienta sobre Dspace en su versión 6 utilizando las tecnologías compatibles con esta versión (XMLUI, Apache Cocoon, XSLT, entre otras), y se desarrolló un módulo llamado Statistics-Discovery que permite la exploración mediante búsquedas sobre los datos estadísticos indexados en Solr, agregando funcionalidades de exportación de resultados en diversos formatos de texto y generación de reportes y gráficas. Finalmente, se muestran capturas de pantalla de la herramienta en funcionamiento sobre el repositorio institucional CIC-Digital, utilizado para las pruebas del prototipo.

Palabras Claves

Repositorios digitales, análisis de datos de uso, web analytics, generación de reportes, statistics, Discovery, Dspace.

Trabajos Realizados

Se desarrolló una herramienta basada en el módulo de búsqueda de DSpace, realizando varias adaptaciones para posibilitar su reuso sobre los datos estadísticos indexados en Solr. Además de la capacidad de exploración mediante búsquedas sobre estos datos, se agregó la capacidad de exportación en formatos CSV y JSON, así como la capacidad de generación de reportes y gráficas a partir de los resultados de búsqueda. Las pruebas fueron realizadas en el repositorio institucional CIC-Digital.

Conclusiones

Durante el desarrollo de este trabajo se detallaron los motivos detrás de la necesidad de crear la herramienta, realizando una investigación sobre el marco teórico del trabajo, la arquitectura de DSpace, el módulo de estadísticas existente y sus deficiencias, y el módulo de búsqueda Discovery y sus ventajas de reuso. Finalmente, se detallaron los casos de uso que la herramienta debería satisfacer y se explicó el modelo de la misma junto con el detalle de las funcionalidades implementadas.

Trabajos Futuros

Integrar la herramienta al sistema de permisos de Dspace; aumentar la cantidad de tipos de reportes predefinidos y las opciones de generación de gráficos; agregar opciones para compartir los reportes mediante botones sociales y embebido HTML; agregar estadísticas simples basadas en los reportes generados; agregar funcionalidades para la depuración de los datos analizados desde la herramienta; migrar el código a futuras versiones de DSpace y contribuir a la comunidad de desarrollo.

Fecha de la presentación: Julio 2018

Reportes estadísticos para repositorios digitales desarrollados en DSpace

Capítulo 1 | Motivación y objetivos

Motivación

Los repositorios digitales institucionales son sistemas de información que almacenan, dan acceso y preservan material intelectual proveniente de una institución académica. Este material se compone de archivos que representan una obra y metadatos que la describen. La producción científica publicada en los repositorios digitales abarca trabajos técnico-científicos, tesis académicas, artículos de revistas, entre otros, y suele ser resultado de la realización de actividades de investigación financiadas con fondos públicos ya sea, a través de sus investigadores, tecnólogos, docentes, becarios postdoctorales y estudiantes de maestría y doctorado.

Los repositorios nacen del movimiento de Acceso Abierto (AA), movimiento que aboga por el acceso libre, sin restricciones o barreras, ya sean éstas económicas o de derechos de explotación, es decir, que el acceso no sólo debe ser gratuito, sino que debe ser libre y permitir la reutilización del material intelectual producido en las instituciones.

A medida que la cantidad de material en el repositorio crece en volumen y antigüedad, también crece su estructura, típicamente definida a partir de un conjunto de comunidades/colecciones, las interrelaciones entre sus objetos y el acceso y uso por parte del público. Existen repositorios que contienen sólo unas pocas obras mientras que otros pueden llegar a tener millones, como arXiv.org que tiene más de de 1.200.000 documentos científicos. En el caso de SEDICI, el repositorio institucional central de la UNLP creado en el año 2003, aloja más de 60.000 documentos.

Debido al creciente volumen de datos en un repositorio, resulta vital evaluar el estado del mismo a través de reportes y estadísticas que simplifiquen tanta complejidad. Obtener información de un volumen de datos tan grande permite una mejor toma de decisiones por parte de las autoridades que rigen el repositorio, así como para la institución que representa. Asimismo, a través de las estadísticas se pueden detectar problemas o fenómenos indeseados que ocurren dentro del repositorio y actuar en consecuencia. Los repositorios que no realizan análisis sobre su funcionamiento, carecen de la información necesarias para realizar un adecuado control de calidad.

DSpace es el software para repositorios digitales más usado del mundo (OpenDOAR, 2018), y da soporte para la gestión, acceso y preservación de documentos digitales; este software actualmente es usado como plataforma de software principal en los repositorios SEDICI (sedici.unlp.edu.ar) y CIC-Digital (digital.cic.gba.gob.ar), utilizando este último como repositorio de prueba para la realización del presente trabajo. Este software tiene soporte para la indexación de ciertos eventos de uso sobre el repositorio, por ejemplo durante los momentos de visitar la página de una publicación, al momento de la descarga del objeto digital asociado a la publicación, al momento de realizar una búsqueda en el repositorio, entre otros. Sin embargo, el problema subyacente a esta plataforma es que no permite generar reportes estadísticos más allá de un conjunto limitado de reportes.

Tomando como repositorio de prueba al repositorio CIC-Digital (digital.cic.gba.gob.ar) y utilizando sus datos para realizar experimentación, la motivación de este trabajo es facilitar el análisis y comprensión del estado y uso del repositorio.

Objetivos

Objetivo general

- Desarrollar un prototipo que facilite el análisis y comprensión de los datos almacenados en un repositorio, de tal forma que ayude o asista en la toma de decisiones a nivel político, administrativo y operativo.

Objetivos específicos

I. Explotar de mejor manera el cúmulo de datos disponibles en los registros de acceso al contenido de los repositorios institucionales.

II. Facilitar la generación de reportes a partir de los datos en el repositorio para cualquier persona que disponga de los conocimientos básicos del dominio, evitando la complejidad de manejar las tecnologías subyacentes al soporte de los datos.

III. Analizar las distintas herramientas con las que DSpace cuenta para la realización del prototipo de herramienta, de tal forma de determinar si se puede reutilizar ciertas herramientas existentes.

IV. Permitir la exportación de los reportes generados en distintos formatos, de tal forma que pueda reutilizarse como entrada de otras aplicación.

V. Generar visualizaciones de los datos obtenidos a partir de los reportes usando distintos tipos de gráficas, que permitan una rápida comprensión de los datos.

VI. Disponer de un conjunto de reportes predefinidos de interés, que sirvan además como distintos ejemplos de uso del prototipo.

Repositorio digital

Definición

Los repositorios (Jacobs, 2016; Johnson, 2002) son archivos digitales provistos de un conjunto de servicios web centralizados, creados para organizar, gestionar, preservar y ofrecer acceso libre a la producción científica, académica o de cualquier otra naturaleza cultural, en soporte digital, generada por los miembros de una organización o comunidad. Los distintos objetos digitales en un repositorio son representados como recursos en el repositorio, y éstos últimos están conformados por un conjunto de metadatos que lo describen y brindan información acerca de ellos; es a través de los metadatos que podemos generar una síntesis y representación del objeto “real”, lo cual nos permite acceder, distribuir y difundir el contenido del recurso sin tener el objeto en sí, sino una representación del mismo. Los repositorios deben tener políticas bien definidas y deben asegurar al menos los siguientes servicios: la posibilidad de autoarchivo (depósito realizado

por el propio autor), la interoperabilidad de sus recursos con otros sistemas, el libre acceso a sus contenidos, y la preservación a largo plazo de los objetos digitales.

Existen distintos tipos de repositorios según los servicios que ofrecen y los contenidos que alojan, entre los que podemos mencionar:

- **repositorios institucionales:** almacenan, preservan y dan acceso a los materiales generados en el ámbito de una institución. Por ejemplo: SEDICI (sedici.unlp.edu.ar), CONICET Digital (http://ri.conicet.gov.ar/), DSpace@MIT (https://dspace.mit.edu).
- **repositorios temáticos**: almacenan, preservan y dan acceso a los materiales según un tema o una disciplina. Por ejemplo: PubMed Central (ncbi.nlm.nih.gov/pubmed/),
- **repositorios de datos**: almacenan y preservan datos primarios científicos. Por ejemplo: ODISEA (odisea.ciepi.org), CERN Open Data portal (opendata.cern.ch), DataHub (datahub.io),
- **agregadores:** recolectan contenidos de otros repositorios por temas, por tipo de documento, o localización geográfica. Por ejemplo: OATD (oatd.org), repositorio de tesis y disertaciones, y BASE (base-search.net), una vasta base de datos especialmente para recursos web académicos.

En particular, para este trabajo se analizarán y trabajará sobre los repositorios del tipo institucional.

Repositorio Institucional

Un repositorio institucional es (De Giusti, 2017; «Repositorios digitales - Red Infod», s. f.) un tipo de repositorio digital donde se depositan recursos derivados de la producción científica y académica de una institución o cualesquiera otros que la institución considera importantes. Sus principales objetivos son facilitar el acceso a estas producciones, aumentar la visibilidad de la producción científica de la institución, asimismo el de preservar los recursos almacenados para asegurar su accesibilidad y uso a largo plazo. Estos repositorios ofrecen un punto de acceso único a la información de la institución y de sus autores, trabajando bajo estándares de catalogación, preservación e interoperabilidad de recursos que maximizan su recuperación, compartición y accesibilidad en el tiempo. El material alojado en estos repositorios se distribuye junto al detalle de sus derechos de uso, notificando a los lectores sus usos permitidos.

Los repositorios institucionales nacen junto al movimiento de Acceso Abierto, término que deriva del inglés *Open Access* (OA). Este movimiento aboga («Open access», 2018) por el acceso libre, sin restricciones o barreras, ya sean estas económicas o de derechos de explotación, es decir, que el acceso no solo debe ser gratuito, sino que debe ser libre y permitir la reutilización de los objetos digitales.

En el caso de Argentina, la «Ley de Acceso Abierto a la Información Científica», Nº 26.899[1], establece que las instituciones del Sistema Nacional de Ciencia y Tecnología y que reciban financiamiento del Estado Nacional, deben crear repositorios digitales institucionales propios o compartidos de acceso abierto y gratuito en los que se depositará la producción

[1] http://servicios.infoleg.gob.ar/infolegInternet/anexos/220000-224999/223459/norma.htm

científico tecnológica nacional. Además se definen obligaciones para que estas instituciones aprueben políticas de acceso abierto a la producción científica, y para que sus autores depositen las obras y los datos primarios en las que éstas se basan en los repositorios institucionales.

Actualidad de los repositorios digitales

Se puede observar que con el paso del tiempo la cantidad de repositorios digitales en el mundo va creciendo notablemente, esto se debe en parte al impulso generado por el movimiento de Acceso Abierto y a la tendencia mundial de exponer la producción de las distintas instituciones académicas a través de repositorios digitales en acceso abierto.

Según uno de los servicios de estadísticas sobre repositorios más usados en el mundo, OpenDOAR, desde el 2003 una abundante cantidad de repositorios de investigación en acceso abierto había crecido en todo el mundo, aumentando rápidamente en respuesta a los llamados de académicos, de investigadores y de defensores del acceso abierto para proporcionar acceso abierto a la información de investigación, hasta llegar a los miles de repositorios registrados hoy en día. Como se ve en las figuras 2.1.1, 2.1.2 y 2.1.3, estadísticas suministradas por el servicio OpenDOAR, hasta el año 2018 se observa un crecimiento mundial en los repositorios en acceso abierto registrados en la base de datos del servicio: de los *3502* que son en total, *308* repositorios (8,79% a nivel mundial) pertenecen a Sudamérica, y *44* repositorios (1,25% a nivel mundial y 14,28% a nivel Sudamérica) son de Argentina.

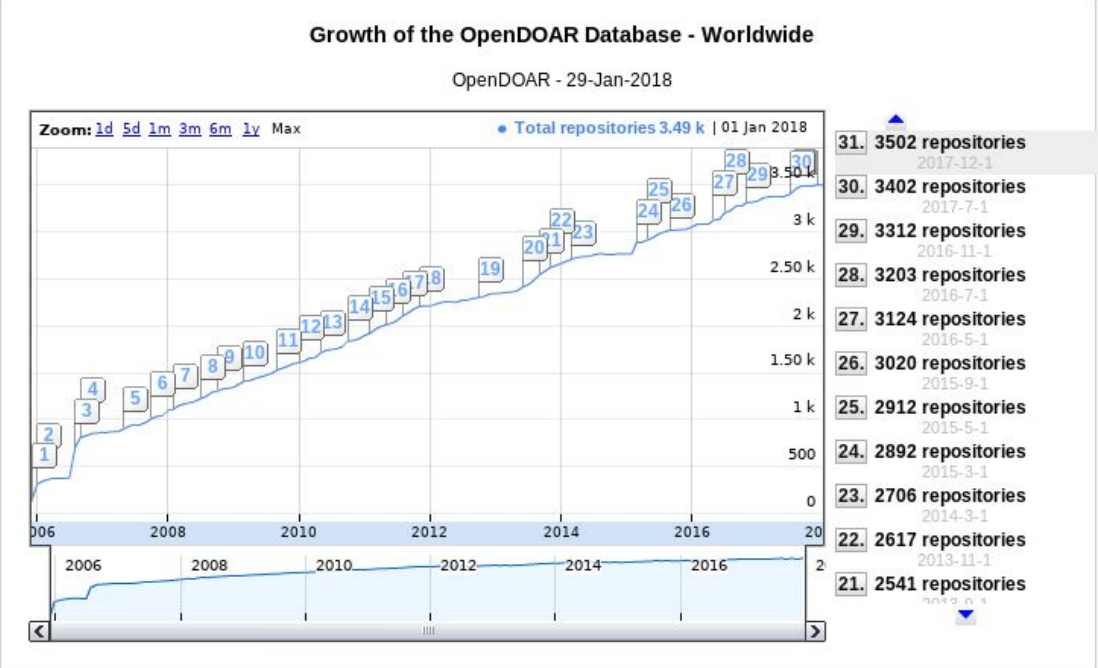


Figura 2.1.1 - Crecimiento mundial repositorios digitales (OpenDOAR)

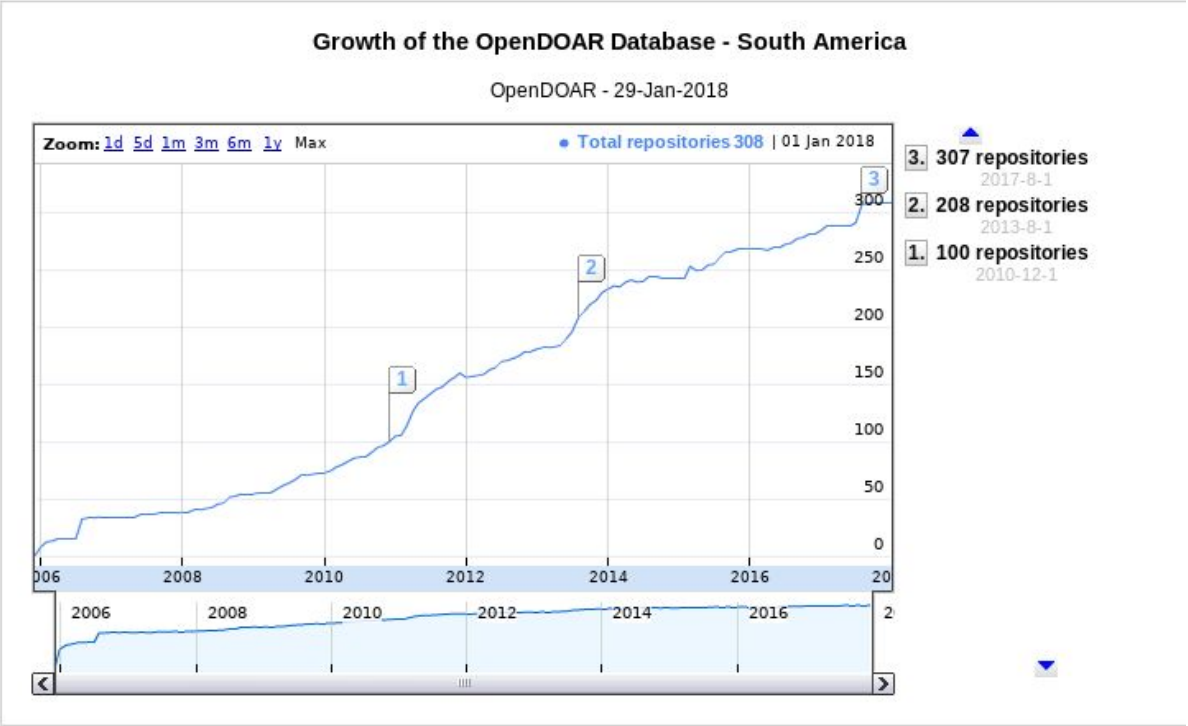


Figura 2.1.2 - Crecimiento repositorios digitales en Sudamérica (OpenDOAR)

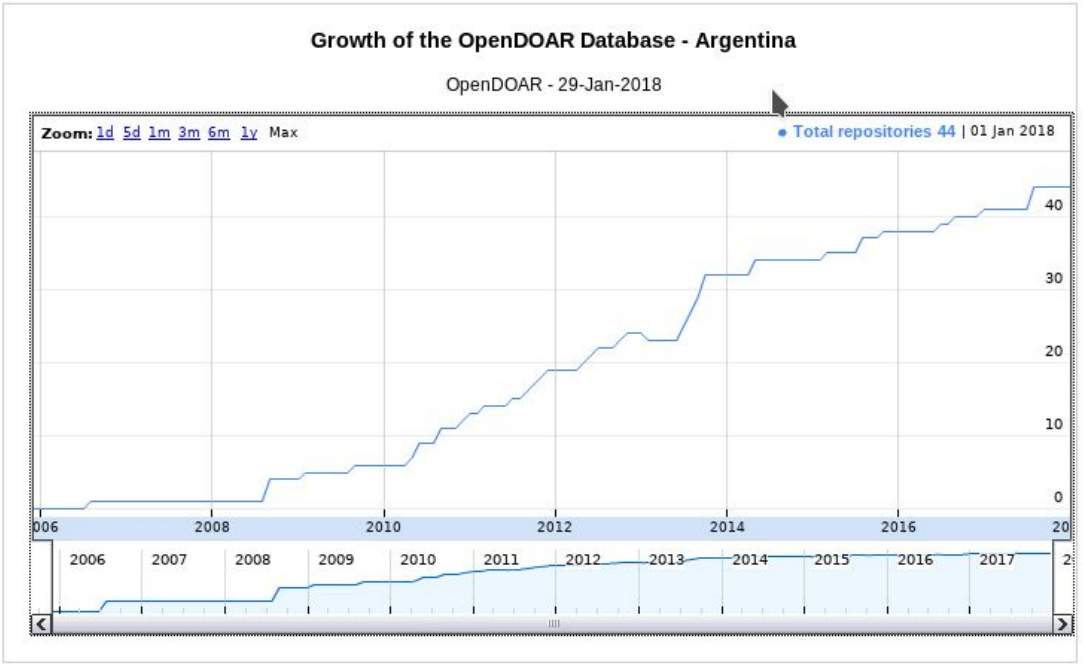


Figura 2.1.3 - Crecimiento repositorios digitales en Argentina (OpenDOAR)

Otra observación a tener en cuenta es la cantidad de repositorios por país registrados en las distintas regiones del mundo. Como se observa en los gráficos de la

figura 2.2, el país con mayor cantidad de repositorios en el mundo es Estados Unidos con 500 repositorios (14,3%), el continente con mayor cantidad de repositorios es Europa con 1604 repositorios (45,9%) -con Sudamérica ocupando el cuarto lugar con un total de 308 repositorios (8.8%)- , y en Sudamérica el país con más cantidad de repositorios registrados es Brasil con 97 repositorios (31,5% en Sudamérica), -con Argentina también en el cuarto lugar con un total de 44 repositorios (8,4% en Sudamérica)-.

En Argentina (Fushimi, 2016), la creación de repositorios en Acceso Abierto fue impulsada con fuerza aunque de forma paulatina desde el año 2008. El Acceso abierto se empezó a difundir ampliamente entre los bibliotecarios universitarios y científicos alrededor de 2003, en confluencia con los primeros años de creación de la Biblioteca Electrónica de Ciencia y Tecnología (BECYT). Desde 2009 comenzó a ser impulsado por el Ministerio de Ciencia y Tecnología (MinCyT), que en 2011 creó el Sistema Nacional de Repositorios Digitales en CyT (SNRD) y elaboró un proyecto de ley que fue aprobado a fines de 2013, transformando el tema en política pública. La Ley 26.899 «Creación de Repositorios Digitales Institucionales de Acceso Abierto, Propios o Compartidos» estableció la obligatoriedad del acceso abierto a la producción financiada con fondos públicos a nivel nacional, a través de repositorios digitales que las instituciones deberán crear, mantener e integrar al SNRD a fin de poder continuar recibiendo financiamiento del estado nacional para el desarrollo de sus proyectos de I+D.

Así como a nivel nacional existen leyes que impulsan el crecimiento de los repositorios institucionales, a nivel regional en Latinoamérica, en materia de acceso abierto, existe el proyecto «LA Referencia» realizado por la *Red Federada Latinoamericana de Repositorios de Documentación Científica*, creada a fines de 2012. Su objetivo principal («LA Referencia», s. f.) es reunir la producción científica en acceso abierto de las instituciones de educación superior de la región, con miras a fortalecer su visibilidad, dando acceso gratis y a texto completo, y ofrecer servicios de valor agregado a los usuarios finales en los países, en la región y en el mundo.

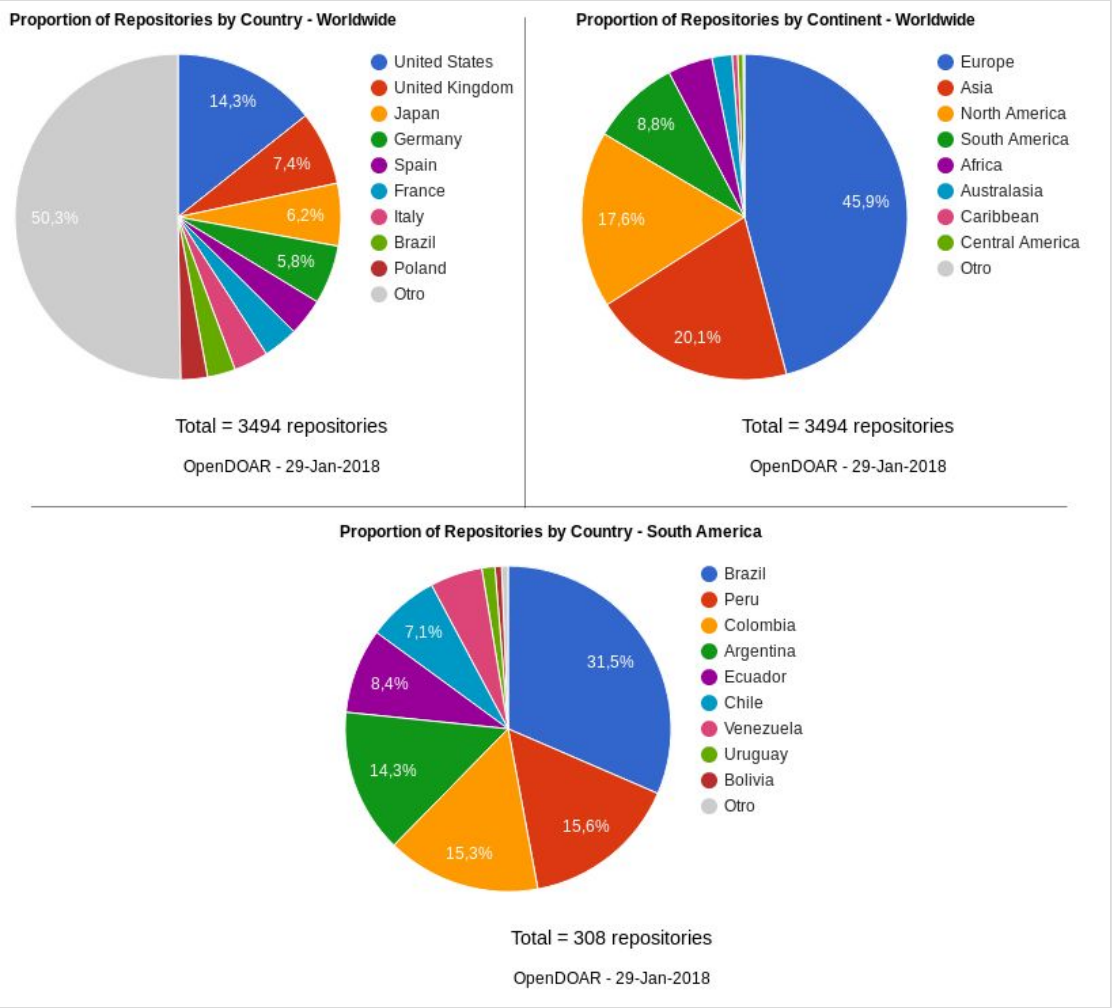


Figura 2.2 - Cantidad de repositorios en el mundo (OpenDOAR), por continente (OpenDOAR) y en Sudamérica (OpenDOAR)

Capítulo 2 | Estadísticas en repositorios digitales

Introducción

Desde la necesidad de crear reportes a partir de un gran cúmulo de datos que diariamente se generan en los repositorios digitales, se define como parte de los objetivos de este trabajo el de integrar el prototipo de herramienta de reportes a la plataforma misma sobre la que está montado el repositorio. Estos reportes de interés están vinculados principalmente a aspectos como el crecimiento del contenido en un repositorio ó el uso de estos contenidos por parte de los usuarios; ejemplos de de estos reportes podrían ser: cantidad de descargas de las publicaciones de un determinado autor, cantidad de descargas por país a un determinado recurso, crecimiento mensual de las publicaciones de una determinada tipología documental, etc. Los repositorios deberían proveer de estos reportes para que sus usuarios, los autores de los contenidos, o los directivos de la Institución vinculada al repositorio puedan basarse en información confiable para la toma de decisiones a distintos niveles.

En diversos lugares del mundo, esta necesidad de analizar los datos alojado en los repositorios o bibliotecas digitales es una cuestión que viene impulsando distintos desarrollos e iniciativas en la materia, entre los que podemos nombrar los estándares *COUNTER* («COUNTER - Code of Practice», 2012) y *SUSHI* (SUSHI Protocol, 2014), además de los distintos servicios basados en estos estándares, tales como *IRUS-UK* («IRUS-UK», s. f.) y *JUSP* («JUSP», s. f.).

A medida que el acceso abierto como modelo de comunicación científica se arraigó en el mundo científico-académico, y fueron surgiendo cada vez más repositorios institucionales, se planteó la necesidad de encontrar criterios que permitan evaluar la evolución histórica del contenido y estado de los repositorios, la usabilidad de sus publicaciones científicas, o el uso de los servicios que ofrecen, entre otras cosas. Los repositorios institucionales son clave para promover la visibilidad de los resultados de la investigación de las financiadas por una institución. Dado que las organizaciones están interesadas en demostrar el valor y el impacto de los repositorios institucionales, la disponibilidad, la fácil recuperación, y la fácil comprensión de las estadísticas en un repositorio es un servicio de un alto valor.

Durante el transcurso de este capítulo se analizará la importancia de las estadísticas en los repositorios institucionales, realizando una introducción a la Web Analytics, además de ahondar en los estándares y servicios estadísticos previamente mencionados.

Ventajas de disponer de estadísticas

Las estadísticas son una herramienta clave a la hora de medir un repositorio en aspectos como su crecimiento, la actividad de sus usuarios y el uso de su contenido, así también lo es la obtención de gráficos que resuman los datos calculados en algo más tangible; los repositorios deben retroalimentarse con estos datos/información y utilizarlos

bajo una *política de expansión* y *mejora continua:* sirven como control de calidad y permiten saber sobre el estado de avance en los repositorios. La recolección sistemática de estadísticas puede ser de utilidad para que los repositorios puedan alcanzar objetivos internos y externos. El análisis del uso interno permite realizar un seguimiento de la producción científica depositada y estudiar así las pautas de crecimiento, lo que ayuda a diseñar el plan de trabajo y las estrategias futuras para alimentar con más contenido.

