



UNIVERSIDAD
NACIONAL
DE LA PLATA

SEDICI

REPOSITORIO INSTITUCIONAL DE LA UNLP



Capacitación IUPA

Clase 1

PREBI
prebi.unlp.edu.ar

Pablo de Albuquerque, Santiago Tettamanti, Ariel Lira
{pablo, santit, alira}@sedici.unlp.edu.ar



Esta obra está bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).

20 de Abril de 2022

Agenda de clases

1. Introducción a DSpace y entorno de desarrollo local
2. Git, configuración básica de DSpace, personalización frontend
3. Submission y Workflow, identificadores, autenticación
4. Configuración del servidor y administración avanzada de DSpace.



Clase 1



Introducción

- Introducción a DSpace
- Arquitectura
- Modelo de contenidos
- Instalación
 - Mvn y ant
- Entorno de desarrollo local
 - Docker
- Actividades clase 1

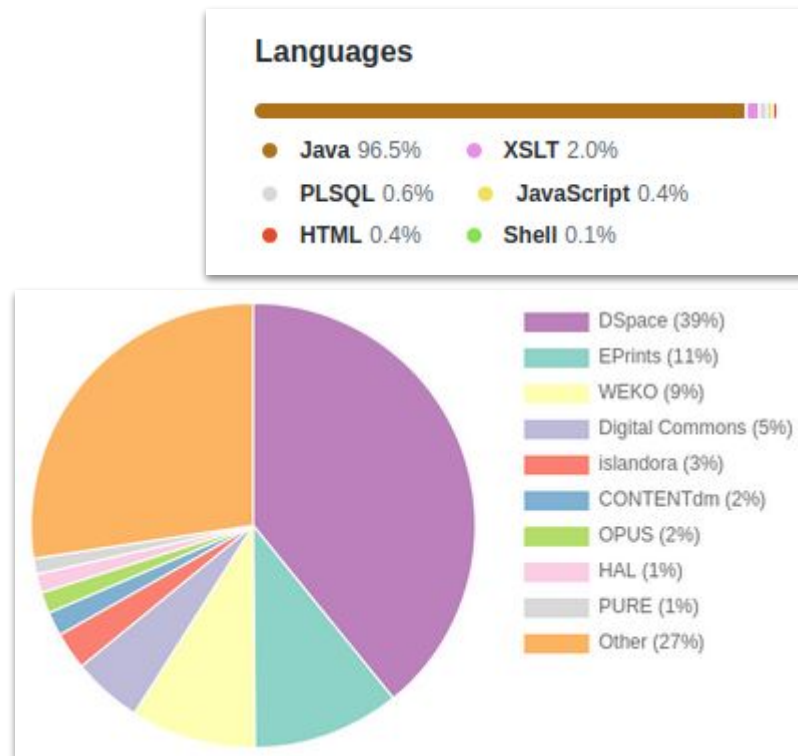


Introducción a DSpace

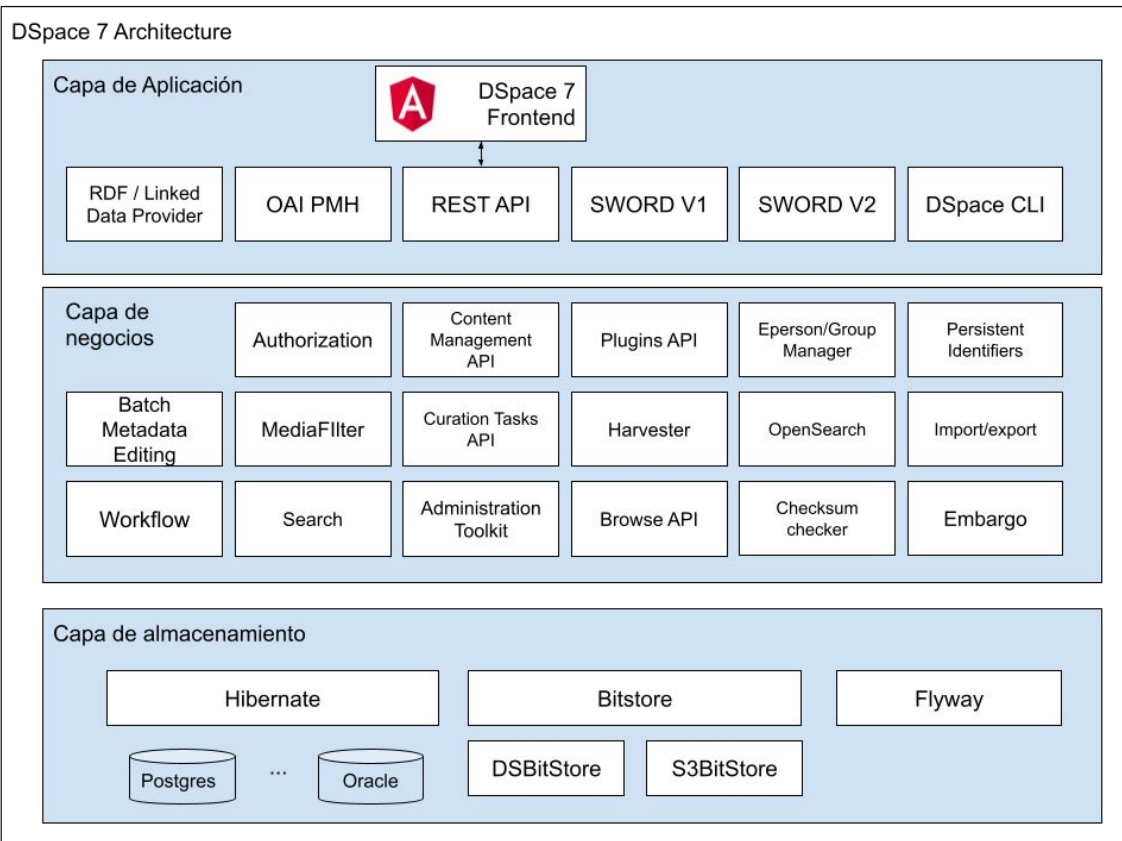


Introducción a DSpace - Características generales

- Software libre y open-source
- Proyecto Maven escrito mayoritariamente JAVA.
- Compuesto por varios subproyectos.
- Amplia **comunidad** de usuarios en todo el mundo
- <https://wiki.lyrasis.org/display/DSPACE/>
- **Extensible y configurable**



Introducción a DSpace - Arquitectura v7



Organizado en 3 capas

- [Aplicación](#)
- [Lógica de negocios](#)
- [Almacenamiento](#)