Como se mencionó previamente, la interpretación de estos datos permitirá la toma de decisiones en varios niveles, no sólo para los miembros que trabajen en el repositorio sino también para la institución a la que el repositorio representa:

- **político / estratégico**: las decisiones estratégicas son las que determinan las metas, los propósitos y la dirección de toda la institución y, por consiguiente, direccionan las metas del repositorio. *Por ejemplo: cooperar con otros grupos que puedan ayudar a fomentar ciertos contenidos institucionales, interactuar más con determinados actores de la institución, etc.*
- **táctico:** éstas decisiones se refieren al desarrollo de tácticas para cumplir las metas estratégicas que definieron los altos niveles administrativos, son más específicas y concretas que las estratégicas y más orientadas a las acciones. Por ejemplo: *cambiar la forma de agrupar cierto tipo de recurso para aumentar su visibilidad, incorporar un nuevo tipo de recurso, implementar una nueva metodología de carga de publicaciones al repositorio, etc.*
- **operativo / tecnológico:** se refieren al curso de las operaciones diarias, estas decisiones determinan cómo se dirigen las operaciones diseñadas para cumplir con las decisiones tácticas. *Por ejemplo: mejorar la estructura de determinada página del repositorio, conseguir más hardware y mejorar la conectividad al repositorio, ampliar el software que soporta al repositorio para integrar cierta tecnología, revisar los índices de la base de datos, etc.*

Como se observó en la introducción del Capítulo 1, a través de las estadísticas se pueden detectar problemas o fenómenos indeseados que ocurren dentro del repositorio y actuar en consecuencia. Es decir, el análisis de las estadísticas nos pueden llevar a acciones concretas para solucionar problemas o realizar mejoras, por ejemplo:

- Incorporar nuevos idiomas, a partir del origen de los usuarios.
- Optimizar las páginas web para maximizar su visibilidad en los buscadores.
- Reorganizar los contenidos para darles mayor relevancia a aquellos menos utilizados.
- Promocionar servicios con bajo nivel de uso.
- Desarrollar servicios, herramientas y estrategias para aumentar el acceso desde ciertos dispositivos.
- Mejorar las herramientas de búsqueda/exploración en el repositorio.

Además de los beneficios a nivel organizacional, también se cuenta con otros beneficios no menos importantes (Bernal & Pemau-Alonso, 2010):

- Reflejan la visibilidad y la difusión internacional y las tendencias de uso de estos archivos abiertos, que son indicadores de su eventual consolidación.

- Pueden ser un medio persuasivo y elocuente para explicar el porqué de los repositorios abiertos ante la institución de la que dependen y su agencia financiadora –mostrando la relación coste-beneficio del repositorio– y ante la comunidad científica cuya investigación difunden y preservan –demostrando su efectividad en potenciar la accesibilidad de los resultados de investigación de un modo gratuito e inmediato en internet–.
- Son de utilidad para los investigadores que están interesados en saber cuánta atención está recibiendo su investigación y cómo los usuarios están accediendo a este material, comparando el grado de «popularidad» de sus trabajos con el de otros compañeros de profesión. De esta forma sirve como estímulo para los autores depositantes y para captar a otros potenciales.

En resumen, los estudios estadísticos son un valor añadido para los administradores de los repositorios y para sus usuarios: miden su popularidad y uso, y contribuyen a la correcta toma de decisiones, a fijar las prioridades, a elaborar mejores políticas, a mejorar y corregir los procedimientos operativos internos, y a controlar la calidad de los datos, entre otras cosas. Externamente a la institución, estos estudios pueden ayudar al desarrollo de nuevos marcos de evaluación científica de la producción de los investigadores.

Qué miden las estadísticas

Al hablar de estadísticas en un repositorio institucional se hace referencia a reportes de diversa índole: reportes sobre las características, tamaño y tasa de crecimiento de las publicaciones alojadas en un repositorio, reportes sobre el estado de las entidades administrativas de un repositorio (permisos, usuarios, metadatos, etc.); además, una de las estadísticas que en la actualidad están en consideración en diversos repositorios del mundo son las *estadísticas de uso* sobre el contenido por parte de los internautas (por ejemplo, visitas o descargas de publicaciones). Cabe mencionar que también existen sistemas de medición desde el punto de vista de los autores (citaciones y factor de impacto de las revistas académicas), tales como el *Journal Impact Factor* (JIF) o el *h-index;* sin embargo, estos sistemas no serán analizados en el presente trabajo.

Cuando se hace referencia a las características, tamaño y tasa de crecimiento en un repositorio, se hace alusión a aspectos como la cantidad de recursos existentes en el repositorio, y al crecimiento cuantitativo y temporal de éstos -detectando posibles mesetas en la curva de crecimiento, tendencias y períodos de mayor o menor actividad de depósito en el repositorio-. El concepto de «tamaño» puede ser bastante amplio y referirse, por ejemplo, a la cantidad de recursos a texto completo, cantidad de usuarios registrados en el sistema, cantidad de recursos sin licencia de uso, etc. Asimismo, el término «crecimiento» también puede interpretarse de diversas maneras, a saber: recursos incorporados anualmente, usuarios registrados semanalmente, recursos de determinada tipología agregados en determinado periodo de tiempo, recursos a partir del origen (dependencia institucional, departamento, área, etc), recursos por temática (informática, ingeniería, etc), entre otras.

Por otra parte, las estadísticas de uso transmiten una de las informaciones más buscadas por los administradores de un repositorio y los miembros de la institución, ya que

indican directamente la actividad y el uso que un usuario del repositorio hace del sistema en sí y de la producción académica de la institución. Entre las estadísticas de uso que más comúnmente interesa medir son:

- cantidad de veces que una página de un servicio fue visitado,
- cantidad de veces que una página de un ítem/recurso fue visitado,
- cantidad de veces que una ítem/recurso fue descargado (con o sin éxito),
- términos de búsqueda mayormente usados en el repositorio,

Existen muchas más cuestiones potenciales a medir sobre el comportamiento del usuario en el sitio, y las ya mencionadas forman parte de una área de análisis mayor llamado «Análisis Web» o «Web Analytics».

Web Analytics

El análisis de la actividad en un sitio web o *web analytics* es («Web analytics», 2018) la medición, recopilación, análisis y generación de informes de datos generados en torno al uso de un sitio web con el fin de comprender y optimizar el servicio provisto; es un conjunto de técnicas relacionadas con el análisis de datos relativos al tráfico en un sitio web con el objetivo de entender su tráfico como punto de partida para optimizar diversos aspectos del mismo. El análisis web no es solo un proceso para medir el tráfico web, sino que puede utilizarse como una herramienta para la investigación comercial y de mercado, y para evaluar y mejorar la efectividad de un sitio web.

Como se mencionó previamente, muchos de las estadísticas de uso que es de interés medir en los repositorios institucionales son un subconjunto de las mediciones definidas por la web analytics. No existen definiciones aprobadas globalmente acerca de la analítica web, aunque existe un organismo de normalización creado para tal fin llamado WAA (*Web Analytics Association*), un grupo de profesionales que tratan de poner todo el conocimiento encima de la mesa para alcanzar un conjunto estándar de directrices para analítica Web y los indicadores asociados con la analítica. En 2006, WAA formó un comité para desarrollar algunas definiciones estándar de web analytics que ayudarían a los usuarios a entender cómo se obtienen los indicadores. Entre los principales indicadores definidos por este comité podemos mencionar:

Hits	Una página web está formada por muchos elementos, por ejemplo imágenes, sonidos, hoja de estilos, logos, archivos javascript, banners, etc. Estos elementos son denominados *hits*. Es decir, cada petición que se hace a un servidor solicitando un archivo es un *hit*.
Page View	Las páginas vistas son las diferentes páginas a las que se accede dentro de una misma web. Las páginas están compuestas de hits, así que la vista de una sola página puede generar múltiples hits así como todos los recursos requeridos para ver la página (imágenes, archivos .js y .css).

Page View Duration	Es el tiempo en que en una sola página (ya sea Blog o anuncio publicitario) es visitada y visualizada por el usuario.
Click	Se refiere a una única instancia de un usuario cuando cliquea un hipervínculo que le lleva de una página a otra.
Click Path	La secuencia de hiperenlaces que uno o más visitantes del sitio web siguen en un sitio dado.
Downloads	Son hits cuyo tipo contenido no es HTML, sino archivos PDF, CSV, DOC, MP3, etc. Por ejemplo, cuando se descarga el PDF asociado a una publicación de un repositorio.
Session	Una visita o sesión se define como una serie de solicitudes de página o solicitudes de imágenes del mismo cliente identificado de manera única. Un cliente único se identifica comúnmente por una dirección IP o una ID única que se coloca en la cookie del navegador. Una visita se considera finalizada cuando no se registraron solicitudes en algunos minutos transcurridos (30 minutos - «*time out*»).
Session Duration	Promedio de tiempo que los visitantes pasan en el sitio cada vez que lo visitan.
Impression	Es la cantidad de veces que un mismo anuncio, un banner, una página o cualquier otro contenido analizado es cargado en la pantalla de un usuario. Cada vez que el contenido es visto cuenta como una impresión.
Otras definiciones	Unique Page View, First Session, Unique Visitor, New Visitor, Repeat Visitor, Exit Page, Landing Page, Impression, Bounce Rate, Exit Rate, Direct Traffic, Referred Traffic, Search Traffic, Session per Unique.

Herramientas de la análisis Web

La *web analytics* utiliza distintas herramientas encargadas de capturar y procesar la información de uso de un sitio web, y proveen información sobre el comportamiento de los usuarios en el sitio: el sitio del que proceden, qué hacen en el sitio, por qué páginas navegan, durante cuánto tiempo, cuántas veces visitan el sitio, de que país son, qué tipo de conexión de internet tienen, en qué punto abandonan el sitio, en qué paso de un proceso de alta desisten, etc. Entre estas herramientas están:

- Analizadores de ficheros de logs: es un tipo de software de la web analytics que analiza los archivos de *logs*[2] alojados en un servidor web y, en función de los valores contenidos en los archivos de logs, deriva indicadores sobre «cuándo», «cómo» y «quién» visita un servidor web. Los indicadores más comunes que permite obtener son: número de

[2] Un archivo de log se refiere a un fichero o base de datos que mantiene un registro secuencial de todos los acontecimientos (eventos o acciones) que afectan a un proceso particular (aplicación, actividad de una red informática, etc.).

visitas y duración de las mismas, usuarios autenticados y última autenticación, dominio/países de los usuarios visitantes, lista de hosts que se conectan al servidor web, páginas más visualizadas, páginas de entrada y de salida, tipos de archivos solicitados, sistemas operativos y browsers usados, accesos de spiders/bots, listado de HTTP referrers, etc. Estos programas deberían admitir la mayoría de los principales formatos de archivos de log de los servidores web, entre ellos: *Apache* (formato de registro NCSA combinado / XLF / ELF o formato de registro común (CLF)), *WebStar*, *IIS* (formato de registro W3C), y muchos otros formatos de registro de servidor web comunes.

Ejemplos: *AWStats* (http://www.awstats.org/) es un software *open-source* que realiza análisis mediante archivos de logs en distintos formatos de logs para servidores web.

- Etiquetado de páginas/Javascript tracking: este método se basa en la incorporación de un script a cada una de las páginas de un sitio. Cada vez que una página es visitada, este script se comunica con una base de datos a la que comunica la impresión de la página junto con, potencialmente, datos adicionales procedentes de las cookies. Estos scripts “etiquetan” a cada usuario que visita un sitio web utilizando una cookie (pequeña pieza de datos sobre eventos DOM HTML de exploración y información con estado, así como otros datos arbitrarios), y luego la envía a un servidor central (propio o tercerizado) para registrar los eventos de uso por parte del usuario.

Ejemplos: *GoogleAnalytics* (https://developers.google.com/analytics/) es una de las herramientas de etiquetado de páginas más utilizadas hoy en día; también existen herramientas *open-source* muy potentes como es el caso de *PIWIK* (https://piwik.org/).

- Sistemas híbridos: Algunas empresas producen soluciones que recopilan datos a través de archivos de logs y etiquetado de páginas, y pueden analizar ambos tipos. Al usar un método híbrido, su objetivo es producir estadísticas más precisas que cualquiera de los métodos por sí mismo.

Ejemplos: *Logaholic* (http://www.logaholic.com/) es una herramienta que soporta ambos métodos de análisis, etiquetado de páginas y análisis de archivos de log.

- **Geolocalización de los visitantes**: Con la geolocalización IP, es posible rastrear la ubicación de los visitantes. Utilizando una base de datos de geolocalización de IP o una API de a un servicio específico, los visitantes pueden ser geolocalizados a nivel de ciudad, región o país. Utilizando la geolocalización en conjunción a otros datos de uso de los usuarios, se pueden determinar comportamientos de usuarios por región, detectar preferencias por región, detectar servicios más utilizados por región, entre otras cosas.

- **Click analytics:** Es un tipo especial de análisis web que presta especial atención a los clics. Un editor de un sitio web utiliza análisis de clics para determinar el rendimiento de su sitio en particular, con respecto a dónde estan haciendo clic los usuarios del sitio.

Ejemplos: *Mouseflow* (https://mouseflow.com/) es una herramienta que rastrea los clics, el movimiento del mouse, los desplazamientos en las páginas, la actividad en los formularios y muchos más. Muestra una grabación de la actividad de cada visitante en un sitio.

- **Packet sniffing:** son herramientas de software que se captura y analiza todos los paquetes del tráfico de red entre el servidor de un sitio y el mundo exterior. La principal ventaja de packet sniffing en cuanto a la recolección de datos es el hecho de que toda la información es capturada, se haya generado o no un *page view*, se haya completado la descarga del contenido o no.

Más allá de la existencia de estas herramientas que permiten analizar y capturar eventos de uso por parte de un usuario, o analizar la actividad de los logs de acceso del servidor web, puede haber casos en el que el software que da soporte a un repositorio almacene o indexe ciertos eventos de uso de forma local y personalizada en bases de datos o indexadores propios. Ésto sucede en el caso del software de repositorios digitales «DSpace», y se explicará en mayor detalle en los capítulos subsecuentes.

Iniciativas sobre estándares y servicios estadísticos en el mundo

Con el advenimiento explosivo del Acceso Abierto desde el año 2003 -luego de las declaraciones de Budapest (2002), Bethesda (2003) y Berlín (2003)-, la implementación creciente de repositorios institucionales en abierto en Europa y el mundo, el surgimiento de las métricas basadas en el uso como una herramienta alternativa para ayudar a evaluar el impacto y el valor de las publicaciones financiadas con fondos públicos, y el incremento en la variedad y cantidad de datos registrados como estadísticos de uso en las diversas plataformas de repositorios digitales, se ha generado un intenso debate en los últimos años: por un lado, para llegar a un consenso sobre qué datos recolectar, y por otro para definir cómo realizar los estudios y cómo armonizarlos con otros sistemas de recolección de datos (principalmente los de los editores tradicionales), con el fin de obtener análisis agregados.

El debate mencionado en el párrafo anterior ha movilizado esfuerzos de tal forma de establecer nuevos estándares y protocolos de alcance internacional que provean una verdadera interoperabilidad en cuanto a las estadísticas de uso, para lo que se necesita una uniformización de descriptores de objetos además de pautas comunes en la captura y transferencia de datos. En los últimos años diversas iniciativas han explorado posibles estándares de alcance internacional para facilitar la creación e intercambio de estadísticas más complejas, y la integración y la interoperabilidad con los datos estadísticos generados por los repositorios, así como también se crearon diversos servicios que funcionan sobre estos estándares: Pirus, COUNTER, SUSHI, Mesur, OAS, IRUS-UK, JUSP, entre otros.

Estándares

COUNTER

El proyecto COUNTER (o en inglés ***C**ounting **O**nline **U**sage of **N**etworked **E**lectronic **R**esources*), lanzado en 2002, es una iniciativa internacional para mejorar la confiabilidad de las estadísticas de uso en línea. El objetivo de COUNTER es garantizar que los informes de uso en línea de los proveedores sean creíbles, compatibles y consistentes. Es una organización sin fines de lucro respaldada por una comunidad global de miembros de

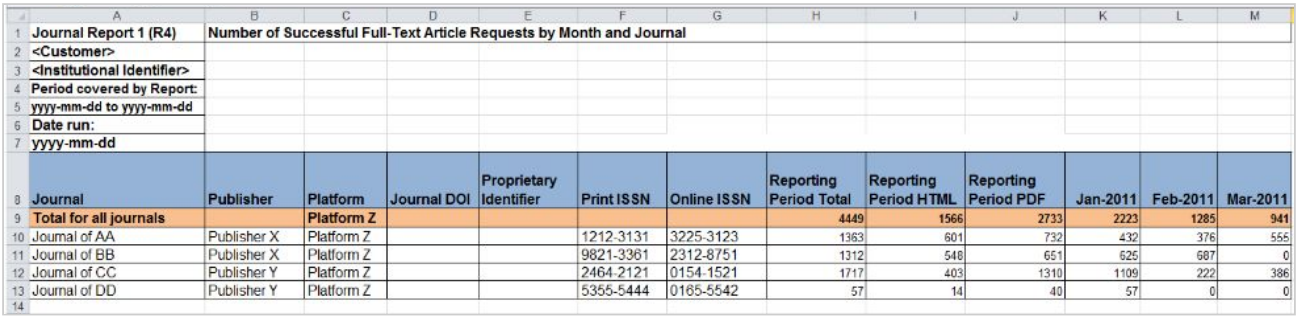
bibliotecas, editores y *vendors*[3], que contribuyen al desarrollo del «Código de Práctica» (*Code of Practice*) («COUNTER - Code of Practice», 2012) a través de distintos grupos de trabajo. Además, COUNTER mantiene los «*Registros de Cumplimiento*» (https://www.projectcounter.org/about/register/) que enumeran los editores y proveedores que han pasado una auditoría independiente de sus estadísticas de uso; este registro también incluye detalles de la implementación de servidor SUSHI del proveedor de contenido.

El *Código de Práctica* es un documento que describe y estandariza los reportes de estadísticas de uso para diferentes tipos de recursos en línea, proporcionando orientación sobre los elementos de datos a medir, las definiciones de estos elementos de datos, y el contenido y el formato de los reportes. Permite a los editores y proveedores informar el uso de sus recursos electrónicos de una manera consistente, habilitando a las bibliotecas comparar datos recibidos de diferentes editores y proveedores; los publicadores de estadísticas de uso según COUNTER deberían exponer al menos las estadísticas de los últimos 24 meses.

COUNTER fue lanzado oficialmente en marzo del 2002, con el primer lanzamiento del Código de Práctica para reportes estadísticos sobre el uso en revistas y bases de datos en enero del 2003. Desde entonces el Código de Práctica fue evolucionando, llegando a su versión 4 en el año 2012, ampliando entre otras cosas la cantidad de recursos a ser medidos (revistas, bases de datos, libros, trabajos de referencia y bases de datos multimedia). El código es abierto en su definición, es decir, que evoluciona constantemente a través de sucesivos *releases* mediante la realización de extensiones, modificaciones y mejoras, en respuesta a las demandas de las comunidades internacionales de bibliotecarios y publicadores.

El Código de Práctica COUNTER en su versión 4 define 23 tipos de reportes (algunos obligatorios y otros opcionales) abarcando 5 tipos de recursos: revistas, bases de datos, libros, trabajos de referencia y bases de datos multimedia. Los reportes deben poder ser exportables a un formato Microsoft Excel (como se observa en la Figura 2.3), TSV, o cualquier otro formato posible de importarse desde Microsoft Excel, así como también debería existir la alternativa de generar un reporte XML-COUNTER compliant (un formato XML definido por el estándar SUSHI para usar con COUNTER). Además, en el apéndice E del estándar, COUNTER define los estándares de *auditoría* que serán utilizados para evaluar periódica y automáticamente mediante scripts, a aquellos editores y publicadores que están registrados en el Registro de Cumplimiento, con el fin de garantizar que estas organizaciones cumplan continuamente con el estándar y generen reportes estadísticos confiables.

[3] **Vendor:** Un editor u otro proveedor de información en línea que entrega contenido licenciado al cliente y con quien el cliente tiene una relación contractual.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Journal Report 1 (R4)	Number of Successful Full-Text Article Requests by Month and Journal											
2	<Customer>												
3	<Institutional Identifier>												
4	Period covered by Report:												
5	yyyy-mm-dd to yyyy-mm-dd												
6	Date run:												
7	yyyy-mm-dd												
8	Journal	Publisher	Platform	Journal DOI	Proprietary Identifier	Print ISSN	Online ISSN	Reporting Period Total	Reporting Period HTML	Reporting Period PDF	Jan-2011	Feb-2011	Mar-2011
9	Total for all journals		Platform Z					4449	1566	2733	2223	1285	941
10	Journal of AA	Publisher X	Platform Z			1212-3131	3225-3123	1363	601	732	432	376	555
11	Journal of BB	Publisher X	Platform Z			9821-3361	2312-8751	1312	548	651	625	687	0
12	Journal of CC	Publisher Y	Platform Z			2464-2121	0154-1521	1717	403	1310	1109	222	386
13	Journal of DD	Publisher Y	Platform Z			5355-5444	0165-5542	57	14	40	57	0	0
14													

Figura 2.3 - Ejemplo de reporte COUNTERv4 - *Journal Report 1: Number of Successful Full-Text Article Requests by Month and Journal*

En cuanto a la forma de recolección de datos de uso de los recursos, COUNTER no define ni recomienda una manera única y específica para hacerlo, sin embargo, introduce a los lectores/desarrolladores del protocolo en dos de las formas mencionadas anteriormente, los *analizadores de ficheros de logs* y el *etiquetado de páginas.* A su vez, define protocolos o recomendaciones para la limpieza de los datos recolectados para así no caer bajo la influencia negativa de los efectos de las búsquedas federadas y los robots/*crawlers* de Internet en las estadísticas de uso.

En su versión 4, COUNTER incluye como anexo un código de práctica suplementario (el *Code of Practice for Articles*[4]) proveyendo especificaciones para el registro y la generación de reportes de uso a el nivel de *artículo* individual, que se basan y son consistentes con el Código de Práctica general de COUNTER para recursos electrónicos. Las principales razones de la existencia de este anexo fueron, primero, el creciente aumento de este tipo de recursos en los distintos repositorios institucionales del mundo, luego la creación de los reportes COUNTER basados en XML y del protocolo SUSHI, y por último, los resultados salientes del desarrollo llamado *PIRUS*, que se explicará en mayor detalle en las secciones posteriores.

SUSHI

A medida que la cantidad de publicadores y proveedores de Europa que exponían datos de uso mediante COUNTER, la recolección y cargar los mismos en los sistemas clientes requería de mucho esfuerzo, por lo que en el año 2005 comenzó la iniciativa conocida como SUSHI (***S**tandardized **U**sage **S**tatistics **H**arvesting **I**nitiative* o *Iniciativa de cosecha de estadísticas de uso estandarizada*) para definir una forma automatizada de intercambio.

SUSHI es un estándar ANSI/NISO (SUSHI Protocol, 2014) que define un modelo automatizado de solicitud y respuesta para recolectar datos de uso de recursos electrónicos (*e-resources*). Está diseñado para funcionar con los informes COUNTER, los informes de uso recuperados con más frecuencia. Entre los beneficios de su utilización podemos mencionar:

- Sustituye a la recopilación de informes de datos de uso mediada por el usuario que consume tiempo.

[4] https://www.projectcounter.org/wp-content/uploads/2016/11/counterart_cop_October2015.pdf

- El protocolo es generalizado y extensible, lo que significa que se puede usar para recuperar una variedad de informes de uso, no sólo COUNTER.

El protocolo SUSHI está diseñado para proporcionar un método automatizado para recuperar informes de estadísticas de uso estandarizados utilizando un contenedor XML procesable por máquina. El protocolo utiliza los servicios web *Simple Object Access Protocol* (SOAP). En la Figura 2.4 se observa a grandes rasgos el funcionamiento del protocolo SUSHI:

1. Un sistema ERM[5] inicia una solicitud mediante un *cliente SUSHI* para la obtención de un reporte COUNTER a un *servidor que implementa SUSHI*. El cliente crea un mensaje SOAP, conteniendo en su carga los distintos parámetros que componen la solicitud: ID del cliente, información de login en el servidor SUSHI, nombre del reporte a solicitar, rango de meses que interesa incluir en el reporte, etc.
2. Luego, el servidor decodifica el mensaje SOAP y procesa la solicitud.
3. Algunos servidores pueden tener estadísticas pre-empaquetadas, y si el reporte estadístico solicitado está en este estado, entonces se lo retorna en el formato COUNTER. En caso contrario, el servidor inicia un proceso que busca entre los datos del proveedor de datos estadísticos con el fin de generar el reporte solicitado.
4. Una vez generado el reporte, se lo retorna al cliente SUSHI mediante una respuesta SOAP, la cual será extraída por el cliente y enviada al sistema ERM para su posterior procesamiento.

Como se mencionó anteriormente, SUSHI es un marco de referencia sobre el que funcionan principalmente 3 formatos XML distintos y posibilitan la interoperabilidad: SUSHI XML Schema, SUSHI WSDL y los reportes COUNTER-XML. SUSHI XML Schema y SUSHI WSDL representan el *contrato de datos* y el *contrato de servicio* entre el cliente y el servidor que operan en un ambiente business-to-business, y el informe COUNTER XML es la carga real o el *payload* de la transacción.

SUSHI Schema es una definición de esquema XML utilizado para realizar la operación SUSHI. En este esquema se definen dos patrones de mensajes, el *ReportRequest* y el *ReportResponse*, que facilitan la interoperabilidad entre un cliente y servidor SUSHI, mediante el intercambio de datos al interactuar con las operaciones de servicio.

[5] La administración electrónica de recursos o ERM (*Electronic Resource Management*) es la práctica y las técnicas utilizadas por los bibliotecarios y el personal de una biblioteca para rastrear la selección, adquisición, licenciamiento, acceso, mantenimiento, uso, evaluación, retención y eliminación de los recursos de información electrónica de una biblioteca. Estos recursos incluyen, entre otros, publicaciones electrónicas, libros electrónicos, streaming media, bases de datos, datasets, CD-ROMs y software. Para agilizar las prácticas ERM por parte de los bibliotecarios, fueron creados distintas herramientas de software. Los **sistemas ERM** (ERMS) son programas de software diseñados para ayudar a los bibliotecarios con la adquisición y administración de recursos electrónicos, entre otras cosas. Proporcionan herramientas para ayudar a administrar el proceso de licencia y adquisición, y para proporcionar acceso a materiales.

- El *ReportRequest* define: la identificación de la organización que está solicitando el reporte de uso para un cliente, el nombre del cliente final que está solicitando el reporte, y la definición del reporte solicitado como se identifica en COUNTER y la versión (p.e. **JR1,** versión 4 COUNTER).
- El *ReportResponse* tiene como carga principal la respuesta generada por el servidor SUSHI; mientras que el esquema SUSHI se escribió como un protocolo generalizado para admitir una variedad de posibles informes de uso, la intención principal de los desarrolladores fue apoyar los informes COUNTER. Por lo tanto, se desarrolló una extensión de esquema COUNTER-SUSHI para su uso con el protocolo general de SUSHI.