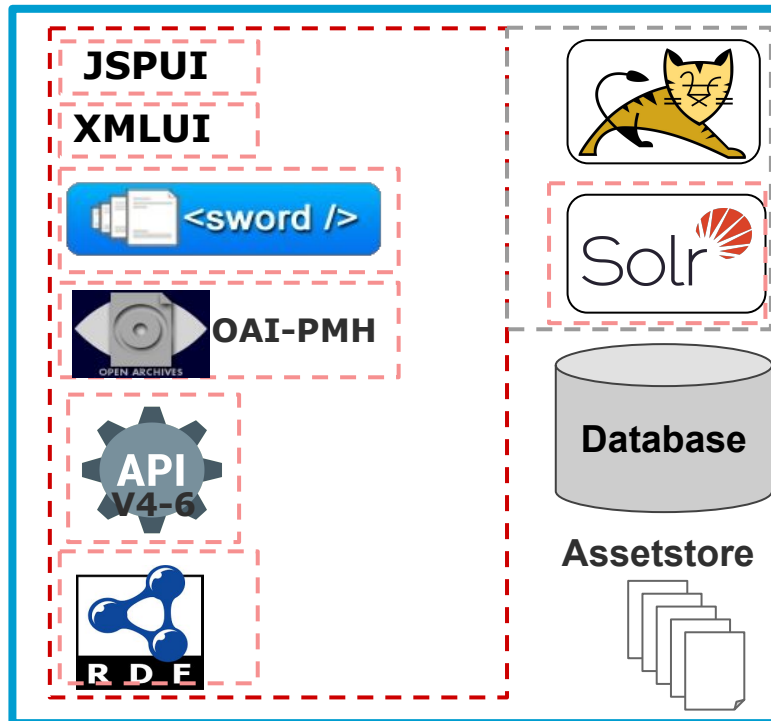


Introducción a DSpace - Arquitectura v1-6

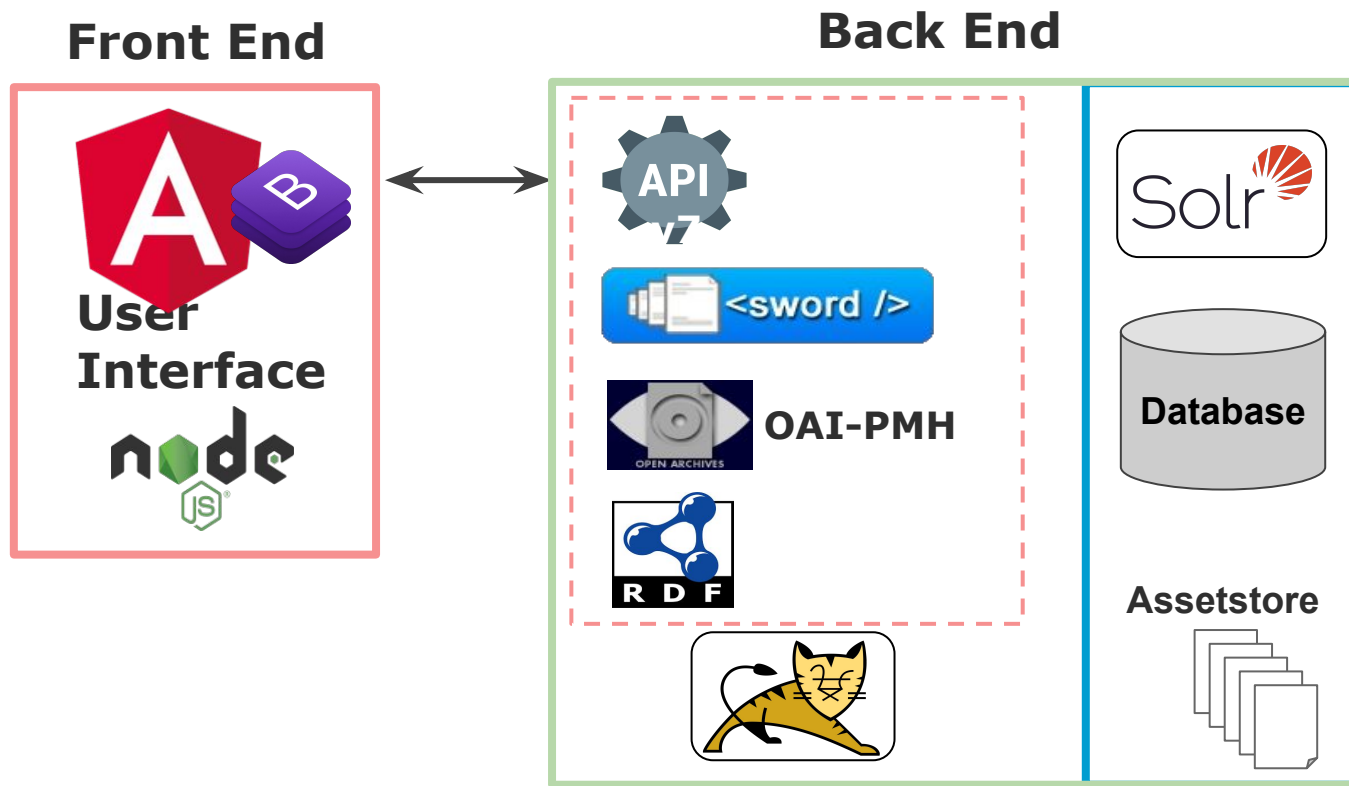
7 webapps separadas:
2 UIs, 4 servicios web &
Solr