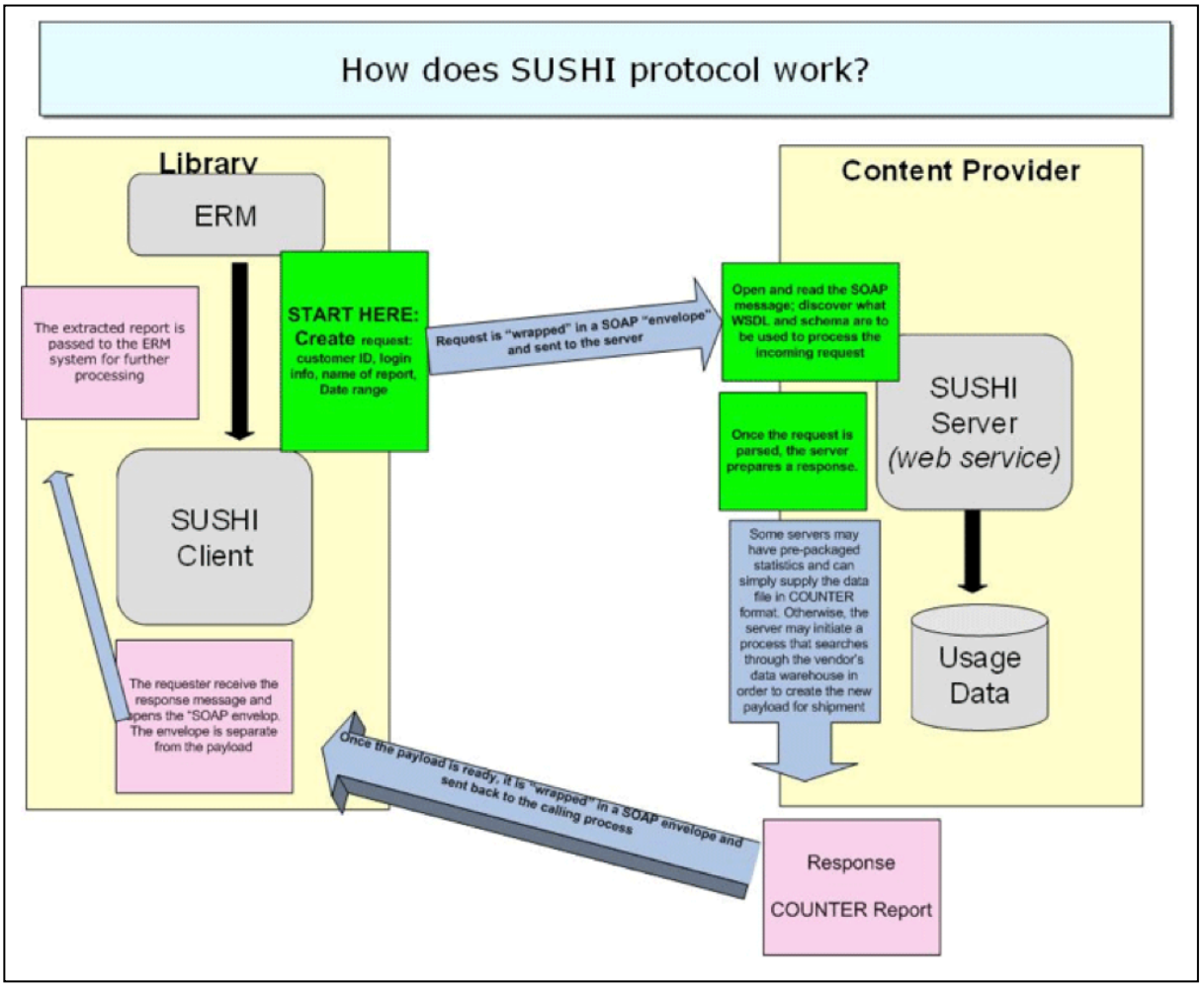


Figura 2.4 - Funcionamiento de protocolo SUSHI

El **SUSHI WSDL**[6] representa un contrato de servicio y describe, con un alto nivel de abstracción, como los lados cliente y servidor de la transacción de servicios web

[6] WSDL (**W**eb **S**ervice **D**escription **L**anguage) es un formato XML utilizado para describir completamente un servicio web. El WSDL define las operaciones o funciones que proporciona el servicio, los formatos y parámetros utilizados para acceder a ellas, el mecanismo de transporte (por ejemplo, *SOAP*) utilizado para enviar y recibir mensajes de entrada y salida, y las direcciones web donde se ubica el servicio .

interoperan. Define una operación específica, *GetReport,* y espera como parámetro los mensajes *ReportRequest* y retorna como parámetro un *ReportResponse.*

SUSHI-Lite

A partir del año 2014, un grupo de trabajo fue formado por NISO para trabajar sobre una versión más «liviana» de SUSHI basada en las tecnologías REST para el intercambio de datos (en lugar de SOAP) y JSON para la representación de los informes COUNTER (en lugar de XML). La utilización de tecnologías JSON+REST está en auge hoy en día debido a que es una de las formas más fáciles de exponer servicios: REST utiliza HTTP y operaciones CRUD básicas, además de que es relativamente simple de entender y documentar. Durante el verano del 2015 se lanzó una versión preliminar o *draft* para esta versión de SUSHI, la que recibió el nombre de SUSHI-Lite; en el momento de redactar este trabajo aún no existen mayores avances.

Servicios basados en COUNTER

Los bibliotecarios a lo largo del mundo (Estelle & Lambert, 2015) usan las estadísticas de uso COUNTER para realizar su toma de decisiones, pero aunque la utilidad de los informes es tan amplia, su recopilación puede ser tediosa y llevar mucho tiempo. Sin embargo, la combinación del Código de Práctica COUNTER con el protocolo de la Iniciativa SUSHI cambia las reglas del juego. Los dos estándares, si se implementan correctamente, pueden crear una infraestructura en la cual los consorcios de bibliotecas y otros terceros pueden construir servicios nacionales y regionales. El Journals Usage Statistic Portal (JUSP) y el Institutional Repository Usage Statistics (IRUS) en el Reino Unido (UK) son ejemplos de dos servicios de confianza que ahorran a bibliotecarios y administradores de repositorios institucionales en el Reino Unido mucho tiempo al eliminar el esfuerzo manual duplicado.

JUSP

El creciente aumento en la presión sobre los presupuestos de las bibliotecas en Europa (Estelle & Lambert, 2015), significa que es fundamental demostrar el retorno de la inversión sobre recursos electrónicos costosos. **JUSP** (*Journal Usage Statistics Portal*), una herramienta actualmente utilizado por las bibliotecas en el Reino Unido y Suecia, tiene como objetivo hacer que este proceso sea más rápido, más fácil y más efectivo. Jisc, una organización que ofrece soluciones digitales para la educación e investigación del Reino Unido, financia a JUSP para ofrecer un servicio gratuito en el punto de uso de las bibliotecas académicas. Evitando la necesidad de visitar varios sitios web de editores para recopilar estadísticas de uso, JUSP permite que las bibliotecas comparen, rápida y fácilmente a través de un único punto de acceso, el uso de los recursos electrónicos en un rango de editores y años; ésto permite dedicar menos tiempo a la recolección de datos y más tiempo en el análisis de datos para respaldar los procesos de toma de decisiones.

Los datos estandarizados que cumplen con la norma COUNTER son cosechados por JUSP en nombre de las bibliotecas utilizando el protocolo SUSHI, lo que resulta en un aumento de la eficiencia. El servicio proporciona acceso a los datos de uso de alrededor de 80 editoriales e intermediarios, aunque este número está en constante crecimiento. Los

informes COUNTER actualmente cosechados son los informes de revistas JR1, JR1a, JR1 GOA, e informes de libros BR1, BR2 y BR3.

Los informes se pueden generar con solo hacer clic en un botón, lo que permite que las personas respondan a las solicitudes de manera rápida y sencilla. Los datos se pueden ver dentro de la interfaz web de JUSP o se pueden exportar para su uso dentro de los propios sistemas de la biblioteca para un mayor análisis, por ejemplo, mediante la adición de datos de costos o fondos.

El servicio proporciona datos de uso precisos y comparables para *respaldar* la **evaluación de los recursos electrónicos**. Además de ver y descargar informes de uso estándar, las bibliotecas pueden acceder a una gama de informes de valor agregado para ayudar a analizar el uso, considerar las tendencias en el tiempo y establecer el valor del dinero para ayudar en las decisiones de compra. El servicio también incluye datos de gateways y hosts, que ayudan a proporcionar una imagen más real del uso cuando se ven junto con los datos directos de los editores.

Las bibliotecas valoran la función de JUSP en términos de proporcionar contenido preciso, confiable y estandarizado (Estelle & Lambert, 2015). Una serie de verificaciones de validación humana y máquina realizadas por el equipo de JUSP, aseguran que los errores se reduzcan significativamente. Sin embargo, la adición de numerosas personas que usan el servicio significa que los datos son examinados constantemente y que los problemas detectados por una sola persona pueden generar beneficios para todos.

IRUS-UK

IRUS-UK (*Institutional Repository Usage Statistics - United Kingdom*) es un servicio de agregación de estadísticas para repositorios en el Reino Unido (Estelle & Lambert, 2015), que les permite compartir y comparar estadísticas de uso utilizando el estándar COUNTER. El servicio recopila y luego procesa los datos de uso de los repositorios y los consolida en estadísticas que cumplen con COUNTER siguiendo las reglas del *Código de prácticas de COUNTER*. Entre sus objetivos se encuentran brindar una visión nacional del uso de los repositorios en el Reino Unido para beneficiar a organizaciones como Jisc, ofrecer oportunidades para la evaluación comparativa, y actuar como intermediario entre los repositorios del Reino Unido y otras agencias. Este servicio se basa en un proyecto antecesor llamado *PIRUS2* (definido en la siguiente sección) cuyo objetivo era crear, registrar y consolidar estadísticas de uso para artículos individuales alojados en repositorios digitales.

IRUS-UK recopila datos de uso sin procesar de los repositorios participantes en todo el Reino Unido y los procesa en estadísticas que satisfacen COUNTER. Esto proporciona a los repositorios de datos comparables, autoritativos y basados en estándares para la creación de perfiles y la evaluación comparativa (benchmarking). IRUS-UK permite generar informes a nivel de repositorio (por ejemplo, cifras totales de descarga en el repositorio) como a nivel de artículo, y define una taxonomía de 25 tipos de artículos distintos, entre ellos: Art/Design Item, Article, Audio, Book, Book Section, Conference Papers / Posters, etc. El software sobre los que están montados los repositorios que participan del servicio

(irus.mimas.ac.uk/about/participants/) en su mayoría son Eprints, siguiéndole DSpace y Fedora en menor cantidad.

El servicio utiliza un robusto proceso de ingesta de múltiples etapas (MacIntyre & Jones, 2016) que incluyen: validación de datos, eliminación de robots y accesos inusuales, filtrado de doble clics, y finalmente la transformación de los datos de uso sin procesar en estadísticas que cumplen con COUNTER. Para la ingesta de datos de uso, el servicio funciona agregando al software de cada repositorio una pequeña porción de código que emplea el «Protocolo del rastreador IRUS» (tracker protocol); actualmente, el código que implementa el rastreador está disponible para las plataformas DSpace, EPrints y las pautas de implementación están disponibles para Fedora. Este rastreador reúne datos básicos en bruto para cada descarga y luego los envía al servidor IRUS-UK. Diariamente, los datos pasan por una serie de filtros y verificaciones antes de ser agregados al portal, y se revisan nuevamente al final de cada mes. Estos procesos ayudan a asegurar que la actividad de robots malintencionados u otra actividad inusual sea detectada y eliminada de la ingesta de tal manera de mejorar la precisión de los datos calculados.

Una vez que se han verificado los datos, se puede acceder a ellos a través de la interfaz del usuario web en el portal IRUS-UK. Dentro del portal, hay una amplia gama de vistas que muestran los datos en diferentes formas de informes, y estando disponible su descarga en varios formatos: CSV, TSV, y HTML.

Otros proyectos internacionales

Existen varios otros proyectos o iniciativas en el mundo, algunos de los cuales han sido abandonadas con el tiempo pero otras aún siguen vigentes. En los párrafos sucesivos se mencionan algunos de ellos.

En el Reino Unido, el proyecto **PIRUS** (Publisher and institutional repository statistics), un proyecto desarrollado por JISC que finalizó en el año 2009, ha demostrado que es técnicamente posible crear, registrar y consolidar estadísticas de uso para artículos individuales usando datos de repositorios y editores, abogando para ello por el enriquecimiento de las estadísticas de COUNTER. Por aquellos tiempos estaba vigente la versión 3 de COUNTER y no se contaba con una estandarización de estadísticas a nivel de artículos, por lo que este proyecto resultó como una respuesta a las necesidades vigentes de la comunidad en ese entonces (Brody et al., 2009). El proyecto recomendó OpenURL *Context Object* como el componente esencial para que los repositorios pudiesen describir y transmitir sus estadísticas de uso, incluyendo los siguientes datos: tiempo y fecha de acceso, DOI, versión y formato de artículo (HTML o PDF), IP y user-agent del cliente, entre otros.

Luego del PIRUS vino su sucesor **PIRUS2**, que promueve la estrecha colaboración entre COUNTER, CrossRef, editores, repositorios, NISO e iniciativas similares en Europa, para establecer una infraestructura y llevar a cabo una serie de programas de software en acceso abierto que promuevan la generación y el intercambio de estadísticas de uso de tipo COUNTER. Los objetivos principales de este proyecto son: (1) Desarrollar un conjunto de programas gratuitos y de código abierto que respalden la generación y el intercambio de

datos y estadísticas de uso (COUNTER-compliant) sobre los elementos individuales en los repositorios, (2) desarrollar un prototipo de servicio de estadísticas de publicadores/repositorios a nivel de artículo, y (3) definir un conjunto básico de informes de estadísticas de uso estándar que los repositorios deben producir para el consumo interno y externo. Posteriormente, PIRUS2 fue tomado como base para desarrollar el *Código de Prácticas COUNTER para Artículos*.

Capítulo 3 | DSpace

Introducción

Como fue mencionado en los capítulos precedentes, en la actualidad existe un importante crecimiento en cuanto a la creación de repositorios institucionales en Argentina y el mundo -gracias a las diferentes hitos mundiales que fueron sucediendo en relación al Acceso Abierto, las leyes e iniciativas nacionales y regionales relativas a la materia, y el crecimiento y mejora de las herramientas de software que dan soporte a los repositorios digitales-, y la gestión de estos repositorios presenta diferentes *problemáticas* a tener en cuenta al momento de elegir el software que les sirva de soporte:

- Gestión de metadatos
- Gestión de políticas de contenido
- Visibilidad de los contenidos
- Búsqueda de los contenidos
- Gestión de la preservación digital
- Generación de estadísticas de uso y contenidos en el repositorio
- Gestión de usuarios y perfiles de autor
- Interoperabilidad de contenidos

Existen varias herramientas de software que buscan solución a las problemáticas anteriores; **DSpace** (dspace.org, 2018) es un software de código abierto que proporciona distintas herramientas y funcionalidades para satisfacer las necesidades que surgen a partir de estas problemáticas. Como se observa en la figura Figura 3.1, a comienzos del 2018, cerca de la mitad de los repositorios en el mundo (*43,9%*, es decir, *1535* repositorios según OpenDOAR) utilizan DSpace como software de soporte para su implementación, mientras que otras alternativas de software son utilizadas en un porcentaje menor (EPrints con el *13,4%,* Digital Commons con el *4,7%*, entre otros). En nuestro país, hay una coincidencia con la tendencia mundial, debido a que la mayoría de los repositorios utiliza DSpace (*42,3%*, es decir, *19* repositorios) en relación a otros alternativas de software. Esta predominancia en la utilización de DSpace se puede deber a varios factores («Top Reasons to Use DSpace», s. f.):

- Es un software open source con disponibilidad gratuita para cualquier persona, tanto de su instalador y de su código fuente. El código utiliza la licencia BSD, la cual permite a cualquier institución usarlo, corregirlo y agregar sus propias modificaciones/mejoras.
- Posee una gran comunidad de usuarios y desarrolladores ubicados en todo el mundo, utilizando distintas herramientas que facilitan la cooperación entre sus miembros, por ejemplo: versionadores de código como *git*, sistemas de reportes de errores y mejoras como *JIRA Software*, foros de consulta para usuarios y técnicos, entre otros.
- Es completamente adaptable a las necesidades de cada institución, con la posibilidad de personalizar la forma en la que se verá el repositorio, el formato de

metadato a utilizar, los campos de búsqueda sobre los que se permitirá realizar búsquedas, el mecanismo de autenticación a utilizar, los idiomas soportados por el repositorio, entre otros.

- Es un software multiplataforma, principalmente desarrollado en JAVA, que permite su instalación en entornos Linux, Mac OSX o Windows.
- Permite administrar y preservar varios tipos de contenidos digitales. Puede reconocer y procesar distintos tipos de formatos de archivos y mimetypes, contando con un registro extensible de formatos soportados por la aplicación.
- Soporta gran variedad de estándares: OAI-PMH, OAI-ORE, SWORD, WebDAV, OpenSearch, OpenURL, RSS, ATOM.
- Dispone de una extensa y completa documentación oficial, mantenida por sus desarrolladores y la comunidad.

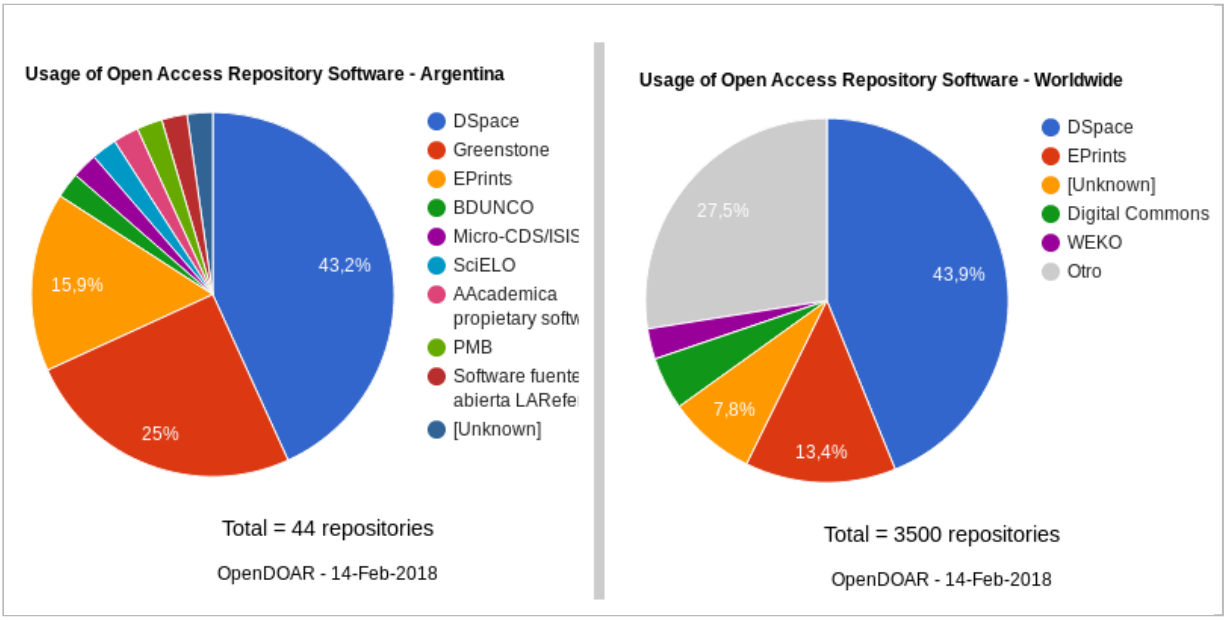


Figura 3.1 - Porcentaje de herramientas de repositorio utilizadas en el mundo (OpenDOAR) y en Argentina (OpenDOAR)

Descripción funcional

Como se mencionó previamente, DSpace es un software con muchas funcionalidades, pero las fundamentales son su capacidad de almacenar y describir mediante metadatos una amplia variedad de contenidos, además de presentar de forma organizada toda esta información a través de un árbol de comunidades, colecciones e ítems (entidades que forman parte del modelo de DSpace, como se verá más adelante). Los comunidades, colecciones e ítems son entidades «contenedoras» susceptibles de describirse mediante un conjunto de metadatos. A su vez, los ítems se componen por uno o más archivos, normalmente llamados «Bitstreams» dado que una vez que son subidos al repositorios son referenciados y almacenados como una secuencia de bits. Una cuestión a considerar es que DSpace no soporta de forma nativa el uso de metadatos estructurados (por ejemplo, MODS), sino que sólo gestiona metadatos simples (por ejemplo, Dublin Core).

Además de la capacidad de almacenamiento y descripción de recursos, otra funcionalidad clave ofrecida por DSpace es la de permitir al usuario encontrar la información deseada de manera rápida y sencilla, mediante la búsqueda en el repositorio a través de los metadatos de los ítems ó por el contenido o *full-text* de los bitstreams. Para complementar este método de búsqueda DSpace tambień permite la navegación por todo el árbol de Comunidades, Colecciones e Ítems, la exploración a través de agrupaciones de categorías (materias, autores, etc.), además de permitir la referencia externa a los elementos del repositorio a través de un identificador persistentes como Handle (Handle, 2018).

Otra de las principales características que provee DSpace es que está optimizado para la indexación de Google, esto permite que el contenido alojado en el repositorio sea indexado por Google Search y Google Scholar, característica que resulta muy importante dado que, según la documentación de DSpace «Repositorios populares soportados por DSpace obtienen un 60% de sus visitas de las páginas de Google» (DuraSpace Wiki, 2018). Asimismo, DSpace implementa OpenSearch, un conjunto de formatos simples que permite describir un motor de búsqueda para el compartimiento de resultados de búsquedas (mediante ATOM y RSS) realizadas por otras aplicaciones clientes (p.e. un explorador), aumentando así la visibilidad de los contenidos del repositorio.

Con respecto a la interoperabilidad entre repositorios, es decir, la capacidad de intercambiar información entre diferentes dos o más sistemas, DSpace implementa distintos protocolos, entre los que podemos nombrar: OAI-PMH, Sword y Sword-v2.

Modelo de datos

Las contenidos en DSpace se distribuyen a lo largo de una estructura jerárquica, como se diagrama en la Figura 3.2, donde el último nodo son los archivos que los usuarios suben al repositorio. Para representar toda esta jerarquía de datos, DSpace define un modelo de datos que consta de seis entidades principales: *Comunidad, Colección, Item, Bundle, Bitstream y Bitstream Format.* Todas estas entidades se agrupan dentro de una categoría más general llamada *DSpace Object*, por lo tanto se podría decir todas ellas son DSpace Objects. Aunque el modelo de DSpace consta de más entidades, estas están más bien relacionadas a la gestión de usuarios y de permisos de la aplicación y no a las publicaciones en sí, por eso no son de consideración en este apartado.

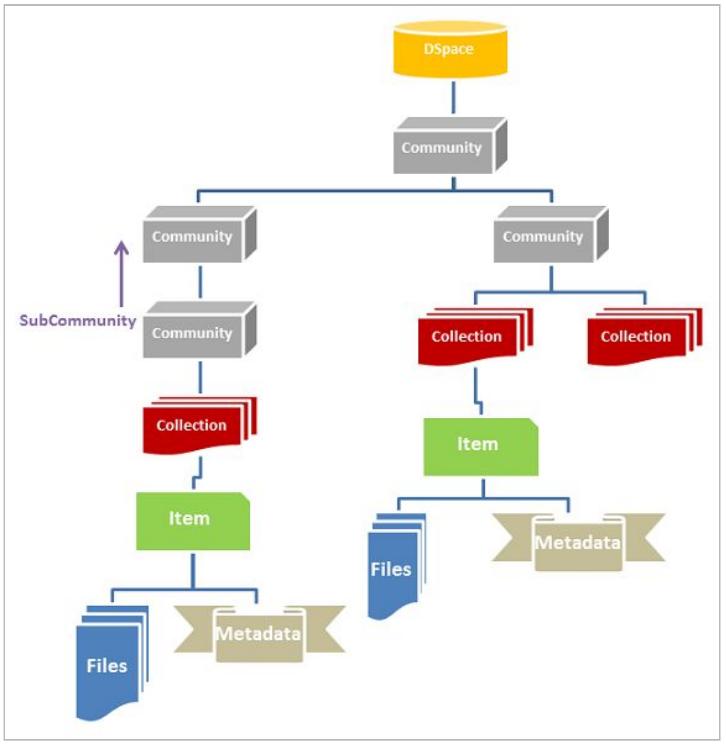


Figura 3.2 - Representación jerárquica de contenidos en DSpace

Como se mencionó, un repositorio construido sobre DSpace está compuesto por una jerarquía o árbol de entidades contenedoras, cuyo modelo de datos se visualiza en la Figura 3.3, y en el primer nivel o nodo en este árbol se encuentra las Comunidades. Una *Comunidad* representa una primer categoría de entidades contenedoras, y puede estar compuesta por otras Comunidades (conocidas como *sub-comunidades*) u otras entidades llamadas Colecciones; la posibilidad de crear sub-comunidades permite representar las jerarquías existentes en las Instituciones (como por ejemplo, las jerarquía existente entre una Universidad y sus Facultades, y éstas últimas a su vez se dividen en Departamentos). Las *Colecciones* representan la segunda categoría de contenedores, y pueden estar ubicadas dentro de una o más Comunidades, es decir, pueden poseer más de una nodo padre, lo que permite no repetir información en distintas partes del árbol; las Colecciones agrupan Ítems, que son el contenido principal y central en el modelo de un repositorio DSpace. Un *Ítem* representa una publicación enviada o agregada al repositorio y contiene metadatos que describen la obra o la publicación en cuestión. Un Ítem sólo pertenece a una única Colección aunque puede ser «referenciado» desde diferentes Colecciones, y se compone por un conjunto de entidades llamadas *Bundles*, entidades simples que agrupan los archivos o Bitstreams subidos por los usuarios o administradores del repositorio; existen distintos tipos de Bundles, como se visualiza en la Tabla 3.1. Los *Bitstreams* son los nodos hoja de esta jerarquía y, como su nombre lo indica, es una secuencia de bits que identifica y localiza al archivo depositado en el repositorio. Por último, cada Bitstream está asociado con un formato o *Bitstream Format*, entidad que representa el tipo de archivo (PDF, PNG, ODF, etc) asociado al Bitstream; DSpace cuenta con un extensa y extensible registro de

tipos de archivos lo que permite reconocer una amplia variedad de archivos que pueden ser almacenados en el repositorio.

ORIGINAL	Contiene los bitstreams publicados luego del depósito de una publicación de parte de un usuario o la administración del repositorio.
TEXT	Contiene el texto completo (full-text) de otros bitstreams. Se genera a partir de la extracción automática de texto sobre otros bitstreams y se usa durante la indexación para mejorar los resultados de búsqueda.
THUMBNAILS	Son archivos con miniaturas extraídas a partir de los bitstreams originales (p.e. thumbnails de PDFs, imágenes, etc.) y que permite hacer un vistazo rápido sobre el contenido.
LICENSE	Contiene la licencia del repositorio que el usuario aceptó al depositar el contenido. Esta licencia generalmente describe los permisos que se tiene sobre el contenido, tales como transformaciones para la preservación a largo plazo.
CC LICENSE	Contiene la licencia de uso que especifica lo que el usuario final puede hacer con el archivo asociado a la publicación, es decir, el *bitstream* en el bundle ORIGINAL.

Tabla 3.1 - Tipos de Bundles existentes en DSpace

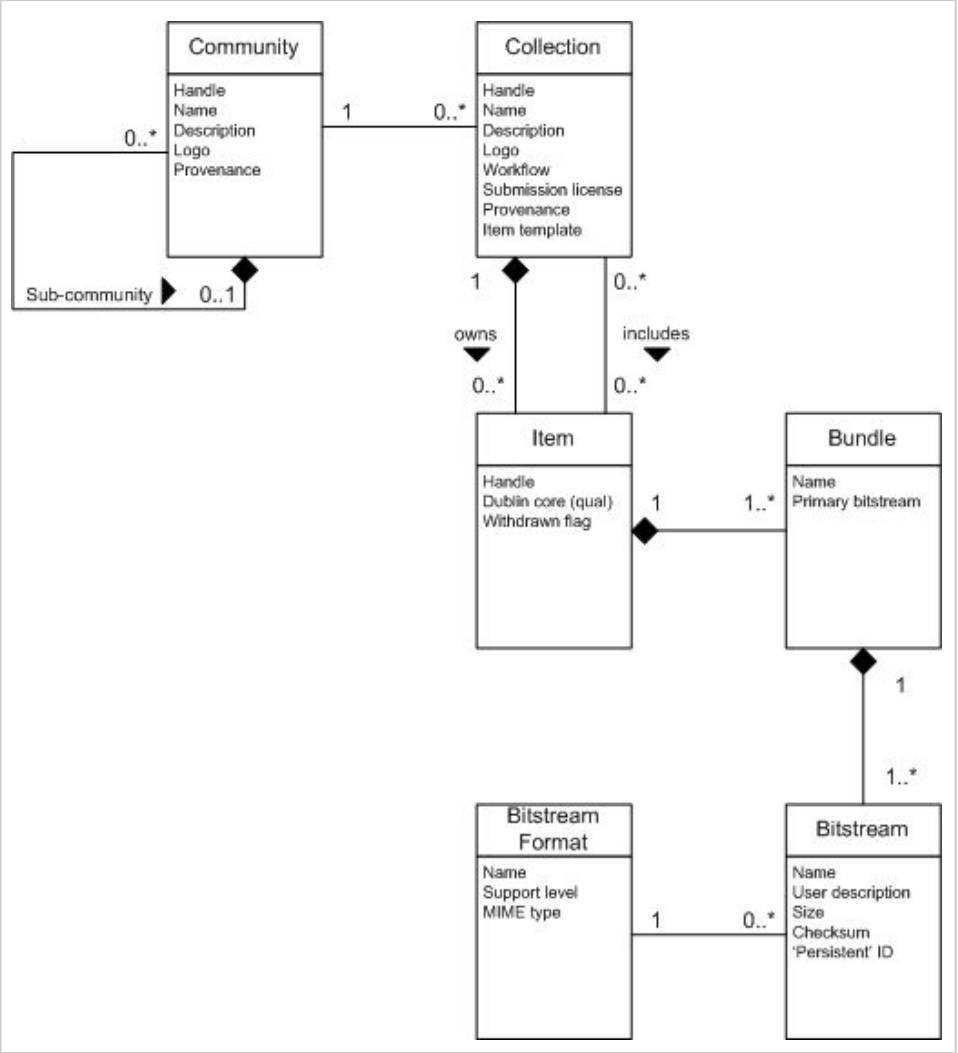


Figura 3.3 - Diagrama de clases del modelo de contenidos en DSpace

Metadatos

Como se observa en la Figura 3.2, otro de los componentes principales de un Ítem son sus metadatos. En términos generales, los *Metadatos* pueden definirse como datos que describen datos, y en particular, los metadatos guardan información acerca de un elemento o aspecto en particular de las entidades que conforman el modelo de DSpace. La capacidad de vincular metadatos con las distintas entidades del modelo, ya sea una Comunidad, Colección, Ítem o Bitstream, permite el almacenamiento de información detallada sobre cualquier elemento alojado en el repositorio. Los metadatos están divididos en tres categorías: Descriptivos, Administrativos y Estructurales. Cada una de las categorías contiene diferente tipo de información: los metadatos Descriptivos almacenan información que permite describir y recuperar el elemento, por ejemplo el título de una tesis, los metadatos Administrativos contienen información para gestionar el recurso es decir información sobre el mantenimiento del mismo o gestión de derechos entre otros, y los metadatos Estructurales contienen información sobre la estructura interna de los elementos y cómo presentarlo al usuario.

De forma nativa, DSpace administra un conjunto de esquemas de metadatos básico, entre los que podemos encontrar el esquema Dublin Core. **Dublin Core** («DCMI: DCMI Abstract Model», 2007; «DCMI: DCMI Metadata Terms», 2012) es un modelo de metadatos, auspiciado por DCMI (Dublin Core Metadata Initiative), que define un conjunto de términos que puede ser utilizado para describir recursos digitales (videos, imágenes, páginas web, etc), así también para recursos físicos como libros o CDs, y objetos artísticos. DSpace cuenta con la versión extendida de Dublin Core llamada «*DCMI Terms*» (con más de 50 metadatos) y la versión simplificada llamada «*Dublin Core Element Set*» (de sólo 15 metadatos) de Dublin Core. Este esquema es utilizado por varios grupos de interés, como por ejemplo Bibliotecas, Instituciones del gobierno, Sectores científicos de la investigación, entre otros.

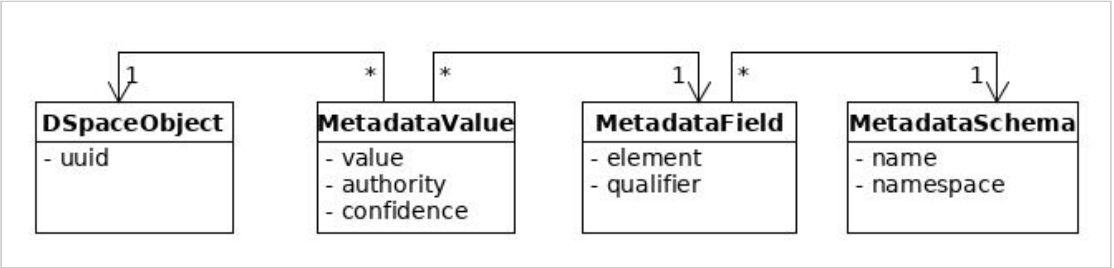


Figura 3.4 - Entidades para la representación de metadatos en DSpace

Internamente a la aplicación, los metadatos son representados utilizando las entidades mostradas en la Figura 3.4:

- *MetadataField y MetadataSchema*: un metadato se compone de (esquema).(elemento).[(calificador)], donde la definición del calificador es opcional. Cada esquema de metadatos (*MetadataSchema*) define un espacio de nombres o namespace (W3C, 2009) único conformado por varios metadatos, y cada metadato (MetadataField) es definido unívocamente en ese namespace. Por ejemplo: el metadato «dc.creator» representa a «la persona u organización responsable de la creación del contenido intelectual del recurso» y está definida unívocamente en el *namespace* http://purl.org/dc/elements/1.1/.

- *MetadataValue*: los objetos del repositorio en DSpace (DSpaceObjects) pueden tener múltiples instancias de un mismo metadato (MetadataValue), y estas instancias pueden tener los mismos o diferentes valores entre sí; asimismo, cada instancia de un metadato pertenece a un único objeto. Por ejemplo: el metadato «dc.creator» puede ser «Pedro C.» en el ítem 1, «Mario Paredes» en el ítem 2, y «Pedro C.» en el ítem 3.

Arquitectura

DSpace posee una compleja arquitectura («Architecture - DuraSpace», s. f.) que se divide en 3 grandes capas o áreas: aplicación, lógica de negocios y almacenamiento, graficada en la Figura 3.5. La *capa de aplicación* incluye todas las herramientas que permiten al exterior (usuarios u otros sistemas) hacer uso del repositorio; por ejemplo, existen distintas módulos que funcionan como punto de acceso al repositorio («User

Interfaces - DuraSpace», s. f.), tales como XMLUI, JSPUI, OAI Server, Discovery, REST-API, entre otros. La capa intermedia de negocios mantiene la lógica transversal a todas las aplicaciones y rige el funcionamiento interno del repositorio. Finalmente la capa de almacenamiento se encarga de todas las tareas específicas de guardado y recuperación desde almacenamiento secundario, es decir, bases de datos y sistema de archivos.

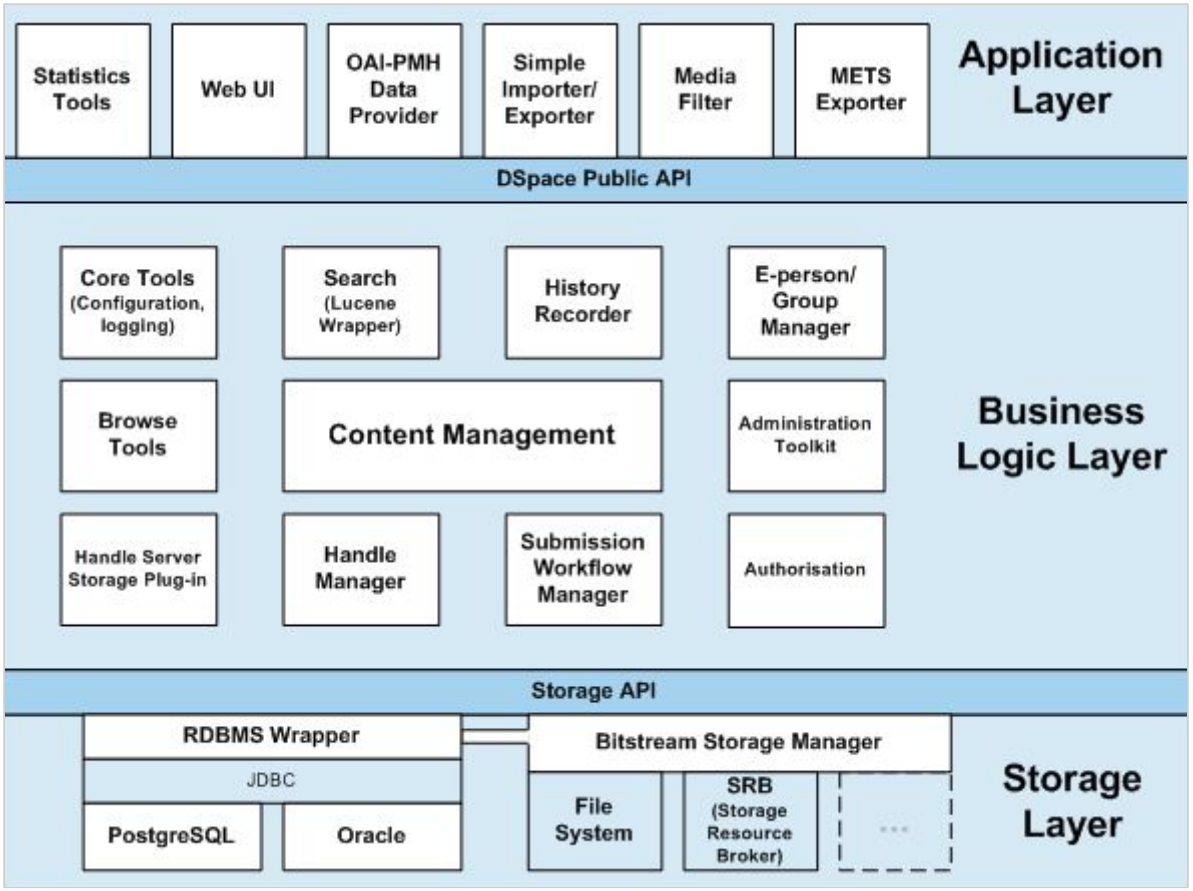


Figura 3.5 - Arquitectura de capas en DSpace

Como se define en los tipos de arquitectura multicapa («Programación por capas», 2018), cada capa sólo invoca la capa debajo de ella mediante el uso de APIs[7]; la capa de aplicación no puede usar la capa de almacenamiento directamente, por ejemplo. Además este diseño arquitectural no sólo ordena la comunicación entre capas adyacentes, sino que reduce el nivel de acoplamiento entre capas, minimizando las dependencias de una capa en otras capas y trayendo beneficios para el mantenimiento de la capa, la actualización, y la escalabilidad. Cada componente en las capas de lógica de almacenamiento y negocios tiene una API pública definida: la API de almacenamiento (en el caso de la capa de almacenamiento) y la API pública de DSpace (en el caso de la capa de lógica de negocios).

A partir de la versión 6.X de DSpace, se realizó un refactoring de todas sus APIs de tal forma de integrar Hibernate[8] para soporte de la capa de persistencia y mejorar la

[7] Application Programming Interface (API) es un conjunto de subrutinas, funciones y procedimientos o métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
[8] http://hibernate.org/

estructura del código interno a la lógica de DSpace en la capa de negocios, separando aún mejor las responsabilidades en cada capa; p.e. existían muchos casos dónde estaba mezclado código SQL en los elementos pertenecientes a la «capa de lógica de negocios». Debido a cuestiones como éstas, la API pública de DSpace se fue transformando en sus últimas versiones en una API basada en Servicios («DSpace Service based api - DuraSpace», s. f.), dividiéndola en las siguientes partes:

- **Capa de Servicios**: esta capa es totalmente pública y puede ser utilizada por la Capa de Aplicación. Se divide en 2 *subcapas*:
    - *Servicios basados en la base de datos*: servicios que darán acceso a operaciones CRUD sobre los objetos en base de datos y a operaciones propias de la lógica de cada objeto. Las operaciones CRUD son delegadas a objetos DAO que implemente estas operaciones, evitando concentrar código de base de datos en esta capa.
    - *Servicios de la capa de negocios*: servicios que se corresponden a los antiguos *Managers* en DSpace (p.e. AuthorizationManager).
- **Capa de acceso a la base de datos**: esta capa solo puede ser accedida únicamente desde alguna de los servicios anteriores, y está habitada por objetos que implementan interfaces DAO para acceder a la base de datos. Gracias al uso de esta capa se puede configurar la aplicación con distintos ORMs (no solo *Hibernate*) y distintos motores de base de datos (responsabilidad del ORM utilizado).
- **Objetos de base de datos**: estos son objetos que se corresponden unívocamente con las tablas en la base de datos, y sólo tienen métodos getters y setters sobre las columnas de las tablas.

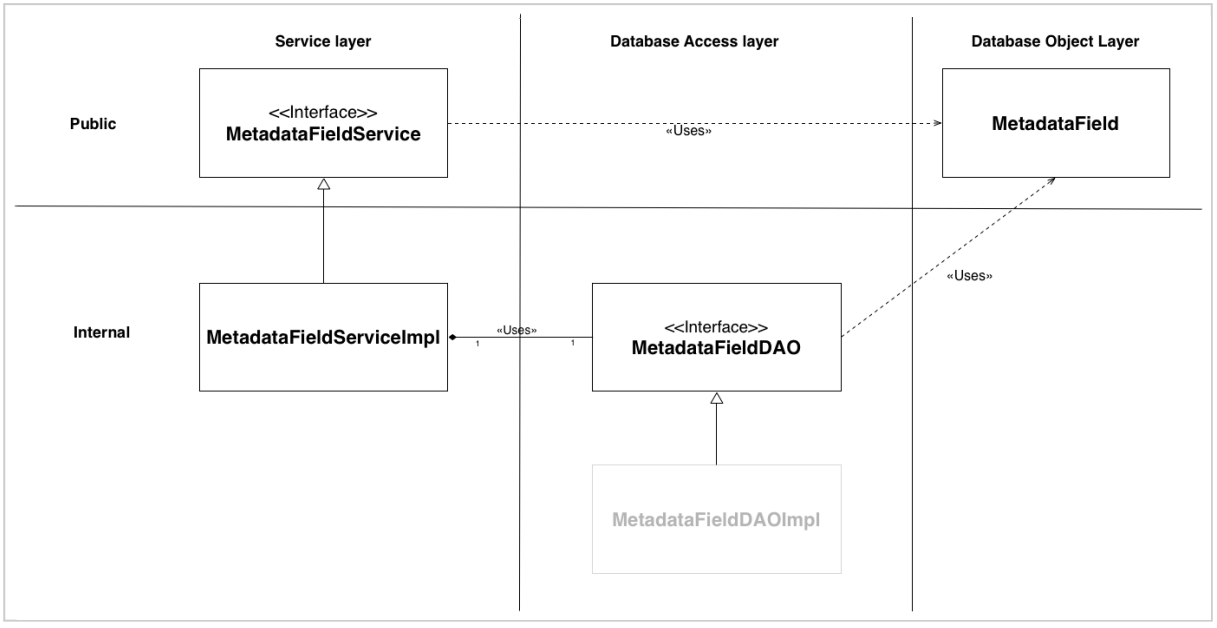


Figura 3.6 - Ejemplificación de API basada en servicios en DSpace

Solr

DSpace viene empaquetado con Apache Solr («Apache Solr», 2018), una plataforma de búsqueda empresarial de código abierto escrita en Java, del proyecto Apache Lucene. Sus características principales incluyen:

- **Búsqueda a texto completo**: Solr proporciona todas las capacidades necesarias para una búsqueda de texto completo[9] como tokens, búsqueda de frases, chequeo de deletreo o *spell-checking*, wildcards, and auto-completado.
- **Hit highlighting**: permite identificar las porciones o fragmentos de los documentos que coincidieron con un término de búsqueda ingresado por un usuario.
- **Búsqueda facetada**: permite la clasificación de cada elemento de información a lo largo de múltiples dimensiones explícitas, llamadas facetas o facets, lo que permite el acceso y la ordenación de los datos en múltiples formas, en lugar de en un único orden taxonómico predeterminado.
- **Restful APIs:** Para comunicarse con Solr y realizar operaciones CRUD en los documentos indexados se puede utilizar el protocolo HTTP mediante servicios RESTful.
- **Flexible y extensible**: mediante la extensión de clases de Java en Solr y una correcta configuración, podemos personalizar los componentes de Solr fácilmente y agregar comportamiento personalizado.
- **Basado en Lucene**: Solr funciona sobre una simple pero poderosa librería de búsqueda, escrita en Java, llamada Apache Lucene[10]. Lucene es una librería escalable y de alto rendimiento utilizada para indexar y buscar prácticamente cualquier tipo de texto, y puede ser utilizada en cualquier aplicación para agregar capacidad de búsqueda.
- Solr también presenta características NoSQL, capacidad de manejo de documentos enriquecidos (por ejemplo, Word, PDF), y la devolución de resultados de búsqueda en variados formatos como XML, CSV, JSON, entre otros.

Al proporcionar replicación de índices y búsqueda distribuida, Solr está diseñado para la escalabilidad y tolerancia a fallas. Se usa ampliamente para casos de uso de búsqueda y análisis empresarial, y tiene una comunidad activa de desarrollo y lanzamientos regulares.

La unidad de información básica de Apache Solr es llamada *documento* (Seeley, s. f.; «Chapter 2 - Understanding Apache Solr», 2015), que es un conjunto de datos que describe algo. Cada documento en Solr se compone de campos o *fields*, y por cada uno se deben definir sus atributos: el nombre, el tipo de dato (string, integer, boolean, etc.), la obligatoriedad al momento de indexar, si debe comprimirse al almacenarse, entre otros. De esta forma, se configura Solr para que comprenda la estructura de los datos que ingresan (procedente de varias fuentes) a partir de su nombre. Estos campos, una vez definidos, estarán disponibles en el momento de la importación de datos o la carga de datos.

Los documentos en Solr se ubican en entidades de almacenamiento llamadas **cores** o núcleos. Un núcleo Solr no es más que la instancia en ejecución de un índice Solr junto

[9] https://en.wikipedia.org/wiki/Full-text_search
[10] http://lucene.apache.org/

con su configuración. Una instancia de Apache Solr puede ejecutarse como un solo núcleo o multinúcleo, administrando de forma unificada múltiples esquemas y configuraciones, una por núcleo.

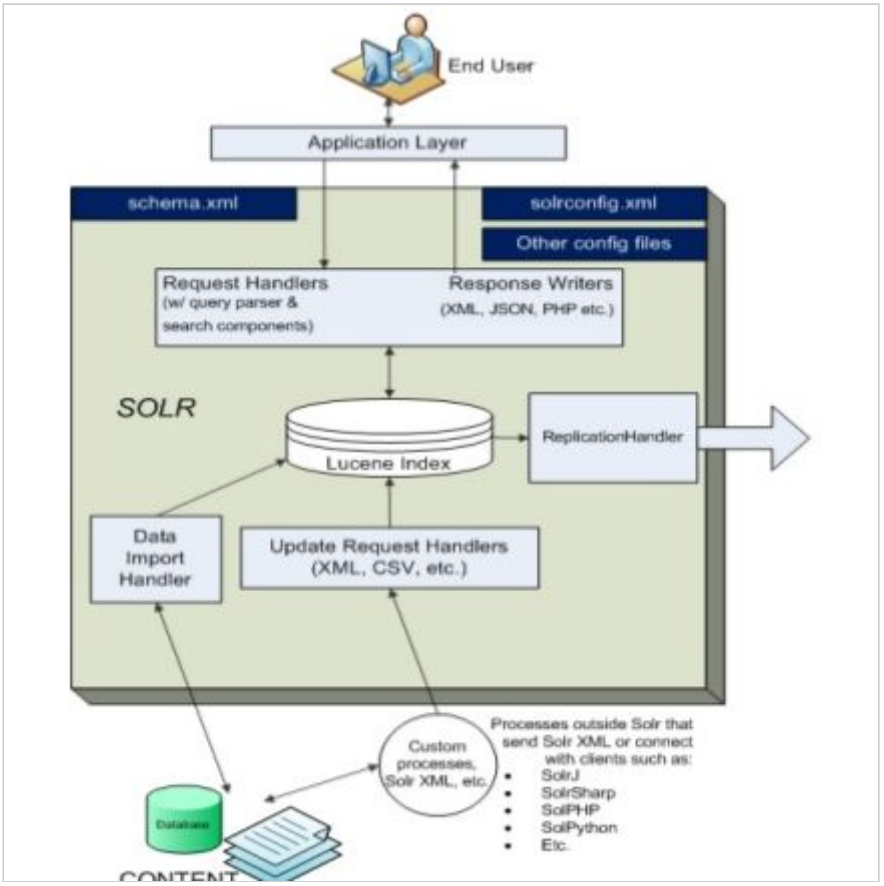


Figura 3.7 - Arquitectura de Solr

En síntesis, Solr es una motor de búsqueda y almacenamiento optimizado para buscar en grandes volúmenes de datos textuales. Como se observa en la Figura 3.7 la arquitectura de Solr (Seeley, s. f.; «Chapter 2 - Understanding Apache Solr», 2015) consta de varios componentes extensibles para administrar los datos que son indexados. En esta figura se muestra cómo una aplicación externa agrega datos o documentos a Solr (*indexación*) mediante manejadores de actualización (*UpdateRequestHandlers*) o manejadores de importación (*DataImportHandlers*) en un determinado "core" de Solr, es decir, una instancia en funcionamiento de un índice Lucene que aloja una colección de documentos. Luego, estos datos pueden ser consultados por cualquier aplicación cliente mediante la interfaz de búsqueda de Solr, consulta que inmediatamente es parseada y procesada por un manejador de peticiones (*RequestHandlers*) para traducirla a Lucene, y retornar los resultados correspondientes (documentos) en cualquiera de los formatos de respuesta disponibles (JSON, CSV, XML, javabin[11], etc) utilizando los manejadores de respuesta (*ResponseWriters*) provistos por el motor.

Por último, como se observa en la Figura 3.7, Solr tiene 3 archivos de configuración principales que definen su comportamiento y la estructura de los datos que importa:

[11] https://wiki.apache.org/solr/javabin

Archivo	Descripción
`solrconfig.xml`	Este es el archivo de configuración principal en una instalación Solr. Permite controlar muchos aspectos de la aplicación, desde el almacenamiento en caché hasta los manejadores de consulta. Existe uno por cada core definido.
`schema.xml`	Define el esquema de los documentos administrados por un core en particular, especificando campos, tipos de datos, operadores por defecto, entre otras cosas. Existe uno por cada core definido.
`solr.xml`	En este archivo se definen los cores que la instancia Solr administra.

Tabla 3.2 - Principales archivos de configuración en Solr

Todas estas características y opciones que ofrece Solr la convierten en una potencial herramienta a tener en consideración para aplicaciones que administran, o pueden llegar a generar y explotar, un gran volumen de datos.

Solr en DSpace

Los datos primarios de DSpace (metadatos, usuarios, permisos, comunidades, colecciones, entre otros) son almacenados en una única base de datos relacional (PostgreSQL por ejemplo), ubicada dentro de la capa de almacenamiento en su arquitectura en capas. Sin embargo, hay ciertas funcionalidades en la aplicación que hacen uso de otros datos alojados en sistemas adicionales y complementarios como, por ejemplo, la plataforma Solr que viene integrada a DSpace. En particular DSpace usa Solr para registrar los siguientes datos:

- datos derivados desde los datos en la base de datos relacional de DSpace, como por ejemplo los utilizados por el servicio de búsqueda Discovery («Discovery - DuraSpace», s. f.) y por el servicio de exposición de registros mediante OAI-PMH[12] («OAI 2.0 Server - DuraSpace», s. f.)
- datos que complementan los datos en la base de datos de DSpace, como por ejemplo datos sobre autoridades ORCID («ORCID Integration - DuraSpace», s. f.) para identificar unívocamente autores de publicaciones, y
- datos derivados a partir de eventos que suceden en la aplicación, como por ejemplo los datos de las estadísticas de uso generados por el uso que otros sistemas o usuarios realizan sobre el repositorio («SOLR Statistics - DuraSpace», s. f.).

En los anteriores casos de uso, se utilizan cores independientes para cada situación, donde los volúmenes entradas en los cores pueden llegar a crecer mucho, en algunos casos más que otros, como por ejemplo: el core que almacena el volumen de datos de uso de estadística puede llegar a crecer a millones de registros y, en cambio, el volumen de datos del core del servicio de búsqueda *Discovery* siempre es lineal a la cantidad de objetos en la

[12] OAI-PMH es un protocolo que define un marco de trabajo para la interoperabilidad de metadatos entre sistemas informáticos en el ámbito de repositorios digitales.

jerarquía de comunidades y colecciones de DSpace. Por defecto, DSpace viene con cuatro cores Solr: «search», «oai», «statistics» y «authority».

En cuanto a la consulta de los datos en los cores Solr en Dspace, Solr se utiliza para realizar búsquedas de términos libres y búsquedas filtradas (y facetadas) sobre todos los objetos de la jerarquía de contenidos del repositorio (a través del módulo *Discovery*), además de realizar búsquedas en el texto completo extraído de las publicaciones archivadas y en todos los campos de metadatos de cada ítem. Como se verá en la siguiente sección, los datos de uso del repositorio también son consultados para generar un pequeño conjunto de reportes de acceso a comunidades, colecciones e items.

En particular para este trabajo, solamente se analizarán los casos de uso para el módulo que administra el core de datos estadísticos y el módulo que administra el core del servicio de búsqueda *Discovery*.

Módulo de estadísticas

DSpace cuenta con un módulo de estadísticas llamado «DSpace Statistics» que se encarga de generar reportes a partir de los accesos o visitas de páginas del repositorio, las descargas de bitstreams, las búsquedas en el repositorio y los eventos de workflow en el repositorio («SOLR Statistics - DuraSpace», s. f.). DSpace Statistics es una arquitectura cliente/servidor basada en Solr que recopila eventos de uso en las aplicaciones de interfaz de usuario JSPUI y XMLUI de DSpace. Como se mencionó en las anteriores secciones de este trabajo, el módulo de estadísticas en DSpace registra todos estos eventos de uso en un core Solr específico llamado *statistics.*

Indexación de eventos

El registro de eventos de uso ocurre en el lado del servidor, y no utiliza técnicas de Javascript tracking como lo hace *Google Analytics*, para proporcionar datos de uso. La definición de los campos que se van a almacenar en el core 'statistics' se encuentra en el archivo `solr/statistics/conf/schema.xml` en el directorio de instalación de DSpace. En este core se indexan distintos tipos de eventos de uso y, aunque estén todos en el mismo índice, los campos almacenados para las descargas, consultas de búsqueda y eventos de workflow son diferentes. Por cada documento en este core, un campo en particular llamado *statistics_type* determina qué tipo de evento de uso está tratando. Los tres valores posibles para este campo son: **view**, **search**, y **workflow**[13]. A continuación se detalla los campos almacenados para cada tipo de registro:

Campos comunes almacenados para todos los tipos de evento de uso	
`type`	Constante que indica el tipo de objeto asociado al registro (*bitstream, item,*

[13] Para gestionar la carga de ítems en un repositorio, DSpace ofrece un mecanismo de revisión que determina cuándo un envío de una publicación está apto o no para su depósito en el repositorio. Este mecanismo recibe el nombre de «Workflow» y básicamente define un conjunto de pasos a seguir. Cuando estos pasos son ejecutados sin interrupción hasta el final, entonces el envío es archivado en el repositorio.

	colección, o *comunidad*).
`id`	Identificador del objeto asociado al registro. Es usado conjuntamente al campo *type* para identificar unívocamente el recurso solicitado.
`ip`	Indica la IP de la máquina asociado el evento de uso registrado.
`time`	Indica la fecha y hora (acorde a la **ISO 8601**[14]) en la que el evento de uso tuvo lugar.
`epersonid`	Indica el ID de la persona logueada en el sitio que realizó el evento de uso. Si la persona no realizó log-in (inicio de sesión) en el sitio, este campo está vacío.
`continent`	Indica el continente (acorde a la **ISO 3166-1 alpha-2**[15]) desde el que se registró el evento de uso. Derivado a partir de la IP.
`country Code`	Indica el país (acorde a la **ISO 3166-1 alpha-2**) desde el que se registró el evento de uso. Derivado a partir de la IP.
`city`	Indica el ciudad desde el que se registró el evento de uso. Derivado a partir de la IP.
`longitude`	Indica una coordenada de longitud derivada a partir de la IP del registro.
`latitude`	Indica una coordenada de latitud derivada a partir de la IP del registro.
`owning Comm`	Indica la comunidad/comunidades ancestras del objeto DSpace asociado al registro.
`owning Coll`	Indica las colecciones en las que se encuentra el objeto DSpace asociado al registro.
`owning Item`	Indica el ítem asociado al registro, sólo válido cuando el campo *type* asociado al registro es *bitstream*.
`dns`	Indica el dominio asociado a la IP del registro a través de una búsqueda DNS inversa[16].
`user Agent`	Indica el nombre del agente de usuario que viene en la cabecera HTTP.
`isBot`	Flag que indica si el registro es un acceso de un bot[17].
`referrer`	Identifica la dirección de la página web que se vinculó con el recurso que se solicita. Con esta referencia, la nueva página web puede ver dónde se originó la solicitud.
`uid`	Representa el identificador único del registro dentro del core *statistics*.
`statistic s_type`	Indica el tipo de evento de uso que se está registrando: *search, view o workflow.*
`Campos únicos para eventos de descargas de bitstreams`	

[14] https://en.wikipedia.org/wiki/ISO_8601
[15] https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2
[16] https://en.wikipedia.org/wiki/Reverse_DNS_lookup
[17] Un **bot** es un software que realiza una tarea automatizada a través de Internet. Uno de los mejores ejemplos de un bot son las “spiders” de los motores de búsqueda, que se encargan de indexar todas las páginas web accesibles desde Internet.