Front & Back End



Introducción a DSpace - Arquitectura v7



Características generales de DSpace - Stack

Proyecto Maven escrito mayoritariamente JAVA. Compuesto por varios subproyectos. Se requiere:

- Java + Spring
 - jre11 o jre17
- Web container o web server
 - Jetty, Tomcat, entre otros
- Índice de búsqueda
 - Solr, Elasticsearch
- Base de datos relacional para persistir el modelo.
 - PostgreSQL, Oracle, etc
- Angular
 - para la interfaz de usuario, solo DSpace 7



Introducción a DSpace - Stack

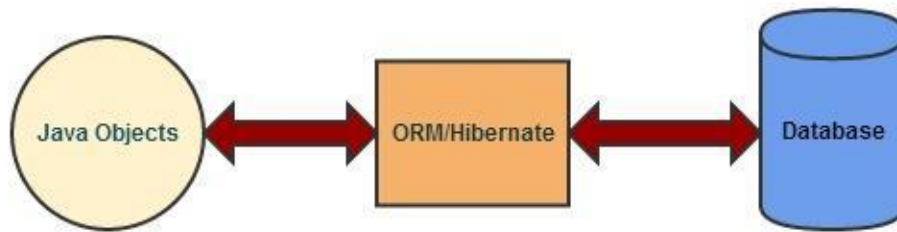
Spring es un framework que implementa el modelo de **Dependency Injection**

- Spring en DSpace administra:
 - Los **servicios** de la Capa de la Lógica de Negocios ([core-services.xml](#))
 - [MetadataValueService](#)
 - [ItemService](#)
 - [CollectionService](#), etc.
 - Las **extensiones** al core de DSpace (generalmente en [\[DSPACE-DIR\]/dspace/config/spring/api](#))
 - [discovery.xml](#)
 - [workflow-actions.xml](#)
 - [bitstore.xml](#), etc.



Introducción a DSpace - Stack

Hibernate ORM es un framework para persistencia a partir del mapeo Objeto a Relaciones.



- **Abstrae** a la aplicación de la BD utilizada
- **Mapea automáticamente** los datos de objetos al momento de leer de/persistir en la BD
- Permite usar consultas abstractas independientes de la BD usando lenguaje **HQL**
- Provee:
 - caching de objetos
 - transacciones de alto nivel
 - entre otros.



Introducción a DSpace - Stack

También se utiliza un indexador de texto para

- registros de acceso (estadísticas)
- índice de búsqueda (metadatos y fulltext)
- otros usos.

Solr:

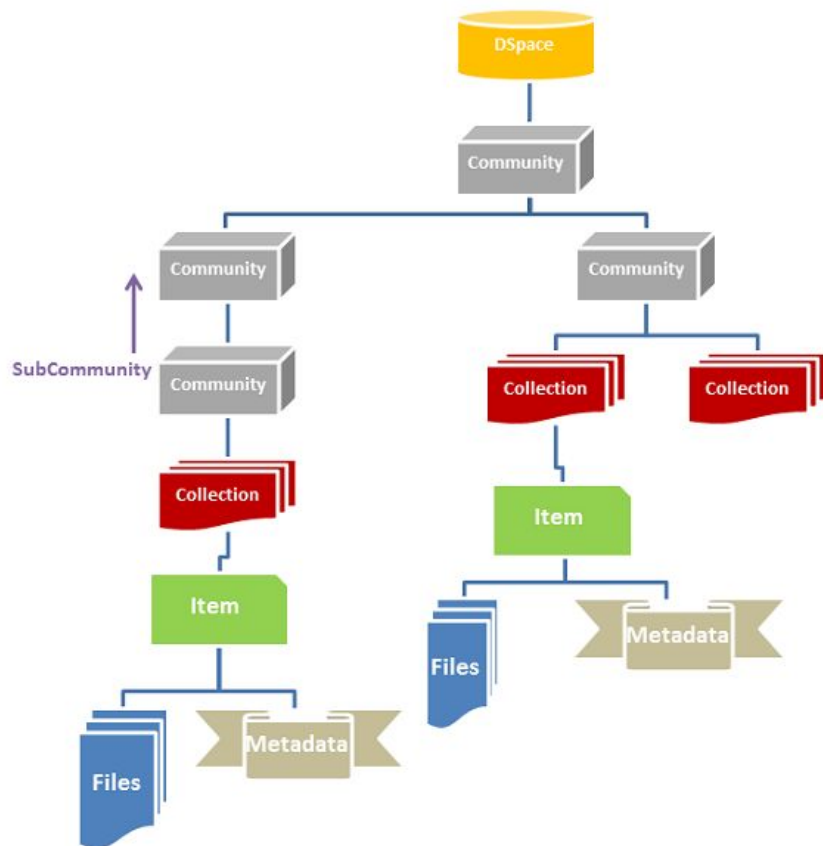
- está optimizado para búsquedas
- permite definir relevancia, filtrado, **pesos** en los campos y más.
- permite gestionar grandes volúmenes de datos



Modelo de contenidos



Modelo de contenidos - Estructura típica



1. Una o más **comunidades** de nivel base
2. Se organizan **jerárquicamente** en subcomunidades.
3. Las **colecciones** agrupan contenido relacionado dentro de las **comunidades**.
4. Los ítems son las obras que van en las **colecciones** y que se pretende que el público encuentre.
5. Los **metadatos** describen al recurso
6. Los **bitstreams** son la **representación digital** del recurso.



Modelo de contenidos - Comunidad y Colecciones

Comunidades en DSpace

Elija una comunidad para listar sus colecciones

▼ Centros [2053]

Centros de la Comisión de Investigaciones Científicas

▶ CIDCA [1]

Centro de Investigación y Desarrollo en Criotecología de Alimentos

▶ CESGI [1]

Centro de Servicios en Gestión de Información

▼ LEMIT [806]

Laboratorio de Entrenamiento Multidisciplinario para la Investigación Tecnológica

Tesis [10]

Artículos y presentaciones en Congresos [20]

Libros [2]

▼ Revista Ciencia y Tecnología de los Materiales [32]

Número 01 [7]

Número 02 [7]

Número 03 [6]

Número 04 [6]

Número 05 [6]

▶ Informes de Investigación [34]

▶ Jornadas de Técnicas de Reparación y Conservación del Patrimonio [419]

▶ Revista Ciencia y Tecnología del Hormigón [46]

▶ Anales LEMIT. Serie II [237]

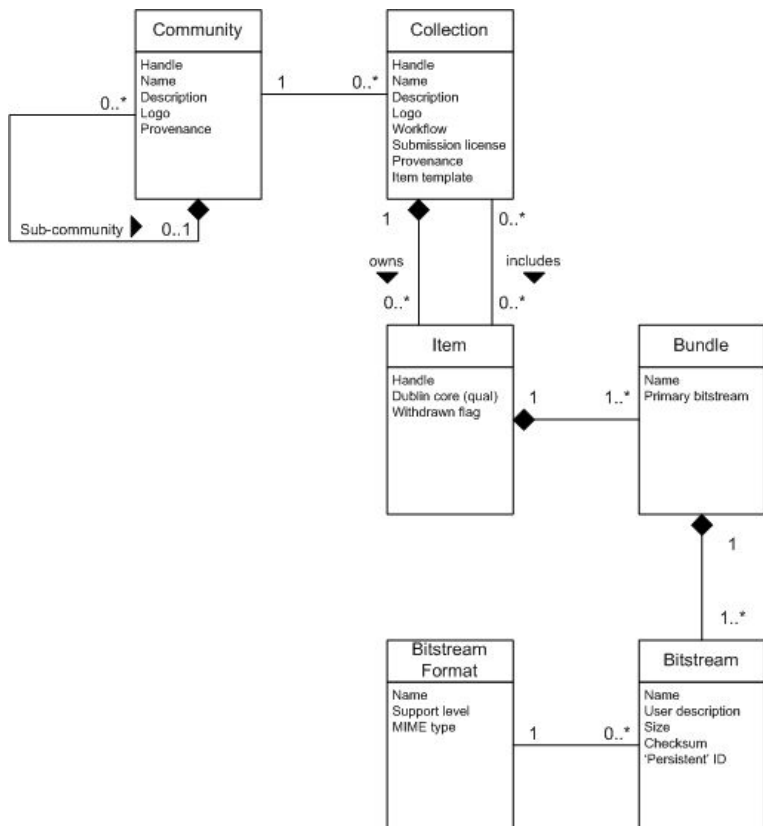
▶ Anales LEMIT. Serie III [6]