`bundleName`	Cuando se está registrando una descarga de bitstream, este campo indica el nombre del Bundle en donde ese bitstream está ubicado (por ejemplo, el bundle ORIGINAL).
Campos únicos para eventos de búsquedas (type='search')	
`query`	Indica el/los términos de búsqueda asociado al evento de búsqueda.
`scopeType`	Indica el tipo de contenedor (colección o comunidad) sobre el que se realizó la búsqueda. Si este campo no existe, entonces la búsqueda fue realizada en todo el RI.
`scopeId`	Indica el identificador del contenedor asociado a la búsqueda.
`rpp`	Para búsquedas paginadas, indica la cantidad solicitada de resultados de búsqueda por página.
`sortBy`	Indica el campo solicitado utilizado para ordenar los resultados de búsqueda.
`sortOrder`	Indica el orden utilizado (ascendente o descendente) para ordenar los resultados de búsqueda.
`page`	Para búsquedas paginadas, indica el número de página solicitado.
Campos únicos para eventos de workflow (type='workflow')	
`workflowStep`	Indica en qué paso del *workflow* estaba un ítem al momento de registrar el evento.
`previousWorkflowStep`	Indica qué paso fue ejecutado anteriormente al indicado por el campo *workflowStep.* Si este campo está no existe en el registro, entonces es el primer paso ejecutado del workflow.
`owner`	Indica el identificador del grupo o usuario configurado para ejecutar el paso de workflow actual, según configuración del workflow.
`submitter`	Indica el usuario que inició el proceso de workflow luego de enviar un ítem para su archivo en el repositorio.
`actor`	Indica el ID del usuario que efectivamente ejecutó el paso de workflow asociado al evento.
`workflowItemId`	Indica el ID del item que está siendo revisado durante el proceso de workflow asociado al evento.

Tabla 3.3 - Campos de los registros estadísticos indexados en Solr

El módulo de estadísticas implementa un servicio llamado *SolLoggerServiceImpl* (ver Figura 3.8) que se encarga de registrar los distintos eventos de uso que suceden en las aplicaciones de interfaz de usuario («User Interfaces - DuraSpace», s. f.) que existen en DSpace. Cada una de estas aplicaciones va registrando acciones de eventos de uso que, posteriormente, serán procesadas por un listener[18] implementado por el módulo, el cual se

[18] Un *listener,* o también llamado manejador de evento, es un objeto que recibe la notificación de que un evento fue lanzado en algún lugar de la aplicación, y actúa en respuesta a ese evento.

llama *SolrLoggerUsageEventListener.* Acorde al tipo de evento de uso del que se trate, el listener invoca alguna de las siguientes implementaciones del *SolLoggerServiceImpl:*

- `postSearch()`: método que indexa un nuevo documento Solr para registrar un evento del tipo *"search"* con los campo correspondientes mostrados en la Tabla 3.3.
- `postView()`: método que crea un nuevo documento Solr para registrar un evento del tipo *"view"* con los campo correspondientes mostrados en la Tabla 3.3. Utilizado tanto para descargas de Bitstreams como vistas de Ítems, Comunidades y Colecciones.
- `postWorkflow()`: método que crea un nuevo documento Solr para registrar un evento del tipo *"workflow"* con los campo correspondientes mostrados en la Tabla 3.3.

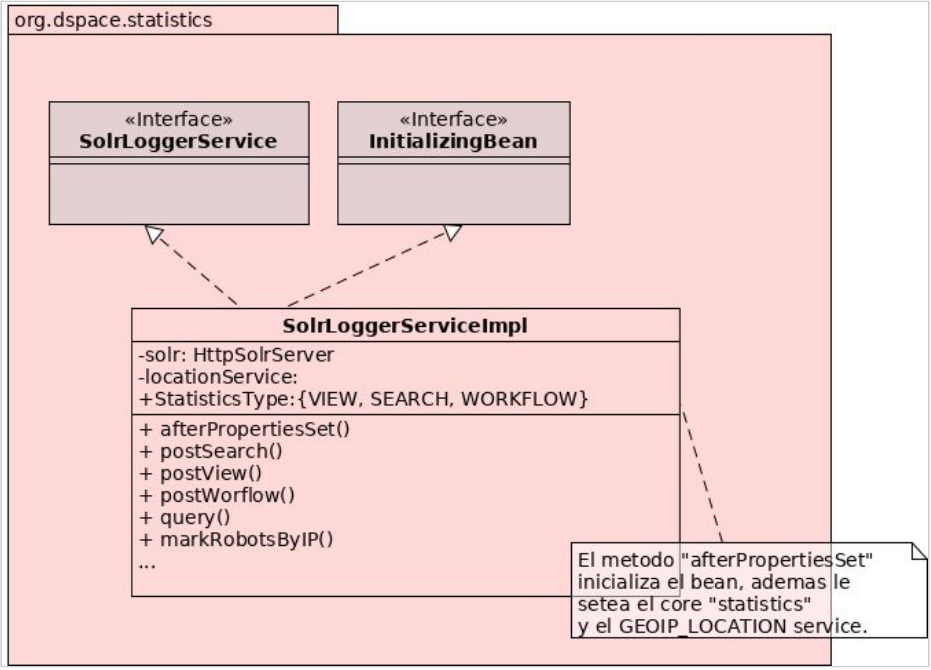


Figura 3.8 - Servicio de indexación DSpace Solr Statistics

Además de brindar funciones de indexación, este servicio también ofrece algunas funcionalidades de búsqueda sobre el core de Solr, así como de optimización. Una de las funciones más importantes que ofrece es la de detectar y marcar aquellos registros del core que son bots, detectados a partir del campo IP del registro o del campo *userAgent* de cada registro. Por último, este servicio utiliza bases de datos de geolocalización gratuitas, ofrecidas por el servicio *GeoLite*[19], a partir de las cuales se determinan distintos datos geográficos (latitud, longitud, país, ciudad, etc.) derivados desde la IP de un registro.

Generación de reportes

El módulo DSpace Statistics implementa páginas con tablas de reportes para los distintos eventos de uso registradas en Solr («SOLR Statistics - DuraSpace», s. f.) en las interfaces de usuario JSPUI y XMLUI. A continuación se indican éstos reportes por cada tipo de evento:

[19] https://dev.maxmind.com/geoip/legacy/geolite/

Accesos y Descargas

- Existe una página de estadísticas general que muestra los 10 ítems más populares de todo el repositorio, es decir, con mayor cantidad de accesos.
- En la página de estadísticas por comunidad se muestra: total de visitas de la página de inicio de la comunidad actual, visitas a la página principal de la comunidad durante un intervalo de tiempo de los últimos 7 meses, top 10 de países que más visitas realizaron a la comunidad, top 10 de ciudades que más visitas realizaron a la comunidad.
- En la página de estadísticas por colección se muestra: total de visitas a la página de la colección actual, visitas mensuales a la página de la colección actual durante un período de los últimos 7 meses, top 10 de países que más visitas realizaron a la colección, top 10 de ciudades que más visitas realizaron a la colección.
- En la página de estadísticas por ítem se muestra: total de visitas del ítem, total de visitas/descargas para los bitstreams adjuntos al ítem, visitas mensuales del ítem durante un período de los últimos 7 meses, top 10 de países que más visitaron el ítem, top 10 de ciudades que más visitaron el ítem.

Estadísticas de consultas de búsqueda

- Existe una página general de búsquedas realizadas en todo el repositorio, en la que se muestra una tabla (ver Figura 3.9) con el top 10 de términos más buscados a través del módulo de búsqueda Discovery. Por cada término se indica: la cantidad de veces que fue buscado (tercer columna), el porcentaje que representa en el total de búsquedas (cuarta columna), y la cantidad de páginas visitadas después de un término de búsqueda particular (última columna). Para este último dato, si su valor es un cero significa que después de ejecutar una búsqueda de una palabra clave específica, ningún usuario ha hecho clic en un solo resultado en la lista.


Términos de Búsqueda mas usados

Total

	Término de Búsqueda	Búsquedas	% del total	Páginas Vistas / Búsquedas
1		1074	8.30%	0.88
2	has_content_in_original_bundle_keyword:true	1023	7.91%	0.00
3	subject_keyword:keyword1	801	6.19%	0.00
4	subject_keyword:keyword2	719	5.56%	0.00
5	subject_keyword:keyword3	638	4.93%	0.00
6	dateIssued_keyword:[1900 TO 1999]	498	3.85%	0.00
7	author_keyword:Cat, Lily	441	3.41%	0.00
8	subject_keyword:cat	354	2.74%	0.00
9	author_keyword:Doe, Jane L	322	2.49%	0.00
10	dateIssued_keyword:[1650 TO 1699]	318	2.46%	0.00

Total

Búsquedas	% del total	Páginas Vistas / Búsquedas
12940	100.00%	0.12

Figura 3.9 - Página de estadísticas de búsqueda en DSpace

- También se puede seleccionar que, en vez de mostrar las búsquedas de todos los tiempos, se limite el reporte a un rango de tiempo determinado: el año anterior, 6 meses antes, el mes anterior.
- Existen páginas similares a la anterior para representar las búsquedas por comunidad y colección, mostrando el top 10 de búsquedas tomando como contexto esa comunidad o colección.

Estadísticas de eventos de workflow

- Por último, existe una página general de eventos de workflow que reporta los pasos activados durante el proceso de workflow para el depósito de ítems en el repositorio. Por cada paso del workflow solamente muestra la cantidad veces que fue activado.
- Al igual que en las reportes anteriores, también puede seleccionarse un rango de tiempo determinado en vez de mostrar el total: el año anterior, 6 meses antes, el mes anterior.

Capítulo 4 | Propuesta de solución

Análisis de problema

Como se explicó en el Capítulo 2 de este trabajo, la disposición de datos de uso confiables sobre el uso del repositorio son necesarios para la toma de decisiones de las autoridades del repositorio en la tarea continua de mejorar los servicios y la calidad del contenido del repositorio. Con este fin en mente, en el Capítulo 3 se analizó la arquitectura de DSpace y algunas de las herramientas que ofrece la plataforma para la obtención de datos de uso del repositorio y la generación de reportes a partir de ellos: DSpace Statistics es el módulo que se encarga de la indexación de eventos de uso en DSpace (visitas, eventos de workflow, y búsquedas) y de la generación de algunos reportes basados en estos datos desde la capa de aplicación, y Solr es la tecnología que subyace a la indexación y almacenamiento de los registros de eventos de uso en la capa de almacenamiento.

Sin embargo, a pesar de disponer de un conjunto de reportes sobre el uso del repositorio y su contenido, el módulo de estadísticas presenta algunas limitaciones que surgen al momento de querer explotar en mayor profundidad los datos indexados en Solr. Entre estas limitaciones se puede mencionar las siguientes:

- En la mayoría de los reportes retorna solo 10 de resultados, por ejemplo, el top 10 de términos más buscados, el top 10 de ítems más accedidos, y no permite seleccionar una mayor cantidad de resultados de manera arbitraria.
- No se puede seleccionar un rango de fecha arbitrario o mayor a un año de antigüedad. Los reportes sólo pueden generarse en un escueto rango de fechas predeterminado, a saber: el mes anterior, rango de 6 meses antes, rango de un año antes. Además no se puede especificar una mayor granularidad de tiempo, por ejemplo, para reportar los eventos de uso diarios.
- No ofrece ninguna funcionalidad para inspeccionar algún otro de los posibles aspectos registrados en el core de statistics, mostrados en la Tabla 3.3, como por ejemplo: IPs, Referrers, etc.
- No ofrece exportación de reportes ni permite exportar los registros involucrados en la generación de los reportes para un análisis en mayor profundidad fuera del sistema.
    - Podría servir de mucho disponer de esta capacidad y utilizarlos como datasets para análisis futuros en materia de detección de bots, tendencias en el uso del contenido del repositorio, detección de picos de actividad a lo largo del tiempo, entre otros.
- No existen visualizaciones de reportes *out-of-the-box*, es decir, que deben implementarse utilizando alguna librería de graficación y se requiere de conocimiento informático para ésto.

Además de las desventajas anteriores, el modelo de clases que subyace a la generación de reportes es limitado en algunos aspectos ya que para modificarlos hace falta disponer de conocimiento en el lenguaje de programación Java; en su lugar, sería conveniente disponer

de configuraciones o parametrizaciones que permitan estos cambios. Los siguientes son aspectos de los reportes que necesitan ser cambiados programáticamente:

- El rango de tiempo para el que se generan los reportes.
- La cantidad de filas retornadas en las tablas de reportes.
- Los filtros que determinan los registros incluidos dentro de un dataset correspondiente a cada reporte.

Estos aspectos se encuentran hardcodeados[20] en las clases que implementan la vista y el modelo del módulo de DSpace Statistics, por lo que cambiar algunas de los mismos requeriría de una completa compilación del código de la aplicación y la posterior actualización de la instalación para que surtan efecto. Esta situación no permite que, al momento de filtrar los registros a ser incluidos en el reporte, el usuario pueda determinar una condición dinámica como, por ejemplo, filtrar sólamente las accesos a ítems realizados durante los últimos 3 años para las 50 ciudades de Argentina que más accesos registraron.

Casos de uso

Luego de analizar los problemas planteados en la sección anterior existentes en el módulo DSpace Statistics, se decidió implementar un prototipo que permita al usuario del repositorio analizar los diversos datos que son indexados en el core Solr del módulo mediante una exploración de su contenido desde la interfaz de usuario del repositorio, así como también permita generar una mayor variedad de reportes a partir de diversos condiciones o filtros.

A continuación se redactan algunos casos de uso o situaciones de ejemplo que el prototipo debería ser capaz de resolver.

CASO 1: Explorar los registros de uso de una comunidad o colección específica. Muchas veces se requiere diversas estadísticas de uso para un conjunto de colecciones/comunidades específicas del repositorio, ya que generalmente (como se observó en el Capítulo 3 - Sección Modelo de datos) cada colección/comunidad específica puede corresponderse con la producción académica producida en un departamento o área dentro de la Institución que gestiona el repositorio y a veces debe evaluarse la utilidad por parte de los usuarios sobre estos recursos.
CASO 2: Explorar los registros de uso de un ítem en particular. En ciertas ocasiones, los autores de las publicaciones podrían desear ver la visibilidad que cada una de éstas logra a través del repositorio, por lo que la visualización de información sobre el uso de cada publicación (cantidad de accesos, fechas, países de origen, etc.) podría ser de utilidad para ésto.

[20] El *hardcode* es una mala práctica en el desarrollo de *software* que consiste en incrustar datos directamente en el código fuente del programa, en lugar de obtener esos datos de una fuente externa como un fichero de configuración o parámetros de la línea de comandos, o un archivo de recursos.

CASO 3: Explorar los registros de uso para los artículos de un autor específico.

Con motivo de examinar la popularidad o usabilidad de la creación intelectual de un determinado autor, muchas veces es requerido ver el uso de sus publicaciones. Esto podría servir en ciertas ocasiones como soporte en la decisión de subvencionar las futuras investigaciones de una persona que lo solicite a un organismo superior, ya sea la Institución relacionada al repositorio ú otra de igual o mayor importancia.

CASO 4: Explorar los registros de uso para los artículos de un autor cuyo origen se correspondan con el país "Argentina".

En ciertas ocasiones podría ser necesario filtrar los registros de uso por diversos criterios para poder examinar aspectos más profundos de los datos registrados a partir del uso.

CASO 5: Explorar todos los registros de uso del repositorio indexados durante la primera quincena del mes de Marzo del año 2017.

Los reportes solicitados en un periodo de tiempo arbitrario podrían ser requeridos en varias ocasiones, como por ejemplo, cuando el alto mando de la Institución vinculada al repositorio solicita un informe de la cantidad de descargas realizadas de la producción del último año.

CASO 6: Exportar los registros de uso correspondientes a los ítems de un autor específico para utilizarlos como fuente de datos en un sistema estadístico externo y calcular estadísticas simples y complejas.

El cálculo de estadísticas simples y complejas, como por ejemplo el cálculo de la media en una serie de valores o el cálculo de la distribución de frecuencias, no forman parte de los objetivos del actual trabajo, por lo que para realizar cálculos sobre los datos indexados en Solr es necesario realizarlo fuera del repositorio en un software dedicado para el cálculo matemático/estadístico como, por ejemplo, MATLAB y R.

Experimentación: Repositorio CIC-DIGITAL

Con motivo de probar el funcionamiento del prototipo implementado, se decidió utilizar como plataforma de experimentación el repositorio institucional de la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires llamada *CIC-DIGITAL* (https://digital.cic.gba.gob.ar), cuyo objetivo (CICBA, 2014) es reunir, registrar, divulgar, preservar y dar acceso público a toda la producción científico-tecnológica y académica de la CIC. Al momento de la escritura de este trabajo, el repositorio contaba con más de 6500 ítems depositados, más de 400 Colecciones, y alrededor de 200 Comunidades, con 4 Comunidades principales: «Centros», «CICBA», «Investigadores en Universidades Nacionales de la provincia de Buenos Aires», y «Otras Instituciones». CIC-DIGITAL es una plataforma basada en el software de repositorios DSpace en su versión 6.2, utilizando la

aplicación XMLUI como punto de acceso principal al repositorio.

Características del prototipo

En línea con los objetivos de este trabajo redactados en el Capítulo 1, el prototipo debería contar con las siguientes características a fin de facilitar su cumplimiento:

- **Fácil exploración de los registros de uso indexados.** Es decir, que el prototipo debería permitir la inspección del contenido del índice Solr *statistics* mediante búsquedas sobre su contenido, sin disponer de conocimientos técnicos propios de Solr, pero sí del significado de los campos indexados en este sistema. La implementación de esta característica facilita el análisis de los datos en crudo que efectivamente se indexan a partir del uso del repositorio.

- **Tiempos de respuesta razonables durante la exploración.** Para que la búsqueda en los datos indexados en Solr sea fluida, se debería garantizar que los tiempos en que se retornen los resultados sean razonables. Estos tiempos estarán influenciados por la gran cantidad de datos que constantemente son indexados (miles de millones de datos), y mientras más datos existan es probable que más tiempo tarde en generarse una respuesta.

- **Múltiples contextos de búsqueda.** Es decir, que el prototipo debería permitir la selección de diversos contextos sobre los que se realizarán búsquedas de registros; en particular, debería implementarse búsquedas por colección, comunidad, o por un conjunto específico de objetos Dspace. Esto permitirá explorar fácilmente el uso realizado sobre ciertos objetos del repositorio que son de interés por el usuario.

- **Exportación de registros**. Es decir, que el prototipo debería disponer de una exportación de registros derivados a partir de una búsqueda específica, y ofrecer variados formatos (csv, json, etc). La finalidad de esta característica es brindar de un conjunto de datos al usuario que lo requiera para poder generar diversas estadísticas de uso en potentes programas estadísticos externos al repositorio.

- **Generación de gráficos.** Es decir, que el prototipo debería disponer de una sección de graficación para permitir visualizar diferentes características o variables de los registros resultantes de una búsqueda. Estos gráficos no deberían representar un cálculo estadístico complejo, simplemente deberían facilitar un paneo de diversos aspectos del conjunto de registros explorados.

- **Ofrecer diversos puntos de configuración.** Ser configurable en la mayoría de sus aspectos (opciones de filtrado, opciones de exportación, opciones de ordenamiento de resultados). Esto permitirá una flexibilidad importante a la hora de personalizar la herramienta acorde a las necesidades de cada Institución que administra un repositorio DSpace.

Alternativas para la implementación del prototipo

Al momento de comenzar con la realización del prototipo se analizaron una serie de consideraciones, a partir de las cuales se determinaría el camino a seguir para su implementación:

1. Sobre qué versión del software DSpace se comenzaría a implementar el prototipo.
2. Qué tecnología o combinación de tecnologías se haría uso para la implementación del prototipo.

El primer punto de las anteriores consideraciones resultó de mayor importancia, ya que dependiendo de la elección resultante entre las versiones candidatas se iban a derivar las tecnologías a utilizar para la realización del prototipo. Entre estas versiones se encontraban:

- *DSpace 6.X* («DSpace Release 6.0 - DuraSpace», s. f.), es decir, DSpace en la rama estable de su versión 6, lanzada en el mes de Octubre del año 2016.
- *DSpace 7.X* («DSpace 7 Working Group - DuraSpace», s. f.), es decir, DSpace en la rama de desarrollo de su versión 7, la cual al momento de escritura de este trabajo todavía no había sido lanzada oficialmente como una nueva versión estable del software.

Para decidir cuál de estas versiones era conveniente utilizar se analizaron las ventajas/desventajas de las mismas:

- *DSpace v7* introduce cambios importantes en la estructuración de la capa de datos y de la vista en DSpace, cuya meta principal es crear una nueva Interfaz de usuario (UI) única para DSpace que implemente todas las funcionalidades actualmente disponibles en las UI de DSpace v6: JSPUI y XMLUI. En particular, el lanzamiento de esta nueva versión de DSpace se concentra en dos características principales (Donohue, Knowles, Lowel, & Bollini, 2017): (1) Una nueva interfaz de usuario basada en *Angular* en su versión más reciente (para reemplazar XMLUI y JSPUI) y (2) una API REST mejorada/refactorizada mediante el uso de tecnologías *Spring Boot* y *Spring Data Rest,* y los estándares más utilizados en REST (HATEOAS, HAL format, ALPS, JSON-Schema) como soporte de esta capa mejorada. Avanzar hacia el uso de estas tecnologías y prácticas es de gran importancia en DSpace como aplicación web para repositorios, debido a que son tecnologías y prácticas que hoy en día están vigentes y son utilizadas en múltiples proyectos web. La gran *desventaja* de esta opción es que al día de hoy aún está en un periodo temprano de desarrollo y sin una fecha específica de lanzamiento, por lo que desarrollar un prototipo sobre esta versión en constante cambio no resultaría factible.

- *Dspace v6* es la última versión estable del software y es utilizada actualmente en la plataforma experimental CIC-Digital; si la implementación del prototipo fuera sobre esta versión del software, entonces podría ser puesto en producción fácilmente ya que no habría que considerar casi ninguna migración en el código fuente ni migración en los datos del repositorio. Una de las desventajas de esta opción es que, al lanzarse la nueva versión del software DSpace, la sección entera del código que implementa la vista (XMLUI) así como algunas porciones del código API o core

del software que determina la lógica interna tendrían que ser re-implementados o migrados para garantizar el funcionamiento en la versión 7 de la plataforma. Otra desventaja de esta opción es que algunas de las tecnologías que DSpace utiliza son muy antiguas; por ejemplo, DSpace utiliza Apache Cocoon para la generación de la construcción de la vista ante una petición HTTP, y su última versión estable (2.2) fue lanzada el 15 de Mayo de 2008. Como detalle adicional, es necesario mencionar que ésta versión (DSpace-CIC versión 6) incluye cambios propios agregados a lo largo del tiempo de vida del repositorio CIC-Digital.

Luego de analizar las ventajas y desventajas anteriores, finalmente se decidió desarrollar el prototipo sobre la versión 6 de DSpace en CIC ya que, aunque las tecnologías y prácticas propuestas en la versión 7 son más atractivas que las opciones en la versión 6, el primero aún se encuentra en una etapa temprana de implementación (aunque bastante activa a nivel desarrollo), por lo que desarrollar un prototipo sobre una versión aún en transición no es factible. Esto se debe, entre otras cosas, a que no existen garantías de que las configuraciones y prácticas propuestas sean las finales al momento de lanzar la versión definitiva de DSpace 7.

Módulo Discovery

Con el motivo de reutilizar módulos ya existentes en DSpace para la implementación del prototipo, se analizó el módulo encargado de comunicarse con el core de búsqueda 'search' en Solr, a fin de determinar si podía ser adaptado mediante algunos cambios para satisfacer la necesidad de explorar y consultar el core «statistics» en Solr.

Como se mencionó previamente, Discovery («Discovery - DuraSpace», s. f.) es el nombre del módulo que permite la búsqueda de contenidos en los repositorios DSpace. Este módulo habilita la búsqueda facetada y la exploración del repositorio mediante distintos metadatos de los ítems que lo componen. El módulo se divide en tres partes: las clases que implementan vista o interfaz de usuario desde donde se hace la búsqueda, las clases que implementan la lógica de funcionamiento y que conforman el núcleo de *Discovery*, y las clases que implementan las configuraciones del módulo.

Interfaz de usuario Discovery

Los principales componentes de vista que implementan la interfaz de usuario y permiten la utilización de Discovery son las clases de código fuente Java alojadas en el paquete *org.dspace.xmlui.aspect.discovery,* visualizadas en la Figura 4.1; existen otras interfaces que generalmente no implementan vistas aunque sí definen puntos de acceso específicos que mediante el uso de parámetros permiten realizar búsquedas, como por ejemplo la interfaz de búsqueda para OpenSearch («Configuration Reference - DuraSpace», s. f.) y una interfaz de búsqueda JSON.

En particular, *SimpleSearch* es el componente que implementa la interfaz más comúnmente utilizada por los usuarios para interactuar con Discovery, agregando las distintas funcionalidades de búsqueda, filtrado, paginación y ordenamiento de resultados. Según la configuración que tenga el módulo, las opciones de filtrado pueden darse sobre

distintos campos definidos en Solr, y el ordenamiento mediante diversos metadatos relativos a los objetos resultantes de la búsqueda. También existe otro componente dentro de este paquete llamado *SidebarFacetsTransformer* que se encarga de habilitar distintas opciones de faceting definidas a partir de la configuración del módulo y que, como fue visto en la sección «Capítulo 3 - Solr», define agrupamiento de objetos según algunas de sus características. Desde la perspectiva del usuario, la búsqueda facetada (también llamada navegación con facetas, navegación guiada o búsqueda paramétrica) divide los resultados de búsqueda en múltiples categorías, generalmente mostrando recuentos para cada uno, y permite al usuario «profundizar» o restringir aún más sus resultados de búsqueda en esas facetas. Por ejemplo, un facet por autor agrupa todos los ítems (resultantes de la búsqueda) de un determinado autor en una única categoría de autor dentro del facet.