▶ CIOp [198]

Centro de Investigaciones Ópticas



Modelo de contenidos - DSpace Data Model



- Las comunidades poseen subcomunidades y colecciones
- Las colecciones poseen ítems
- Los ítems se componen de bundles
- Cada bundle posee bitstreams (archivos)



Modelo de contenidos - Bundles

Los ítems están vinculados a sus bitstreams a través de entidades llamadas **Bundles**. Estos agrupan **bitstreams** según ciertas categorías:

- **ORIGINAL**: contiene los bitstreams a publicar.
- **THUMBNAILS**: archivos con thumbnails extraídos a partir de los bitstreams originales (p.e. thumbnails de PDFs, imágenes, etc.).
- **TEXT**:
 - Texto completo (full-text) de otros bitstreams.
 - Se genera a partir de la extracción automática de texto sobre otros bitstreams
 - se usa durante la indexación para mejorar los resultados de búsqueda.
- **LICENSE**: licencia que el usuario aceptó al depositar el contenido.
- **CC_LICENSE**: contiene la [Licencia CC](#) (si es que existe) seleccionada por el usuario durante la carga.



Modelo de contenidos - Bitstreams

Los ítems del repositorio se componen de **objetos digitales** que representan la obra en sí que se quiere publicar. Estos pueden ser:

- Audios
- PDFs
- Documentos de texto (.doc, .odt., etc)
- Diapositivas de presentaciones
- Planillas de cálculo
- Imágenes
- Videos
- etc.

Estos objetos digitales reciben el nombre de **BITSTREAMS.**

The screenshot shows the CIC DIGITAL website interface. At the top, the logo 'CIC DIGITAL' is displayed next to the text 'Repositorio Institucional Comisión de Investigaciones Científicas'. Below the logo is a navigation menu with options: 'Inicio', 'Explorar', 'Aportar Material', 'Mas información', and 'Contacto'. On the right side of the header, there are links for 'Mi cuenta', 'Estadísticas', and 'ES'. The main content area shows a breadcrumb trail: 'Centros / LINTA / III Jornadas "Turismo y desarrollo". Turismo cultural: perspectivas y desafíos'. Below this, the document title is 'El Falcon Verde como lugar simbólico y la posibilidad de su activación para un uso turístico responsable', with the author 'Sahores Avalis, Virginia'. A 'Resumen:' section follows, containing text about the historical significance of the Falcon Verde in Argentina. To the right of the text, there is a box indicating the year 'Año 2013'. At the bottom right, there is a 'Descargas' section with a PDF icon and the text 'Documento completo Archivo PDF (5.525Mb)'. A red rectangular box highlights the PDF download area.