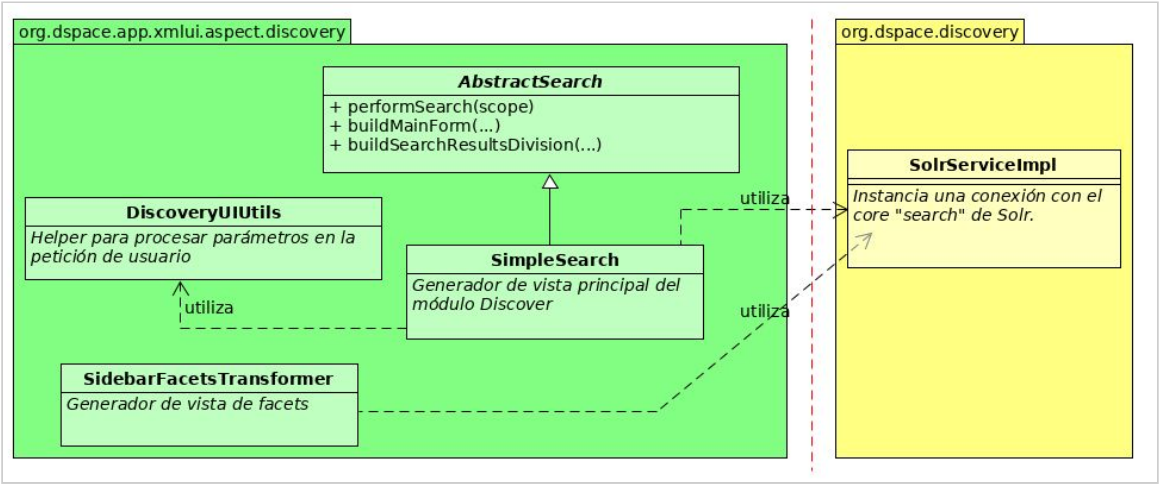


Figura 4.1 - Paquete de interfaz de usuario en Discovery

Algunos de los componentes de la interfaz de Discovery se visualizan en la Figura 4.2. A través de la caja de búsqueda se permiten buscar términos arbitrarios definidos por el usuario sobre los metadatos de un ítem. Asimismo, se pueden definir algunos filtros avanzados en la sección de filtros para acotar la búsqueda: filtro por exactitud («Es»), filtro por coincidencia de al menos un término («Contiene»), filtro por valor de autoridad («Authority»), y además filtros que implican la negación de todos los anteriores. Las opciones laterales permiten realizar el faceting de los resultados, de tal forma de refinarlos por su valor de Autor, Materia o Palabra Clave (en este ejemplo). Además, en la sección de paginación el usuario puede definir para la lista de resultados la cantidad de resultados a ver por página así como el orden (ascendente o descendente) en que desea visualizar los resultados. Luego de que el usuario realice una consulta, estos componentes de la vista se comunican con las clases núcleo de *Discovery* para consultar al core «search» en Solr.

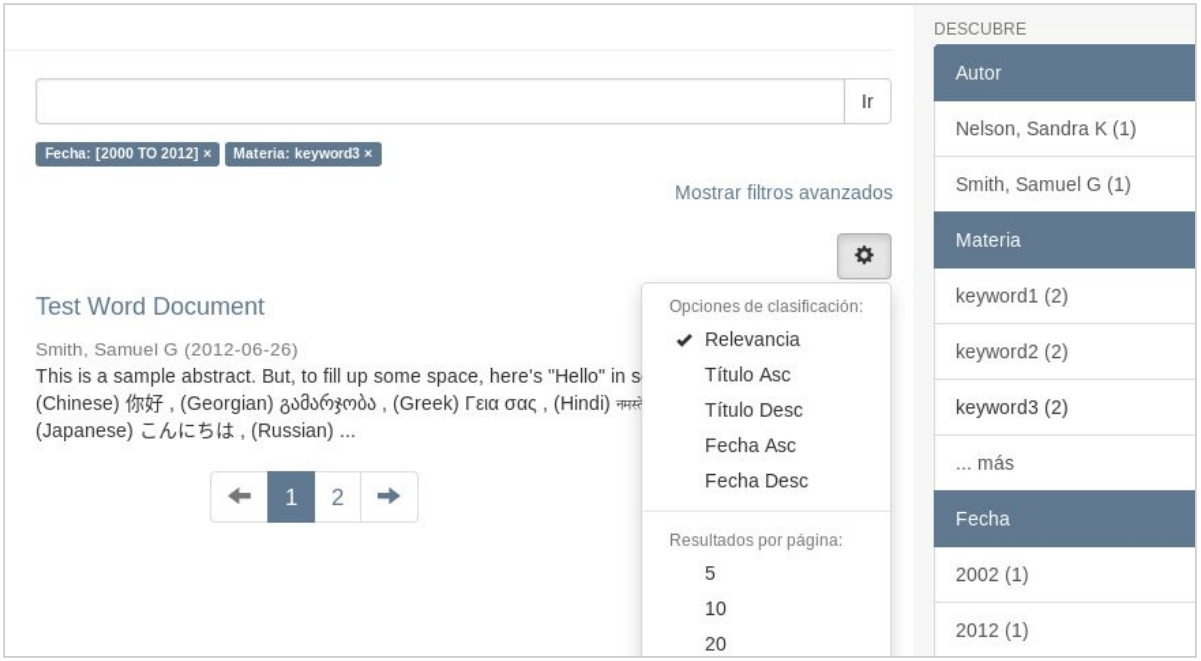


Figura 4.2 - Interfaz de usuario SimpleSearch (Discovery)

Núcleo Discovery

La capa de vista en *Discovery* se comunica con las clases que conforman el núcleo de *Discovery* para transformar la petición del usuario en una consulta específica con una sintaxis que Solr entiende, y transformar luego los resultados devueltos por Solr en algo que DSpace pueda manejar para retornar la respuesta al usuario. Esta lógica de comportamiento se define en el paquete *org.dspace.discovery* de DSpace, y el principal componente del paquete es el servicio *SolrServiceImpl.* Como se observa en la Figura 4.3, este servicio puede ser utilizado desde cualquier punto de la aplicación, y debe implementar dos interfaces[21] específicas: la interfaz *SearchService* que define los métodos de búsqueda que deberá implementar el servicio de búsqueda en Solr, y la interfaz *IndexingService* que define los métodos de indexación a implementar por el servicio para agregar o eliminar contenidos del core «search» en Solr.

Para comunicarse con Sor, DSpace utiliza una librería determinada llamada *SolrJ* («Solrj - Solr Wiki», s. f.): una API que facilita la comunicación entre las aplicaciones Java y Solr. SolrJ oculta muchos de los detalles de la conexión a Solr y permite que DSpace interactúe con Solr con métodos simples de alto nivel a través de la clase *HttpSolrServer*.

DSpace utiliza los componentes *DiscoveryQuery* y *DiscoveryResult* para construir y encapsular las peticiones y las respuestas retornadas hacia y desde Solr, respectivamente. Estos componentes son utilizados por la vista para permitir la visualización de los contenidos resultantes de una búsqueda de usuario. En particular, *DiscoveryQuery* acepta distintos parámetros o propiedades y mediante su manipulación permite configurar la

[21] Un interfaz es una lista de acciones que puede llevar a cabo un determinado objeto. En un interfaz sólo existe el prototipo de una función (nombre del método, parámetros entrantes, tipo de retorno), dejando la definición del código de su comportamiento para las clases que la implementen.

consulta final que se realizará a Solr a través de *SolrServiceImpl*. Entre estas propiedades se pueden encontrar:

- la propiedad «query» que permite definir un término de búsqueda libre,
- la propiedad «filterQueries» que permite definir uno o más filtros avanzados en Solr de tal forma de reducir el conjunto de objetos sobre los que realizar una búsqueda,
- las propiedades «maxResults» y «start» que permite configurar la cantidad máxima de objetos que Solr debe retornar por petición y el índice de comienzo dentro de la lista total de resultados (permitiendo de este modo la *paginación* de resultados),
- la propiedad «sortField» que permite definir un campo de ordenamiento específico,
- y algunas propiedades más.

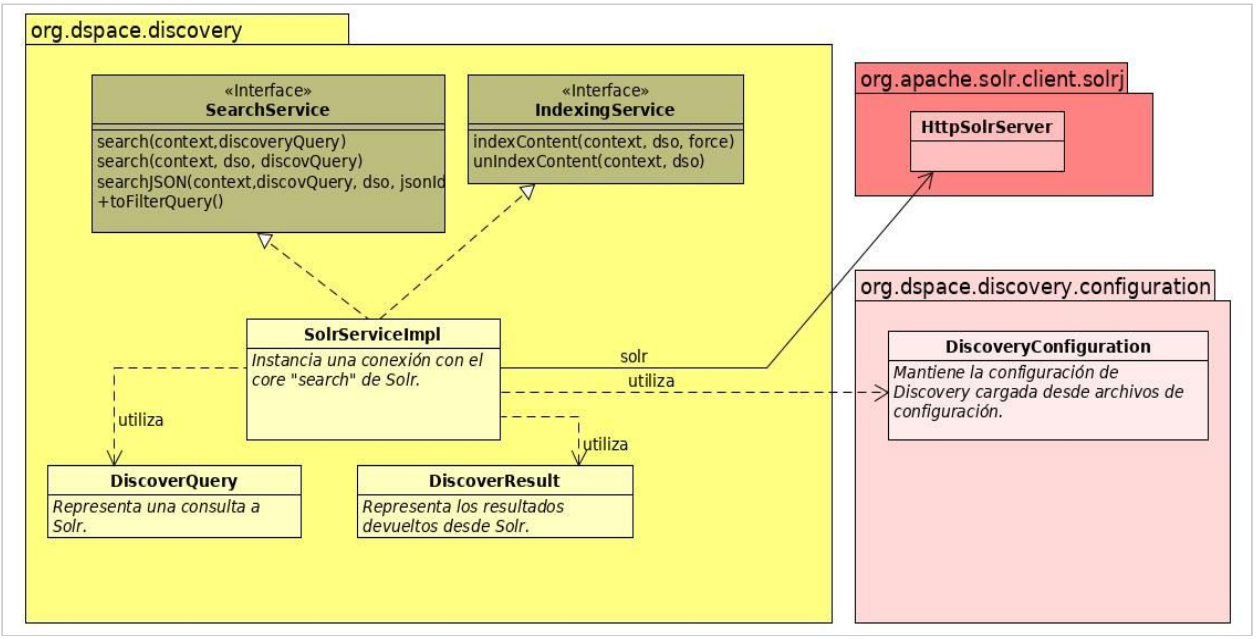


Figura 4.3 - Paquete de clases núcleo en Discovery

Por último, para que *SolrServiceImpl* funcione correctamente debe utilizar una configuración específica, como se verá en la siguiente sección.

Configuración Discovery

Para determinar los elementos que debe mostrar la interfaz de usuario de Discovery es necesario modificar algunos de los archivos de configuración que el módulo provee. Como se observa en la Figura 4.4, en el paquete *org.dspace.discovery.configuration* se definen los componentes de configuración en *Discovery*, y el componente principal es la clase *DiscoveryConfiguration*. Esta clase es inicializada por el framework *Spring* (https://spring.io/) mediante la técnica de *dependency injection*[22] («Spring Framework - Core Technologies», s. f.) a partir de beans[23] definidos en los archivos de configuración XML de Discovery.

[22] *Dependency Injection* es una técnica donde un objeto provee las dependencias a otro objeto.
[23] Un bean es un objeto instanciado, ensamblado y gestionado por un contenedor Spring utilizados para realizar *dependency injection*.

La configuración de *Discovery* («Discovery - DuraSpace», s. f.) se encuentra separada en 2 archivos:

- Configuración General: *dspace/config/modules/discovery.cfg*,
- Configuración de la interfaz: *dspace/config/spring/api/discovery.xml.*

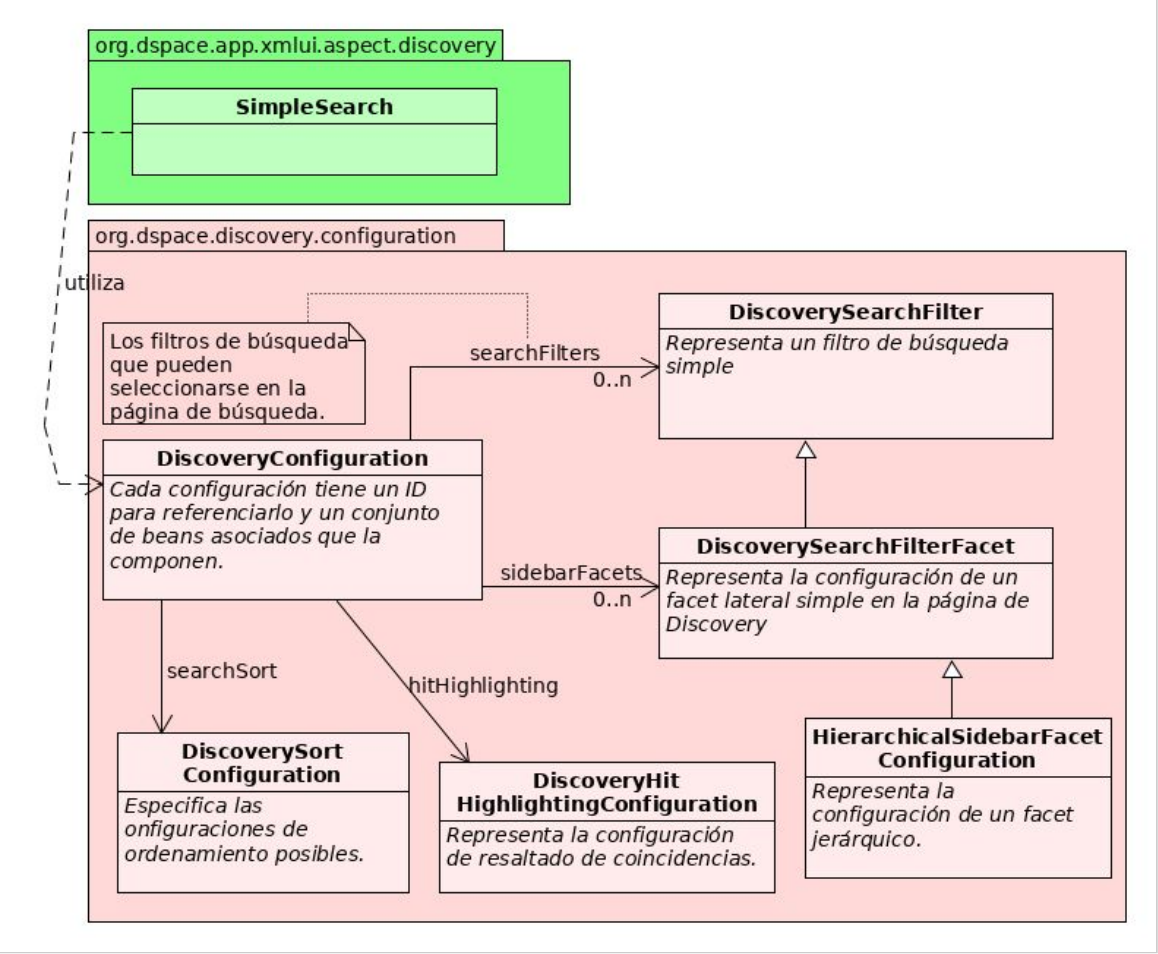


Figura 4.4 - Paquete de clases de configuración en *Discovery*

Como se mencionó, *Discovery* es inicializado por *Spring* mediante los beans definidos en el archivo *discovery.xml*, y posteriormente, la vista será renderizada por la interfaz web seleccionada (XMLUI o JSPUI) a partir de estas configuraciones. Algunos de estos beans se describen en la Tabla 4.1.

DiscoverySearchFilter	Define qué campos de metadatos específicos de los objetos en DSpace deben estar habilitados para aplicar filtros de búsqueda sobre ellos.
DiscoverySearchFilter Facet	Define qué campos de metadatos se deben ofrecer como opciones de facetado en la barra lateral contextual.
DiscoverySort Configuration	Define las opciones de ordenamiento que se habilitan al usuario sobre los resultados de búsqueda.
DiscoveryHitHighlightin	Define qué campos de metadatos pueden contener resaltado

gConfiguration	de coincidencias de búsqueda (*hit-highlighting*).
DiscoveryConfiguration	Agrupa configuraciones de facets, filtros de búsqueda, opciones de ordenación y otras más.

Tabla 4.1 - Algunos componentes del archivo *discovery.xml*

Discovery como base del prototipo

Luego de analizar los componentes del módulo *Discovery*, se decidió utilizarlo como base para la implementación del prototipo por las siguientes razones:

- El módulo existe desde la versión 1.7 del software («Discovery - DuraSpace», s. f.), por lo que en las sucesivas versiones se fueron realizando varias correcciones y mejoras en su funcionamiento por una extensa comunidad de desarrolladores.
- Su interfaz de usuario es mayormente intuitiva y permite explorar los contenidos del repositorio de forma eficiente.
- Su funcionalidad de base para la exploración del core «search» de Solr podría ser extendida o adaptada para usarlo sobre el core «statistics» sin mucho esfuerzo.
- La flexibilidad de su configuraciones mediante el uso de beans permite una amplia posibilidad de adaptarlo a las necesidades de los repositorio en relación a la búsqueda de contenidos. Podrían llegar a crease beans específicos en el caso de querer implementar alguna funcionalidad puntual para la búsqueda.

Capítulo 5 | Implementación

Luego del análisis realizado en el *Capítulo 4* acerca de las opciones de tecnologías base para la realización del prototipo, se concluyó que la mejor alternativa al momento del desarrollo de este trabajo era la utilización de DSpace en su versión 6 y todas las tecnologías y herramientas derivadas de ésta: Apache Cocoon, Spring Framework, XSLT, Javascript, y el módulo *Discovery* implementado por la plataforma, entre otras. Considerando lo anterior, en el presente capítulo se procederá a explicar el desarrollo realizado para la implementación del prototipo mediante el uso de esta configuración de tecnologías.

Funcionamiento del prototipo

A grandes rasgos, el prototipo funciona de una manera muy similar a como lo hace *Discovery*. Sin embargo, como se observa en el Figura 5.1 se tuvieron que realizar algunas modificaciones para que DSpace utilice el core «statistics» en vez del «search»:

1. Un usuario del repositorio envía solicitudes a través de la interfaz implementada en XMLUI, la interfaz de DSpace implementada en Apache Cocoon.
2. XMLUI se comunica con el nuevo módulo de estadísticas correspondiente (Statistics-Discovery Module) para resolver la petición de búsqueda.
3. El módulo se comunica con el core «statistics» en Solr a través de la librería SolrJ, la cual envía peticiones a la interfaz HTTP en Solr («Client APIs | Apache Solr», 2017).
4. Por último, la respuesta retorna hasta el usuario a través del tratamiento de los resultados devueltos por Solr.

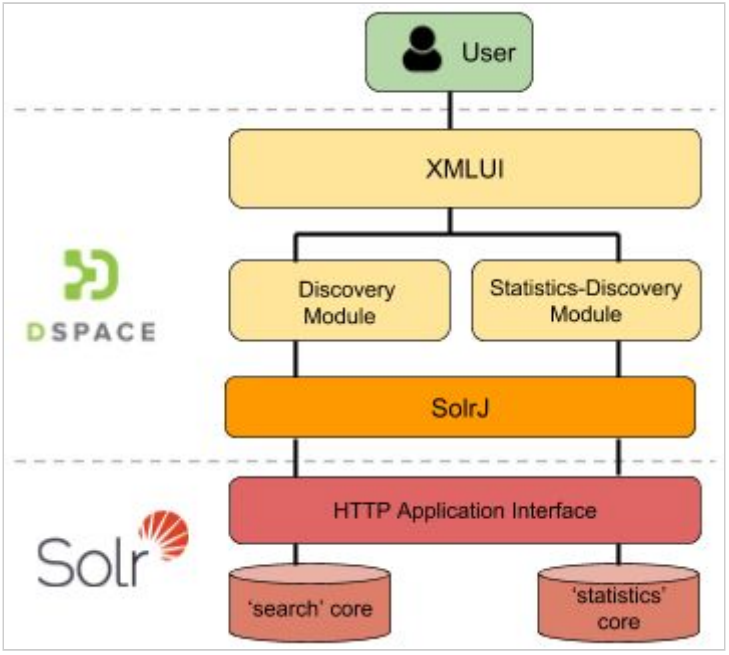


Figura 5.1 - Funcionamiento módulo Statistics-Discovery

Desde la interfaz XMLUI el cliente puede realizar búsquedas, aplicar filtros y facets, solicitar exportación de registros, y generar un conjunto de reportes predefinidos con gráficas. La

URL de acceso al módulo de estadísticas tiene la siguiente forma *http://<hostname>:<port>/<dspace_path>/statistics-discover?<search_parameters>*.

Modelo del prototipo

El modelo del prototipo está basado en el modelo original planteado por Discovery (visto en mayor detalle en el *Capítulo 4* de este trabajo), aunque debido a la fuerte dependencia que existe en varias partes del código con el core «search», fue necesaria adaptarlo para el uso sobre el core «statistics». Con este motivo en mente, se creó un nuevo módulo llamado *Statistics-Discovery*, que reimplementa la mayoría de los componentes de la vista y el core de Discovery para su adaptación, y además agrega algunos componentes de configuración nuevos para extender los existentes.

Como se observa en las Figuras 5.2, 5.3 y 5.4, los paquetes de componentes de la vista y el núcleo con la lógica de Statistics-Discover se mantienen bastante similares a los de Discovery. Sin embargo, aunque son similares a nivel diagrama, la codificación de estos componentes tuvo que readecuarse al nuevo contexto de uso. Una refactorización de Discovery, además de incompatible, hubiera sido inaceptable según las prácticas de desarrollo en DSpace, cuya filosofía es cambiar lo mínimo e indispensable del código fuente a menos que sean grandes aportes (como p.e. la gran refactorización realizada en DSpace 6 debido a la migración en el uso de Hibernate («DSpace Release 6.0 - DuraSpace», s. f.) para concentrar el acceso a la capa de almacenamiento mediante una única herramienta), y generar modificaciones significativas en el módulo Discovery para su utilización conjunta con el core «search» y «statistics» no hubiese sido motivo suficiente para que se acepte tal refactorización. En cuanto a la incompatibilidad mencionada anteriormente, ésta se debe a factores como los siguientes:

I. Discovery fue pensado para indexación en Solr (además de búsqueda), por lo que existía cierta lógica incompatible en este sentido que no podía adaptarse, ya que el módulo Statistics-Discovery no indexa, sólo realiza búsquedas. La indexación es realizada por el módulo *Statistics* (visto en el Capítulo 3).
II. Existen operadores de búsqueda en Solr de Discovery que no tienen sentido en el contexto de *Statistics-Discovery*. Por ejemplo: (1) el operador «authority» de Discovery, que realiza búsquedas sobre los valores de autoridad[24] de los metadatos, no puede aplicarse en *Statistics-Discovery* ya que los fields en «statistics» no usan autoridades; (2) otro caso es el operador «contiene» que busca un término en los campos «tokenizados»[25] de Discovery, mientras que en Statistics-Discovery no existen estos tipos de campos, por lo que la lógica de este operador debería adaptarse.
III. Los resultados esperados por Discovery desde la búsqueda en Solr siempre son objetos DSpace, mientras que en *Statistics-Discovery* son registros de uso en el repositorio, que pueden o no estar vinculados a objetos DSpace.

[24] Una *autoridad* en DSpace es una fuente externa de valores fijos para un dominio dado, cada valor único identificado por una clave.
[25] Los tokenizers en Solr son responsables de fragmentar el valor de un campo en unidades léxicas (p.e. palabras) o *tokens*.

Además de las adaptaciones realizadas, hubo que agregar algunos nuevos componentes. En la *vista* del modelo se agregaron los componentes *StatisticsDiscoveryExporter*, que añade funcionalidad de exportación de resultados de búsqueda en variados formatos, y *StatisticsDiscoveryJSONReport*, que habilita un punto de acceso o endpoint JSON a través del cual se permite la generación de reportes a partir de los resultados de búsqueda; se detallará más sobre éstos en las siguientes secciones. En cuanto al *core* del modelo, se tuvo que hacer una generalización de los resultados encapsulados por Discovery mediante la jerarquía formada entre *GenericDiscoverResult, DiscoverResult y StatisticsDiscoverResult,* de tal forma de poder interpretar los resultados devueltos por 'statistics' como registros de uso en vez de como objetos DSpace. Por último, en la sección de *configuración* del modelo, solamente se agregaron 3 nuevos componentes de configuración: la subclase *ExtendedDiscoveryConfiguration* que extiende la configuración base para que entienda los otros componentes agregados, el componente *StatisticsDiscoveryCombinedFilterFacet* que permite utilizar 2 o más campos del core 'statistics' en una misma opción de facetado, y el componente *DiscoveryMultipleSearchFilter* que permite definir múltiples campos de búsqueda de 'statistics' en un único bean de configuración en vez de utilizar un DiscoverySearchFilter individual por cada campo de búsqueda a agregar como se hace en Discovery.

El motivo de la creación del componente *StatisticsDiscoveryCombinedFilterFacet* se debe a una situación particular que se dá en el core 'statistics'. Por ejemplo, los campos *type* y *scopeType* (definido en el Capítulo 3) representan los tipos de DSO que determinan el scope de los registros estadísticos del tipo «view» y «search» respectivamente; si se quisiera hacer un facet único a partir de estos dos campos distintos, entonces hay que utilizar este tipo de filtro.

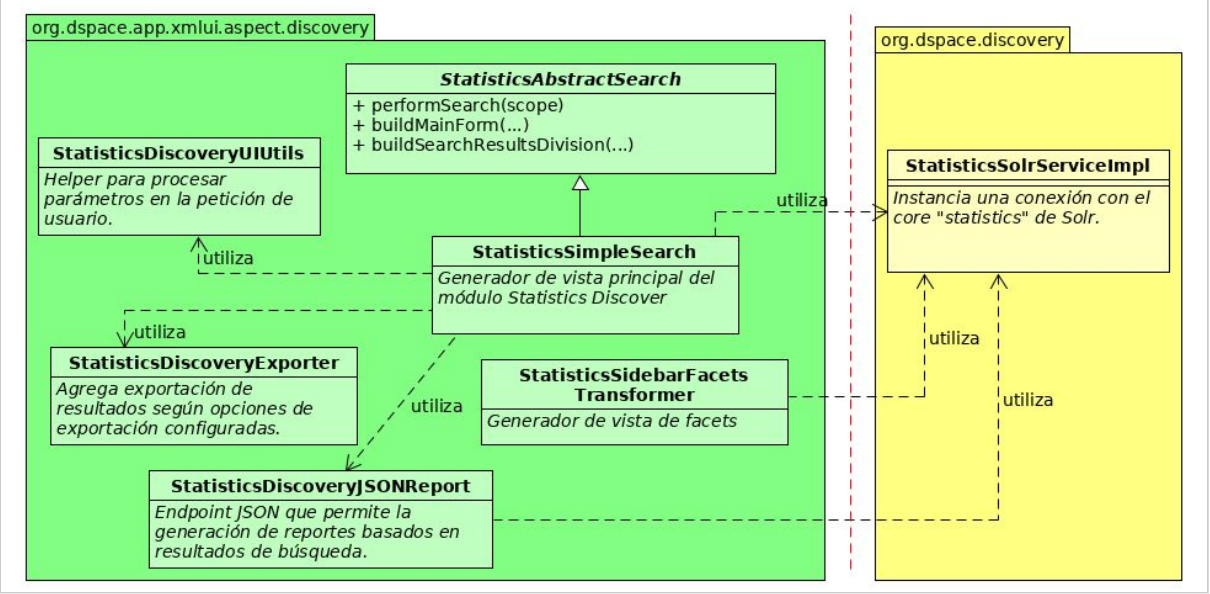


Figura 5.2 - Modelo de la vista en *Statistics-Discovery*

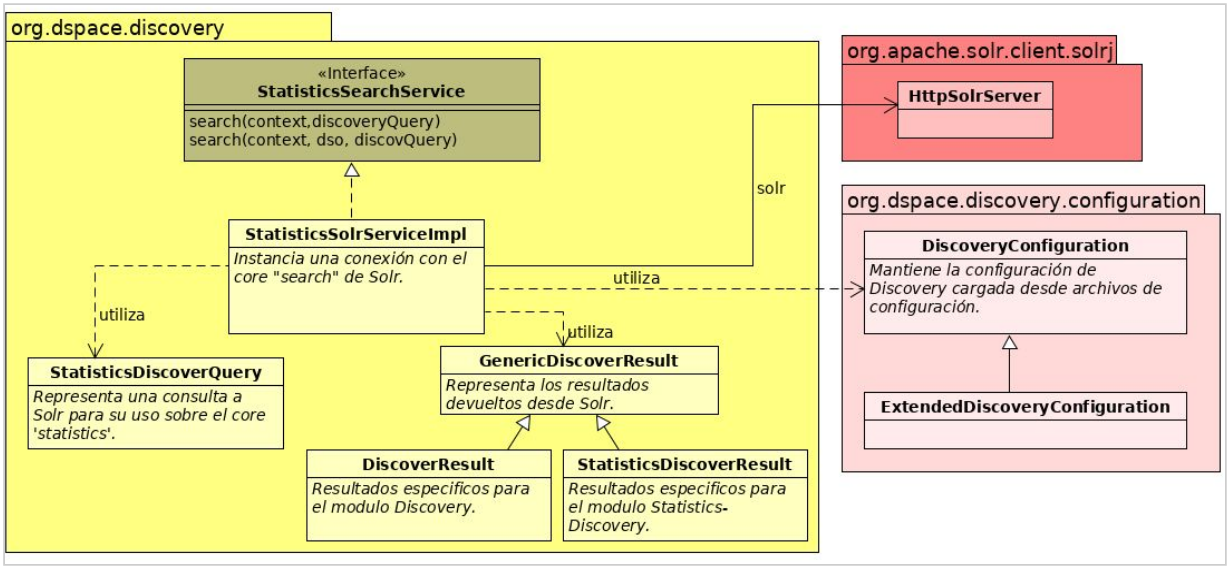


Figura 5.3 - Modelo del núcleo en *Statistics-Discovery*

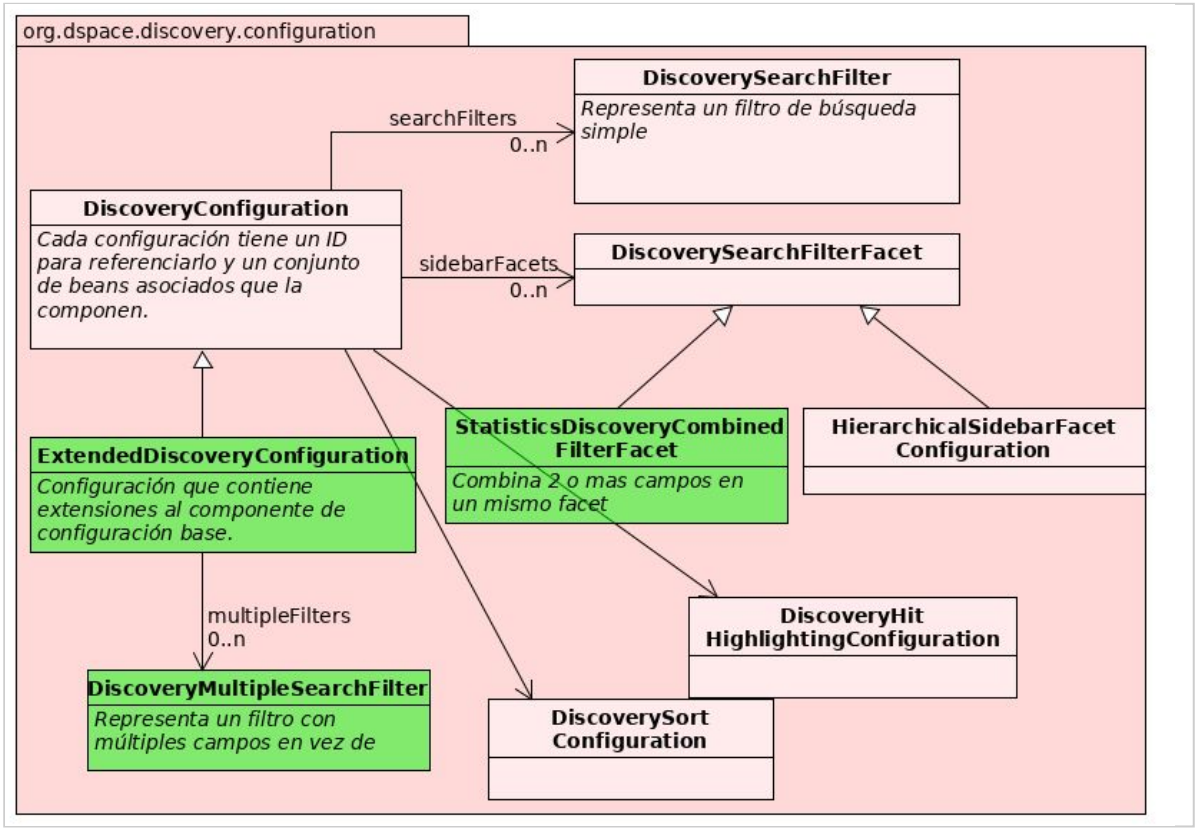


Figura 5.4 - Modelo de configuración de *Statistics-Discovery*

Extensiones creadas

En esta sección se explicará en mayor detalles las extensiones más destacadas que fueron realizadas al módulo original de *Discovery*.

Scopes

Se definió un *esquema selección de scopes*. Un *scope* en *Statistics-Discovery* es el contexto sobre el que se va a realizar una búsqueda; este contexto se define a nivel de objetos DSpace, es decir, que dado un conjunto A de objetos DSpace solamente se buscarán los registros de uso para los objetos en este conjunto. Este scope se puede especificar de varios modos:

- Mediante el prefijo «/handle/XX/YY» antes del componente «/statistics-discover» en la ruta al módulo Statistics-Discovery. Este tipo de scope se llama *scope fijo.* Determina que el scope será el objeto DSpace asociado con el handle[26] XX/YY en el repositorio.
- Mediante el parámetro «scope» en la ruta al módulo Statistics-Discovery. Este tipo de scope se llama *scope variable*. El funcionamiento es igual al scope fijo.
- A partir de los objetos resultantes de una búsqueda en Discovery mediante el *parámetro* «*discovery_query*» en la ruta al módulo *Statistics-Discovery*. Mediante este parámetro los objetos que delimitan el contexto de búsqueda en el módulo son los objetos resultantes de la búsqueda *Discovery* y sus descendientes (p.e. los descendientes de una comunidad son sus colecciones); opcionalmente, se puede indicar que no se incluyan los descendientes en el scope mediante el parámetro «discovery_scope_no_hierarchical».

Exportación de resultados

Se creó un mecanismo que permite la exportación de resultados de búsqueda en *Statistics-Discovery* a partir de diversos formatos de exportación definidos. Como se observa en el modelo de la Figura 5.6, se implementaron dos estrategias de exportación por defecto: una estrategia para la exportación en formato CSV y otra para el formato JSON. El modelo puede ser extendido para la implementación de nuevas estrategia de exportación, y para esto debe crearse una subclase de la clase abstracta *StatisticsExportStrategy*. Además, cada estrategia de exportación debe implementar el método *export(),* que recibe un conjunto de resultados de búsqueda devueltos por Solr y retorna un archivo con los registros a devolver al usuario en el formato de exportación solicitado.

Desde la interfaz de usuario (mostrada en la Figura 5.5), el componente que agrega las opciones de exportación es el *StatisticsSimpleSearch* (visto anteriormente en este capítulo), agregando formularios que al enviarse se comunican con el componente *StatisticsDiscoveryExporter:* componente central que maneja el proceso de exportación al formato solicitado. Además de realizar exportaciones, este último componente agrega la capacidad de transformación de los valores en los campos de los registros que conforman los resultados de búsqueda. La transformación de estos resultados se debe a una serie de componentes que implementan la interfaz *StatisticsResponseTransformers,* mediante la implementación de los siguientes métodos:

[26] El *handle* es un identificador persistente para un recurso en la web, es decir, es una URL que no varía aunque la página cambie de ubicación. En DSpace, cada ítem, colección y comunidad tiene un handle asociado.

- *beforeQuery()*: método hook utilizado para realizar alguna operación antes de realizar la consulta a Solr (por ejemplo, setear el tipo de respuesta que queremos que retorne, evitar la impresión de ciertos campos, etc)
- *afterQuery()*: método hook utilizado para realizar modificaciones sobre el resultado devuelto a partir de la consulta a Solr, como la aplicación de transformaciones en el valor de un campo (p.e. en vez de retornar el código de país «AR», se retorna el valor «Argentina»).

Figura 5.5 - Opciones de exportación de resultados

La herramienta viene con una sencilla transformación implementada por el componente *StatisticsCommonResponseTransformers*, el cual modifica el valor del campo 'type' de los registros (que originalmente es una constante numérica que representa el tipo de objeto asociado) y lo convierte a la cadena de texto BITSTREAM, ITEM, COLLECTION, o COMMUNITY, según corresponda.

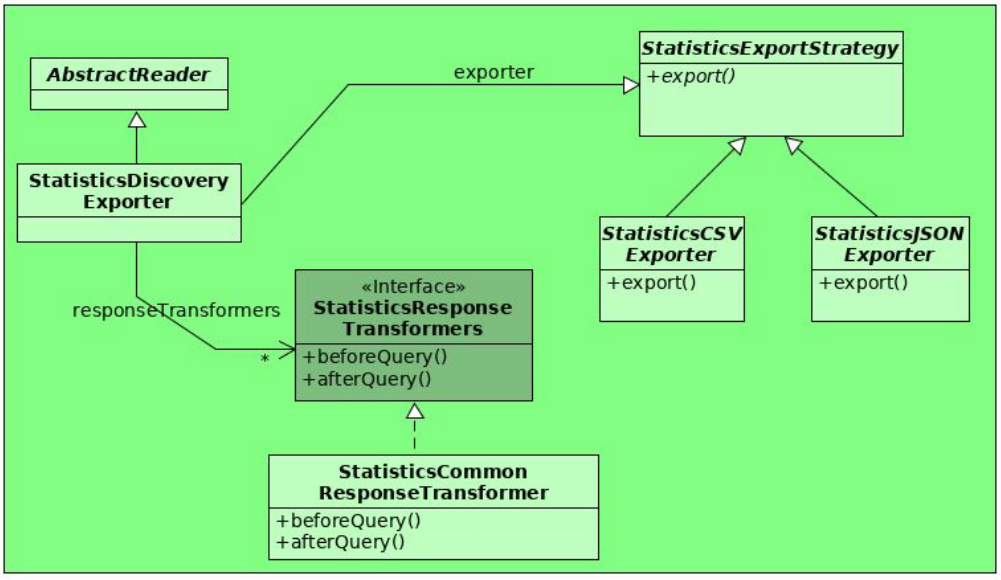


Figura 5.6 - Mecanismo de exportación en *Statistics-Discover*

Por último, se agregó un archivo de configuración llamado *statistics-discovery.cfg*, desde el que se puede configurar y declarar qué formatos de exportación están habilitados y qué transformadores se aplicarán al conjunto de resultados de búsqueda a exportar.

Generación de reportes y graficación

Con el motivo de facilitar el entendimiento de los datos expuestos a través del módulo Statistics-Discovery, se creó una sección en la interfaz de usuario que permite la generación de reportes a partir de un conjunto de reportes predefinidos. Para posibilitar ésto, se habilitó un endpoint de consulta[27] implementado por el componente *StatisticsDiscoveryJSONReport* (ver Figura 5.2), que mediante ciertos parámetros recibidos retorna un archivo en formato JSON acorde al tipo de reporte solicitado. La *población de datos* utilizada para generar los reportes está restringida al conjunto de resultados retornados a partir de la búsqueda en el módulo, es decir, si una búsqueda retornó 100 registros como resultado entonces el reporte se generará a partir de esos 100 registros.

El endpoint de consulta JSON tiene la forma «/*statistics-discover /report/json/<tipo_de_reporte>?<parámetros*>», donde el tipo de reporte es:

- **reporte de una variable** (*onevar*): permiten realizar reportes sobre algún campo específico del core «statistics», generando un faceting sobre el campo solicitado. Por ejemplo: «cantidad de registros categorizados por countryCode».
- **reporte de una variable restringida** (*twovarsonefixed*): ídem a los reportes de una variable, pero limitando el faceting de la variable principal por el valor de otra variable o condición secundaria. Por ejemplo: «cantidad de registros categorizados por countryCode, pero sólo para los registros de descargas de bitstreams». En el core «statistics» los tipos de eventos de uso se determinan por el campo «statistics_type», y las descargas tienen valor «statistics_type = 2».

Los parámetros posibles son:

- *count_of:* determina la variable principal utilizada para generar el reporte acumulado.
- *by:* determina la condición secundaria que limita los registros incluidos en el reporte.
- *timelapse:* parámetro opcional para indicar que se quiere generar un reporte acumulado segmentado por un período o lapso de tiempo específico. Los valores posibles son «*month*» o «*year*», e indican un período mensual o anual, respectivamente.
- *mincount:* parámetro opcional que determina el valor mínimo que cada categoría debe tener para ser incluido en el reporte. Por ejemplo: si se desea generar un reporte de «cantidad de registros por ciudad» con aquellas ciudades que tienen más de 100 registros acumulados, entonces debe utilizarse éste parámetro.

Excepto para el último de los parámetro, los valores aceptados por el resto de éstos ya están preestablecidos desde la aplicación, es decir, solo pueden utilizarse ciertos valores cerrados y no pueden ser valores libres.

Por último, como se introdujo al comienzo de esta sección, el componente *StatisticsSimpleSearch* agrega en la vista del módulo un formulario para la generación de estos reportes, mostrado en la Figura 5.7. Luego de que el usuario haya seleccionado el

[27] Un endpoint es el punto de interacción con un servicio montado en un servidor web.

tipo de reporte que desea, el módulo se comunica mediante AJAX[28] con el endpoint JSON enviando los parámetros correspondientes. Cuando el *StatisticsDiscoveryJSONReport* termina de procesar la petición y realizar la consulta correspondiente a Solr, se retorna el reporte correspondiente al cliente del usuario. Finalmente, luego de recibir el reporte, se genera una gráfica utilizando la librería javascript **c3.js** (http://c3js.org): c3.js está construida sobre la librería *d3.js* (https://d3js.org/) y ofrece una API muy sencilla de utilizar (en comparación a d3.js) con la que se puede realizar variados tipos de gráficas utilizando pocas líneas de código y una amplia cantidad de opciones de personalización a disposición. Mediante esta librería, el módulo genera un gráfico de barras cuando el reporte solicitado no especifica rango de tiempo, como por ejemplo: un reporte de la «cantidad de descargas de bitstreams por país»; en caso contrario, se genera un gŕafico de líneas mostrando el despliegue de cantidades acumuladas a lo largo del lapso de tiempo solicitado, como por ejemplo: un reporte de la «cantidad de descargas de bitstreams por país registradas por año».

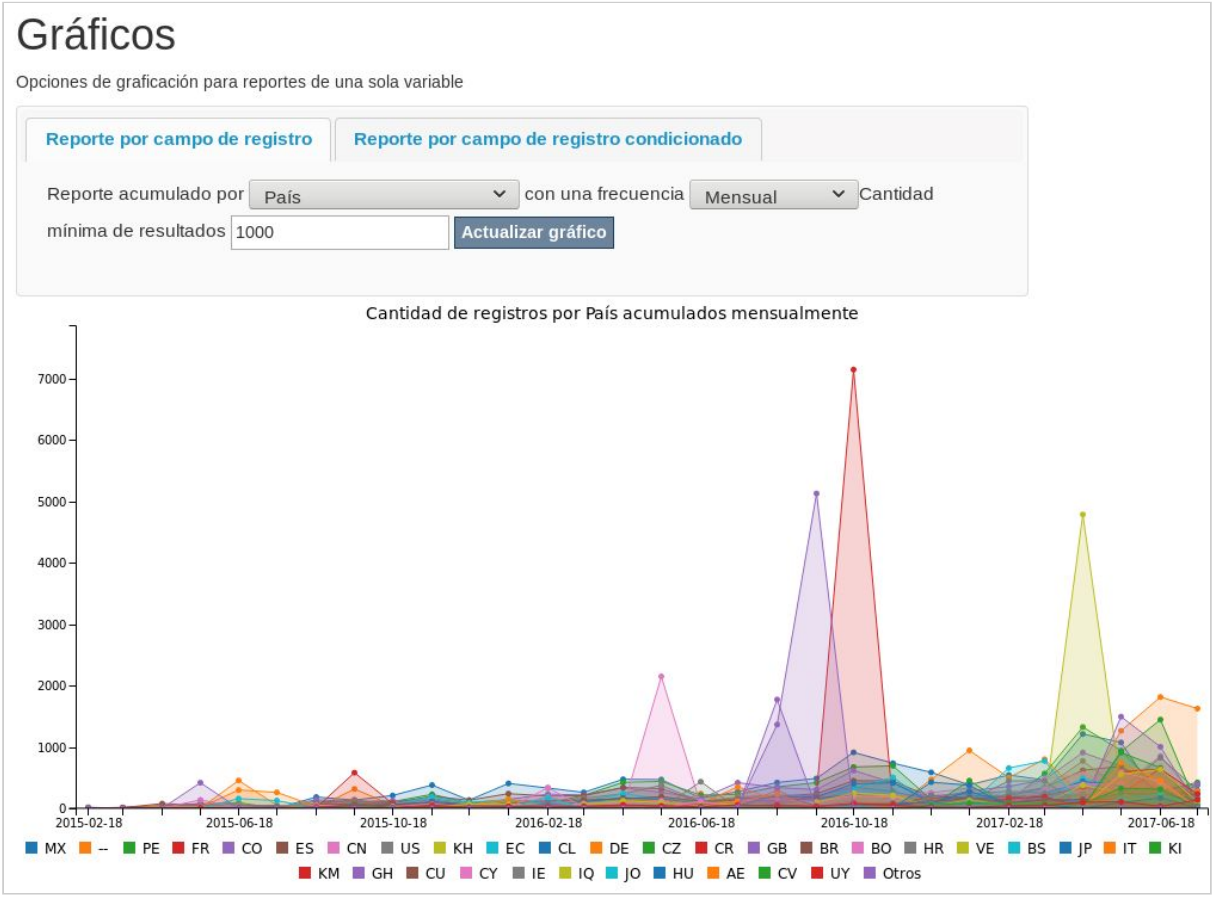


Figura 5.7 - Generación de reportes y gráficas

Filtros y facets

En la sección de filtros avanzados del módulo (visualizada en la Figura 5.8) se agregaron diversas opciones de filtrado, y entre los campos configurados sobre los que filtrar se encuentran los siguientes: IP (*ip*), Código de país (*countryCode*), Tipo de

[28] AJAX es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene una comunicación asíncrona con el servidor en segundo plano.

estadística (*statistics_type*), Tipo de objeto DSpace (campos *scopeType* y *type*), Código de Continente (*continent*), Ciudad (*city*), Agente de usuario (*userAgent*), y Referrer (referrer). La opción «Tipo de objeto DSpace» está configurada como un filtro combinado desde el archivo discovery.xml, es decir, que en un único filtro se combinan 2 o más campos del core statistics (ver Tabla 3.3) sobre los que aplicar una operación de filtrado (ver Figura 5.4).

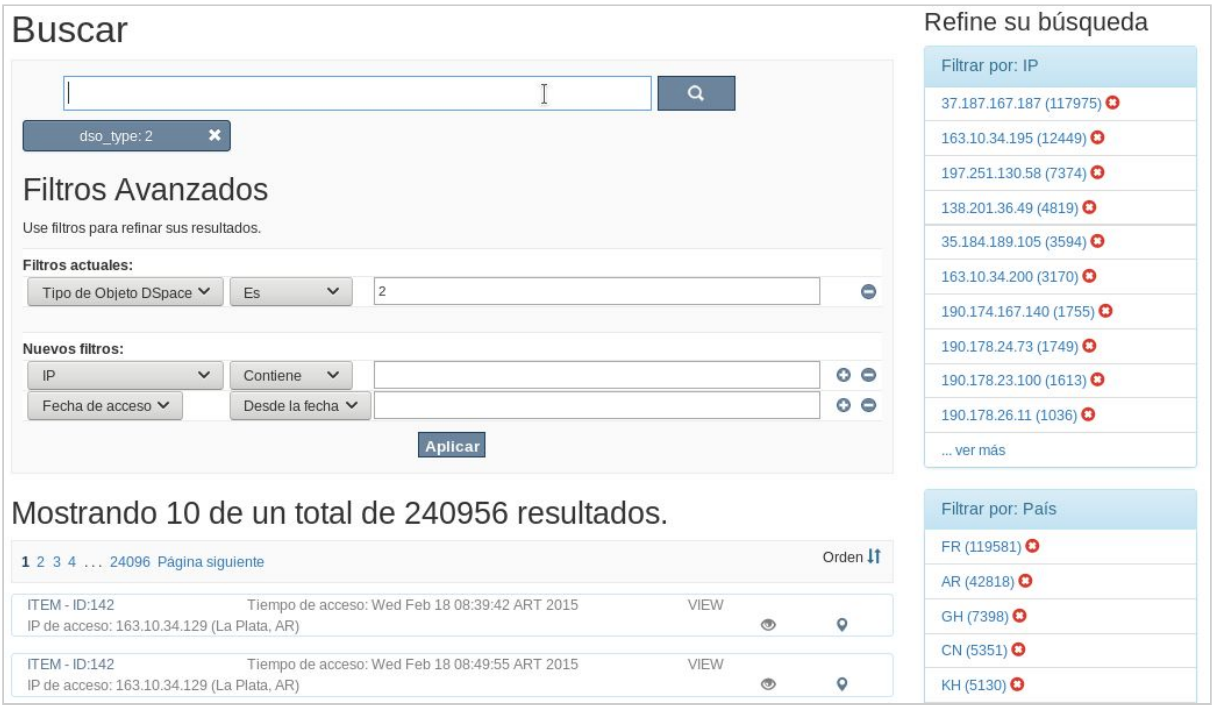


Figura 5.8 - Vista principal del módulo Statistics-Discovery

Además, se agregó la posibilidad de filtrado para los campos de tipo fecha , es decir, los campos que son indexados siguiendo el estándar ISO 8601. El único campo configurado de esta forma es el filtro «Fecha de acceso» (*time*), como se muestra en la Figura 5.9. Para estos tipos de filtros también se definieron algunos operadores especiales: los operadores (1) «Desde la fecha» y (2) «Hasta la fecha», que permiten restringir el conjunto de registros mediante un rango de fecha (una fecha de comienzo y/o una fecha de fin respectivamente), y los operadores (3) «Es» y (4) «No es» que permiten buscar por la coincidencia exacta o la exclusión de una fecha dada. Para asistir a la selección y la escritura de la fecha a especificar, se utilizó un plugin javascript llamado *TimePicker*[29], que agrega a la vista del módulo un selector gráfico con forma de calendario, y permite seleccionar al usuario de forma rápida y sencilla una fecha y una hora determinada. Este plugin se encuentra configurado para guardar la fecha seleccionada acorde al formato ISO 8601.

[29] https://github.com/trentrichardson/jQuery-Timepicker-Addon

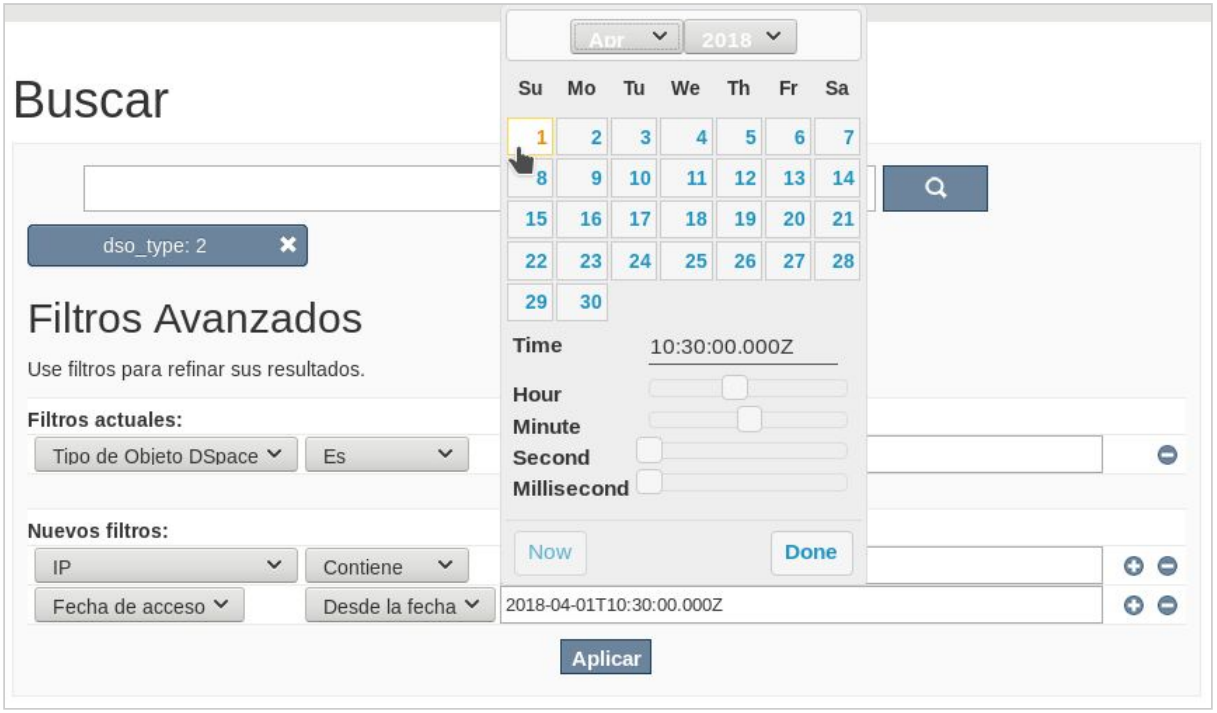


Figura 5.9 - Filtros de fecha en el módulo Statistics-Discovery

En cuanto a los facets del módulo, en el archivo discovery.xml se configuraron las siguientes opciones: *IP, Código de país, Tipo de registro, Tipo de objeto Dspace, y Fecha de acceso*. Los primeros tres facets son simples de tipo textual, el facet «Tipo de objeto DSpace» se corresponde con el filtro combinado descrito en los párrafos anteriores, y el facet "Fecha de acceso" es un facet de tipo date que muestra categorías por año, como por ejemplo, la categoría «2017-2018» permite refinar la búsqueda por los registros del año 2017. Estas opciones de facet permiten dar un pantallazo de las categorías de datos existentes entre los millones de registros del índice «statistics», de tal forma de realizar un sencillo análisis a simple vista (aunque no completo) de ciertos aspectos en cuanto al uso del repositorio; por ejemplo, si en el facet IP hay muchos accesos de una categoría que no corresponde a la IP de un motor de búsqueda tipo Google o un bot conocido, entonces es un indicio de una IP sospechosa a marcar como un bot malicioso[30] (Catá, Lira, & De Giusti, 2016), aunque sólo este análisis no es suficiente para confirmar esta sospecha.

Por último, como se observa en la Figura 5.8, se agregó un botón con forma de cruz en rojo que habilita la exclusión de una categoría de facet específica de los resultados. Es una forma ágil de escribir un filtro de negación «No es» por el valor de facet seleccionado. Por ejemplo: si se desea excluir el valor de país «AR», entonces se podría seleccionar el botón de exclusión sobre esta categoría de facet y, automáticamente, se generará el filtro «País - No es - Argentina».

Otras utilidades desde la interfaz de usuario

Como utilidades complementarias, se agregaron botones de vinculación desde distintas partes del repositorio al módulo de *Statistics-Discovery*. En la Figura 5.10 y Figura

[30] Un *bot* malicioso es un proceso informático que busca acceder y descargar todo el contenido posible del repositorio, generando un número muy alto de accesos y contaminando los registros estadísticos, además de buscar vulnerabilidades en el sistema para explotarlas.

5.11 se observan dos enlaces de vinculación entre las comunidades, colecciones e items con sus respectivas registros de uso en el módulo. Además, desde el módulo *Discovery* se agregó el botón «Ir a estadísticas» que permite determinar los registros de uso a partir de los objetos resultantes de una búsqueda, como se observa en la Figura 5.12 y se define en la sección «Scopes» de este capítulo, con una opción para determinar si el contexto es jerárquico o no. Esta última utilidad permite vincular el core «search» con el core «statistics» de forma directa; por ejemplo, para ver el uso realizado sobre las Tesis del repositorio, primero se tendrían que filtrar los ítems del tipo «Tesis» desde las opciones de facetado en el módulo Discovery y luego seleccionar el botón para ir a las estadísticas de esos resultados.

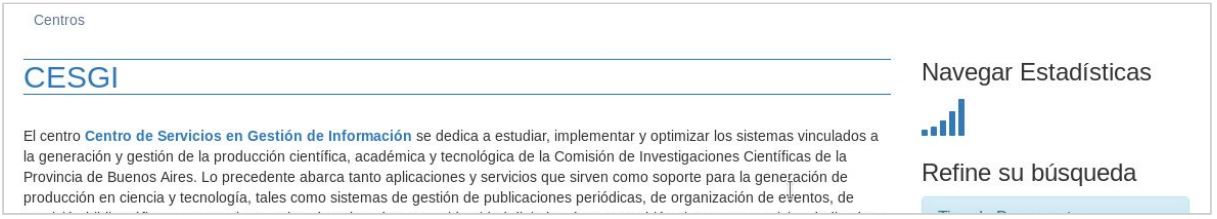


Figura 5.10 - Agregados a la vista de comunidad/colección

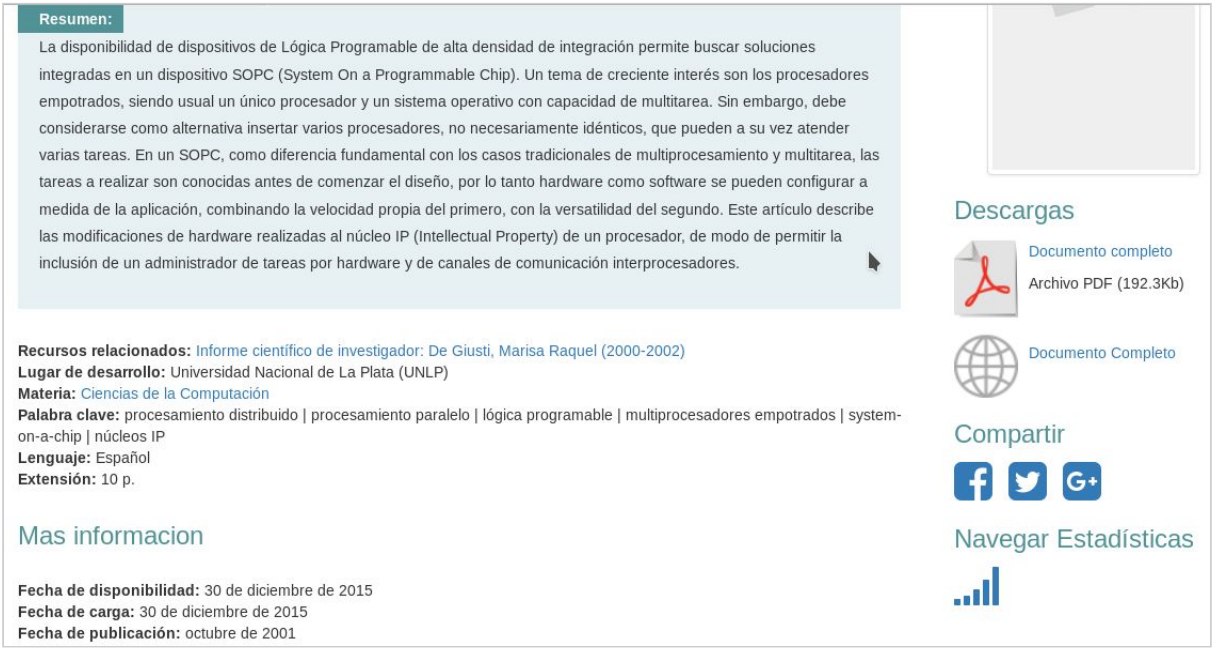


Figura 5.11 - Agregados a la vista de un ítem

Figura 5.12 - Vista modificada del módulo *Discovery*

También se agregaron algunas herramientas alineadas a cada resultado de búsqueda del módulo *Statistics-Discovery*, visualizadas en la Figura 5.14: una de estas es un inspector que permite visualizar en forma detallada todos los campos indexados para cada registro retornado, y la otra herramienta permite visualizar en un mapa la posición geoespacial del usuario que accedió al repositorio a partir de las coordenadas geoespaciales indexadas en cada registro, conformadas por los campos *longitude* y *latitude;* el servicio de mapas web utilizado para la visualización se llama *OpenStreetMap* (https://www.openstreetmap.org) (ver Figura 5.13).

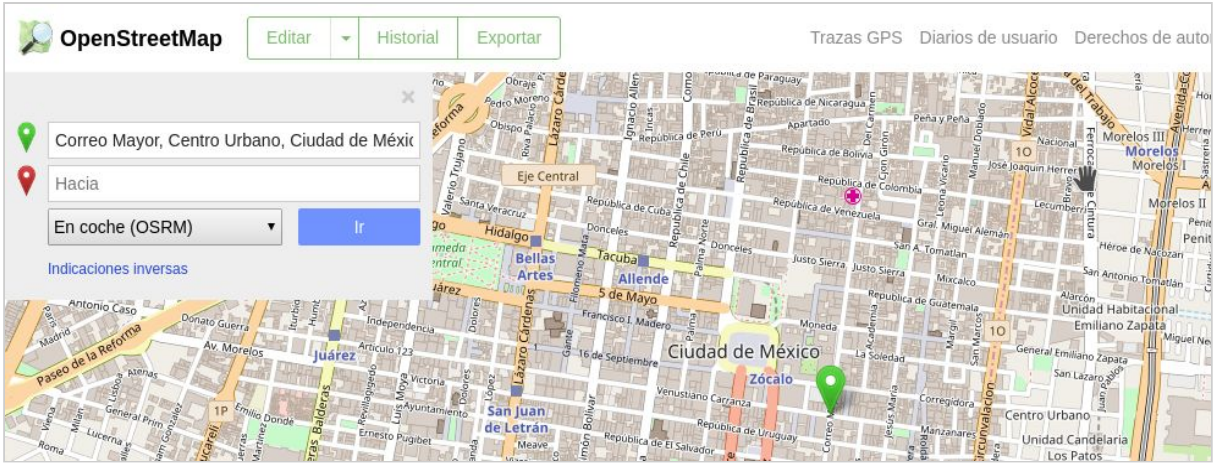


Figura 5.13 - Visualización de coordenadas en *OpenStreetMap*

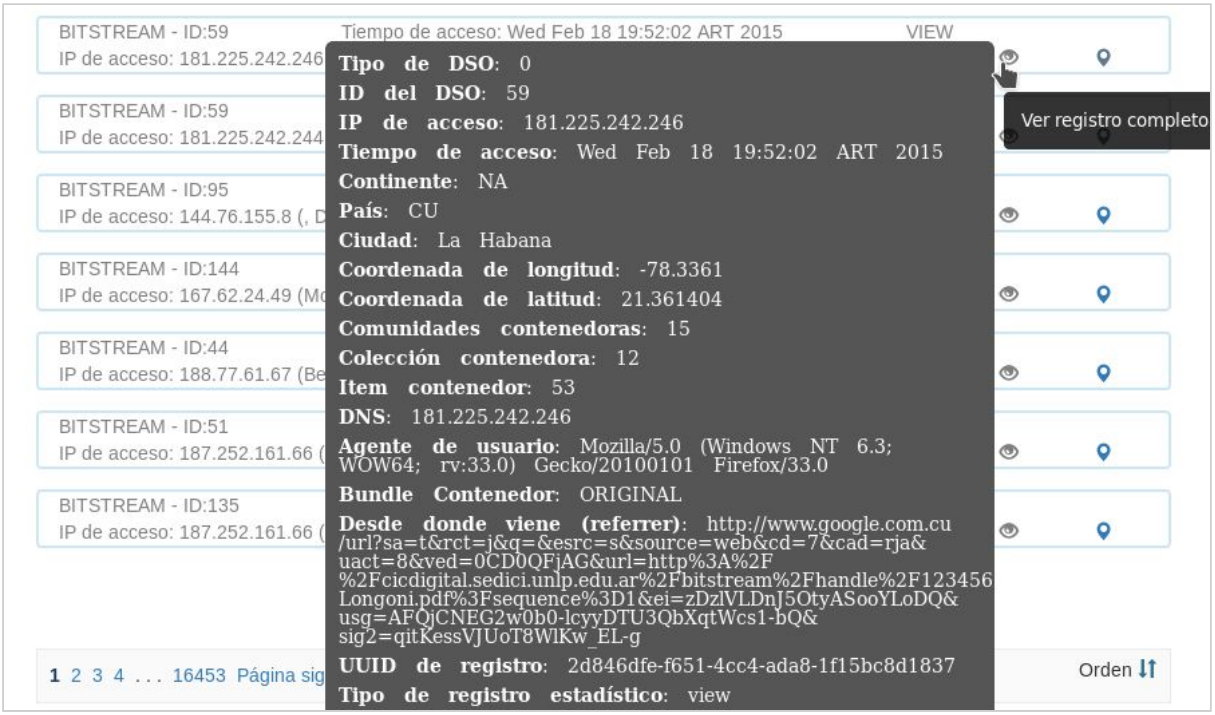


Figura 5.14 - Vista detallada de registros

Código del prototipo

El código fuente del prototipo basado en la versión 6 del software DSpace de CIC-Digital se encuentra en el repositorio git https://github.com/FacundoAdorno/DSpace/tree/tesina_discovery_xmlui. También se agrega como material complementario a este trabajo un cd/dvd anexo donde se incluye este mismo código fuente.

Capítulo 6 | Conclusiones y trabajos futuros

Conclusiones

Durante el transcurso de este trabajo se detallaron los motivos detrás de la necesidad de crear una nueva herramienta que mejore la explotación de los datos de uso indexados en el core «statistics» de DSpace. Para ésto, se realizó una investigación del marco teórico sobre el que encuadra este trabajo, junto con una especificación de la arquitectura del software DSpace, la plataforma sobre la que se desarrolló la herramienta; asimismo, se analizaron las ventajas y desventajas sobre su implementación en las versiones 6 y 7 de DSpace, y las tecnologías vinculadas a cada una de éstas. Finalmente, luego de inspeccionar las carencias del módulo de estadísticas existente en el software, se detallaron algunos casos de uso que la herramienta debería satisfacer y se explicó el desarrollo de la misma junto con el detalle de las funcionalidades implementadas.

Concretamente, la herramienta se desarrolló sobre la versión 6 del *software* DSpace mediante la creación de un módulo específico basado en el módulo *Discovery* de la plataforma, y la utilización de las tecnologías que funcionan sobre esta versión: Solr, Apache Cocoon, Spring Framework, XSLT, Javascript, entre otras. Si bien la alternativa de la versión 7 del software era la mejor a largo plazo por sus ventajas a nivel tecnológico, ésta todavía se encuentra en una etapa reciente de desarrollo y sin una fecha definitiva de lanzamiento. Por último, la herramienta fue probada en el repositorio institucional CIC-Digital, que se encuentra en la versión 6 de la línea de desarrollo de DSpace.

Las ventajas de disponer de esta herramienta en un repositorio son varias. Entre ellas se podría mencionar la fácil exploración de los datos de uso indexados en el repositorio (que pueden a llegar a crecer a una cantidad de millones de datos), y todo ésto sin requerir de conocimientos técnicos sobre Solr, la plataforma de indexación utilizada para almacenar estos datos. Además, la posibilidad de exportación en diversos formatos de los datos explorados desde la herramienta, permite la realización de análisis más avanzados y complejos en herramientas externas de mayor potencia y dedicadas al análisis estadístico de datos (como por ejemplo, Matlab o R). Otra de las ventajas visibles es la posibilidad de vinculación entre los datos administrados por los módulos *Discovery* y *Statistics-Discovery*, de tal forma de poder determinar los registros de uso de los ítems resultantes de una búsqueda en Discovery; esto posibilita la rápida obtención de los datos de uso de los objetos en DSpace a partir de sus características (por ejemplo, para los ítems del tipo «Tesis de grado», los ítems del autor «Juan Pérez», o los ítems cuya año de publicación fue el 2018).

Problemas encontrados

Durante el desarrollo de la herramienta se encontraron diversos problemas, principalmente situaciones sobre el uso de la plataforma Solr y los datos indexados en ésta, algunos de los cuales tuvieron que resolverse para poder avanzar con la implementación.

Por una parte, algunos de los problemas se debieron a la definición del esquema de campos para el core «statistics» en Solr. Entre estos problemas se puede mencionar la existencia de campos que tienen diferentes nombres pero que comparten una misma semántica, y que solo aplican sobre determinados tipos de registros. Un ejemplo de esto son los campos *type* y *dsoType*, que determinan el tipo de objeto DSpace vinculado a un registro estadístico, donde el campo *type* solamente existe para los registros del tipo «view» mientras que el *dsoType* solamente existe para los de tipo «search»; la necesidad de utilizar estos dos campos en un mismo facet llevó a la implementación del nuevo tipo de filtro combinado. Otro problema encontrado en relación al esquema es que a partir de la versión 6 de DSpace se comenzó a utilizar UUIDs (cadenas de texto de caracteres hexadecimales que representa un identificador único universal, p.e. a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11) como identificador de los objetos asociados a cada registro, sin embargo el esquema de 'statistics' trataba los campos utilizados para guardar este valor (*id, owningItem, owningColl, owningComm*) como del tipo numérico en vez de cadena de texto; debido a esto, para estos campos tuvo que cambiarse la configuración de esquema del índice Solr, ya que generaban errores al cargar algunos de los registros desde Solr hacia DSpace.

Por otra lado, surgieron algunos problemas causados por la longitud que algunas consultas a Solr llegaban a tener. Uno de los aspectos de longitud estaba asociado a la cantidad máxima de caracteres que una URL soporta, y el otro se vinculaba a la cantidad máxima de condiciones booleanas que una consulta a Solr podía manejar. En el primer caso, tuvo que actualizarse el método HTTP de consulta a Solr, es decir, en vez de utilizar el método GET se cambió para la utilización del método POST, superando así el problema del tamaño máximo de la URL soportada por GET ya que por POST esta cuestión es irrelevante. En el segundo caso, para que Solr determine los registros a retornar se examinan un conjunto de cláusulas booleanas determinadas a partir de los parámetros de la consulta; cuando las cláusulas a analizar superan una cantidad máxima configurada (en la propiedad *maxBooleanClause* en los cores de Solr) entonces Solr falla en la resolución de la consulta. Para resolver este problema se tuvo que actualizar la propiedad *maxBooleanClause* modificando el valor por defecto (1024 cláusulas) a un valor superior. Para ejemplificar, una situación en la que la consulta sea demasiada larga puede ocurrir al utilizar la funcionalidad de obtención de registros de uso a partir de un scope determinado por una consulta Discovery, que debido a su implementación a veces requiere de la definición de múltiples cláusulas booleanas.

También existen algunos problemas de performance en la funcionalidad de generación de reportes y gráficas cuando la cantidad de registros involucrados asciende a los millones, y en particular sucede para la generación de «reportes con un lapso de tiempo específico» (p.e. un reporte de (A)*cantidad de registros por ciudad registrados mensualmente* o (B)*cantidad de descargas de bitstreams por país registrados anualmente*). La solución óptima a esta situación hubiera sido utilizar la funcionalidad en Solr llamada *facet.pivot por rangos de fecha*[31]*;* por ejemplo, en la situación (B) se podría definir un facet sobre el campo

[31] Ver información sobre facet.pivot por rango de fecha en la documentación de Solr https://lucene.apache.org/solr/guide/6_6/faceting.html#Faceting-CombiningFacetQueriesAndFacetRangesWithPivotFacets

«*countryCode*» utilizando como pivote el campo «*statistics_type*» y vinculandolo con un *rango de fecha anual* generado sobre el campo «*time*». Sin embargo, como esta funcionalidad está disponible a partir de 5.4 de Solr y la versión de Solr utilizada por DSpace es la 4.10.2, entonces los reportes de este tipo tuvieron que generarse de una manera indirecta, produciendo un deterioro en la performance en la generación de estos reportes en comparación a que si esto hubiese sido resuelto directamente desde Solr.

Trabajos Futuros

Si bien la herramienta cumplió con la mayoría de los objetivos inicialmente propuestos, podría ser mejorada en varios aspectos. A continuación se especifican algunos de estas mejoras, que quedarán como posibles trabajos futuros a realizar sobre la base de este trabajo.

Una cuestión pendiente en la implementación de la herramienta es el agregado de una capa de seguridad, para permitir acceder a la exploración de los registros y el resto de las funcionalidades sólo a los usuarios con los respectivos permisos. Por ejemplo, permitir explorar los registros sólo a los Administradores del repositorio, o permitir que cada usuario registrado puede acceder a los registros de uso de sus propias publicaciones, entre otras alternativas. Este mecanismo podría estar basado en el sistema de permisos mediante *resource policies* que ya existe en DSpace.

También sería de utilidad permitir la compartición de los distintos reportes generados por los usuarios. Por ejemplo, si un usuario quisiera compartir la gráfica del reporte de descargas de sus publicaciones correspondientes al año 2018 (ya sea por correo, por embebimiento de código HTML, por alguna red social, etc.), sería de utilidad agregar en la herramienta algún mecanismo que lo permita, como por ejemplo algún paquete de botones sociales y botones para el embebido HTML, y la posibilidad de exportación en distintos tipos de formatos de imagen.

Con el motivo de enriquecer aún más la sección de reportes, sería conveniente posibilitar la generación de reportes más específicos y diversos, mediante el agregado de más opciones en cuanto al tipo de gráficos a generar (que hasta ahora solo se realizan gráficas de barra y de líneas), la posibilidad de editados de leyendas de los datos y de los ejes de la gráfica, o el aumento de la cantidad de tipos de reportes que un usuario puede crear. Otra mejora importante dentro de esta sección sería la inclusión de datos estadísticos que se calculan a partir de estos reportes (como p.e. el cálculo de una tabla de distribución de frecuencias, la moda, media, mediana, etc.), así como la representación gráfica de esos datos estadísticos.

Otra trabajo a tener en cuenta sería agregar funcionalidades que desde la interfaz de usuario de la herramienta permitan mejorar la calidad de los datos indexados en el core ‘statistics’. A partir del análisis de los datos explorados desde la herramienta se pueden encontrar situaciones anómalas en los datos (por ejemplo, una IP que cuenta con millones de accesos y que no está declarada como bot, o una IP que se corresponde con múltiples países y ciudades), y la depuración de estas anomalías permitiría la generación de reportes de uso más confiables y precisos.

Un trabajo complementario a la herramienta, pero alineado a los objetivos de la misma, es el de habilitar la generación de reportes orientados a lo que son las estadísticas de crecimiento del repositorio (explicado en mayor detalle en el Capítulo 2). De esta manera, se podría ampliar el marco de estadísticas disponibles para la asistencia en la toma de decisiones, es decir, basándose no solo en los reportes generados a partir de los registros de uso de los recursos en el repositorio, sino también de reportes a partir de las características conjuntas de un conjunto de recursos y su crecimiento en el tiempo; por ejemplo: sería interesante poder consultar cuestiones como «el crecimiento de las publicaciones de Artículos del departamento de Física durante el período 2015-2018».

Finalmente, sería importante en un futuro poder migrar el código de la herramienta para las próximos lanzamientos de DSpace y en particular para DSpace 7, donde las tecnologías utilizadas para implementar la arquitectura del cliente y el servidor presentan cambios notables. Asimismo, se podría considerar también presentar esta herramienta como una contribución a la comunidad de desarrollo y así integrarlo oficialmente en Dspace.

Bibliografía

About OpenDOAR. (s. f.). Recuperado el 12 de junio de 2018, a partir de http://www.opendoar.org/about.html.

American National Standards Institute (2014). *ANSI/NISO Z39.93-2014 The Standardized Usage Statistics Harvesting Initiative (SUSHI) Protocol*. Recuperado a partir de https://www.niso.org/publications/z3993-2014-sushi.

Apache Solr. (2018, mayo 24). En *Wikipedia*. Recuperado a partir de https://en.wikipedia.org/w/index.php?title=Apache_Solr&oldid=842816568.

Architecture - DSpace 6.x Documentation - DuraSpace Wiki (s. f.). [Wiki]. Recuperado 12 de junio de 2018, a partir de https://wiki.duraspace.org/display/DSDOC6x/Architecture.

Bernal, I., & Pemau-Alonso, J. (2010). Estadísticas para repositorios: sistema métrico de datos en *Digital.CSIC*. *El Profesional de la Información*, *19*(5), 534-544. https://doi.org/10.3145/epi.2010.sep.15.

Brody, T., Gedye, R., MacIntyre, R., Needham, P., Pentz, E., Rumsey, S., & Shepherd, P. (2009, enero). PIRUS – Publisher and Institutional Repository Usage Statistics. Recuperado a partir de https://www.webarchive.org.uk/wayback/archive/20140615025802/http://www.jisc.ac.uk/media/documents/programmes/pals3/pirus_finalreport.pdf.

Catá, J. M., Lira, A. J., & De Giusti, M. R. (2016). Detección de bots en reportes estadísticos. Presentado en *VI Conferencia Internacional BIREDIAL-ISTEC* (San Luis Potosí, México, 17 al 19 de octubre de 2016). Recuperado a partir de http://hdl.handle.net/10915/57473.

Chapter 2 - Understanding Apache Solr (2015). En H. V. Karambelkar, *Scaling Big Data with Hadoop and Solr - Second Edition*. Packt Publishing Ltd.

CICBA (Comisión de Investigaciones Científicas de la Provincia de Buenos Aires). Creación de CIC-Digital - Resolución 1146/14 (2014). Recuperado a partir de http://digital.cic.gba.gob.ar/handle/11746/64.

Client APIs | Apache Solr (2017, junio 9) [Wiki]. Recuperado el 22 de junio de 2018, a partir de https://lucene.apache.org/solr/guide/6_6/client-apis.html.

Configuration Reference - DSpace 6.x Documentation - Duraspace Wiki (s. f.) [Wiki]. Recuperado el 12 de junio de 2018, a partir de https://wiki.duraspace.org/display/DSDOC6x/Configuration+Reference#ConfigurationReference-OpenSearchSupport.

DCMI: DCMI Abstract Model (2007, junio 4). Recuperado el 22 de junio de 2018, a partir de http://dublincore.org/documents/abstract-model/.

DCMI: DCMI Metadata Terms (2012, junio 14). Recuperado el 22 de junio de 2018, a partir de http://dublincore.org/documents/dcmi-terms/.

De Giusti, M. R. (2017). Curso de posgrado: Bibliotecas y repositorios digitales. Tecnología y aplicaciones. Presentado en Curso de posgrado de repositorios digitales (Facultad de Informática). Recuperado a partir de http://hdl.handle.net/10915/62871.

Discovery - DSpace 6.x Documentation - DuraSpace Wiki (s. f.) [Wiki]. Recuperado el 12 de junio de 2018, a partir de https://wiki.duraspace.org/display/DSDOC6x/Discovery.

Donohue, T., Knowles, C., Lowel, A., & Bollini, A. (2017, febrero). *Introducing DSpace 7 Webinar Slides*. Technology. Recuperado a partir de https://www.slideshare.net/DuraSpace/22817-introducing-dspace-7-webinar-slides.

DSpace 7 Working Group - DSpace - DuraSpace Wiki (s. f.) [Wiki]. Recuperado el 12 de junio de 2018, a partir de https://wiki.duraspace.org/display/DSPACE/DSpace+7+Working+Group.

DSpace Release 6.0 Status - DSpace - DuraSpace Wiki (s. f.) [Wiki]. Recuperado el 12 de junio de 2018, a partir de https://wiki.duraspace.org/display/DSPACE/DSpace+Release+6.0+Status.

DSpace Service based api - DSpace - DuraSpace Wiki (s. f.) [Wiki]. Recuperado el 12 de junio de 2018, a partir de https://wiki.duraspace.org/display/DSPACE/DSpace+Service+based+api#DSpaceServicebasedapi-Servicebasedapi.

Estelle, L., & Lambert, J. (2015). Data That Counts, Charleston Conference 2015. Purdue University Press. Recuperado a partir de https://doi.org/10.5703/1288284316296.

Features - Top Reasons To Use DSpace (s. f.). Recuperado el 12 de junio de 2018, a partir de https://duraspace.org/dspace/about/features/.

Fushimi, M. (2016). Desarrollo de repositorios digitales institucionales en las universidades nacionales en Argentina, período 2004-2015 (p. 27). Presentado en Segundo Congreso Argentino de Estudios Sociales de la Ciencia y la Tecnología (CAESCyT), San Carlos de Bariloche – Provincia de Río Negro – Argentina. Recuperado a partir de http://www.memoria.fahce.unlp.edu.ar/library?a=d&c=eventos&d=Jev7888

IRUS-UK (s. f.). Recuperado el 12 de junio de 2018, a partir de http://www.irus.mimas.ac.uk/.

Jacobs, N. (2016, mayo). What is a repository? | Jisc scholarly communications. Recuperado el 22 de junio de 2018, a partir de https://scholarlycommunications.jiscinvolve.org/wp/2016/05/31/what-is-a-repository/

Johnson, R. K. (2002). Institutional Repositories: Partnering with Faculty to Enhance Scholarly Communication. *D-Lib Magazine*, *Volumen 8* (11). https://doi.org/10.1045/november2002-johnson.

Journal Usage Statistics Portal (s. f.). Recuperado el 12 de junio de 2018, a partir de https://jusp.jisc.ac.uk/.

LA Referencia (s. f.). Recuperado el 12 de junio de 2018, a partir de http://www.lareferencia.info/joomla/es/.

MacIntyre, R., & Jones, H. (2016). IRUS-UK: Improving Understanding of the Value and Impact of Institutional Repositories. *The Serials Librarian*, *70*(1-4), 100-105. https://doi.org/10.1080/0361526X.2016.1148423.

Merlino, C. S. (2014, septiembre). *#Aprender3C - Métricas y estadísticas en Repositorios y Bibliotecas Digitales*. Educación. Recuperado a partir de https://es.slideshare.net/Aprender3C/aprender3c-mtricas-y-estadsticas-en-repositorios-y-bibliotecas-digitales.

OAI 2.0 Server - DSpace 6.x Documentation - DuraSpace Wiki (s. f.). Recuperado el 12 de junio de 2018, a partir de https://wiki.duraspace.org/display/DSDOC6x/OAI+2.0+Server#OAI2.0Server-OAI2.0.

Open access (2018, junio 10). En *Wikipedia*. Recuperado a partir de https://en.wikipedia.org/w/index.php?title=Open_access&oldid=845287685.

ORCID Integration - DSpace 6.x Documentation - DuraSpace Wiki (s. f.) [Wiki]. Recuperado el 12 de junio de 2018, a partir de https://wiki.duraspace.org/display/DSDOC6x/ORCID+Integration.

Programación por capas (2018, abril 11). En *Wikipedia, la enciclopedia libre*. Recuperado a partir de https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_por_capas&oldid=106948221.

Repositorios digitales - Red Infod. (s. f.). Recuperado 12 de junio de 2018, a partir de https://red.infd.edu.ar/articulos/repositorios-digitales/.

Seeley, Y. (s. f.). *Lucene/Solr Architecture*. Recuperado a partir de https://people.apache.org/~yonik/presentations/solr_architecture.ppt.

SOLR Statistics - DSpace 6.x Documentation - DuraSpace Wiki. (s. f.). [Wiki]. Recuperado el 12 de junio de 2018, a partir de https://wiki.duraspace.org/display/DSDOC6x/SOLR+Statistics.

Solrj - Solr Wiki. (s. f.). [Wiki]. Recuperado el 12 de junio de 2018, a partir de https://wiki.apache.org/solr/Solrj/.

Spring Framework - Core Technologies (s. f.) [Wiki]. Recuperado el 12 de junio de 2018, a partir de https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/core.html#beans.

The COUNTER Code of Practice for e-Resources: Release 4 (2012, abril). Recuperado a partir de https://www.projectcounter.org/wp-content/uploads/2016/01/COPR4.pdf.

User Interfaces - DSpace 6.x Documentation - DuraSpace Wiki. (s. f.). [Wiki]. Recuperado el 12 de junio de 2018, a partir de https://wiki.duraspace.org/display/DSDOC6x/User+Interfaces.

Web analytics (2018, mayo 14). En *Wikipedia*. Recuperado a partir de https://en.wikipedia.org/w/index.php?title=Web_analytics&oldid=841158888.

W3C (2009, diciembre). Namespaces in XML 1.0 (Third Edition). Recuperado a partir de https://www.w3.org/TR/xml-names/.