Modelo de contenidos - BitStore

Los bitstreams son alojados en un **Bitstore**, que puede estar alojado de forma local o en la nube (según la configuración en [dspace/config/spring/api/bitstore.xml](#)):

- **ASSETSTORE:** si tenemos esta forma de almacenamiento, los bitstreams se alojan en un directorio local en el servidor:

```
{dspace-install-dir}/assetstore/
```

- **S3Store:** si tenemos esta forma de almacenamiento, los bitstreams se alojarán en la nube (en el Cloud Storage de [Amazon S3](#)). Para que esto funcione debemos configurar diferentes parámetros como, p.e, una *accessKey* al sistema de Amazon.



Modelo de contenidos - Metadatos

dc.format.extent	116 p.	es
dc.language	es	es
dc.title	Un metalenguaje de programación orientado al diseño de interfaces gráficas	es
dc.type	Tesis	es
sedici.creator.person	Santana, Carlos Ariel	es
sedici.creator.person	Coni, Cintia Vanesa	es
sedici.subject.materias	Ciencias Informáticas	es
sedici.subject.keyword	arquitectura dirigida por modelos	es
sedici.subject.keyword	transformaciones de modelos	es
sedici.subject.keyword	interfaz de usuario	es
sedici.description.fulltext	true	es
mods.originInfo.place	Facultad de Informática	es
sedici.subtype	Tesis de grado	es
sedici.rights.license	Creative Commons Attribution 4.0 International (CC BY 4.0)	
sedici.rights.uri	http://creativecommons.org/licenses/by/4.0/	
sedici.contributor.director	Pons, Claudia	es
thesis.degree.name	Licenciado en Sistemas	es
thesis.degree.grantor	Facultad de Informática	es

Los metadatos

- son datos estructurados que describen otros datos
- son datos sobre datos

← Ej: representación del recurso

<http://sedici.unlp.edu.ar/handle/10915/47008>



Modelo de contenidos - Metadatos - Representación



Algunos esquemas:

- **Dublin Core Element Set (DCES)** es uno de los esquema de metadatos estándar más utilizados <http://dublincore.org/documents/dces/>
 - **Título** → dc.title
- **DCMI Terms (dc extendido o calificado)** es una extensión al esquema básico de DC <http://dublincore.org/documents/dcmi-terms/>
 - **Título alternativo** → dcterms.alternative
- Un repositorio puede utilizar un **perfil de metadatos** basado en esquema propio, estándar y/o mixto
 - Ej: subconjunto o extensión de DC.



Modelo de contenidos - Metadatos - Representación

Ejemplo: SEDICI utiliza un perfil de metadatos propio basado en:

→ **DC simple**

- ◆ dc.title

→ **DC calificado**

- ◆ dc.date.accessioned

→ **ETD**

- ◆ thesis.degree.name

→ **MODS**

- ◆ mods.location

→ **Esquema de metadatos SEDICI**

- ◆ sedici.subtype

dc.date.issued	2009	
dc.identifier.uri	http://hdl.handle.net/10915/1102	
dc.description.abstract	Tomo I: Memoria descriptiva; Informe medioambiental; Trazado; Proyecto de arquitectura la Estación Plaza Moreno. Tomo II: Proyecto estructural de la Estación Plaza Moreno. Tomo III: Plan de trabajos; Evaluación económica	es
dc.format.extent	3 vol. + planos	es
dc.language	es	es
dc.title	Línea "A" de subte - La Plata	es
dc.type	Tesis	es
sedici.creator.person	Liaudat, Joaquin	es
mods.location	http://biblio.ing.unlp.edu.ar/cgi-bin/koha/opac-detail.pl?bib=INGC-MON-017361	es
sedici.subject.materias	Ingeniería	es
sedici.subject.materias	Ingeniería Civil	es
sedici.subject.eurovoc	transporte subterráneo	es
sedici.subject.eurovoc	medio de transporte	es
sedici.description.fulltext	false	es
mods.originInfo.place	Departamento de Construcciones	es
sedici.subtype	Tesis de grado	es
sedici.contributor.director	Venier, Leonardo P.	es
thesis.degree.name	Ingeniero Civil	es
thesis.degree.grantor	Facultad de Ingeniería	es

Modelo de contenidos - Metadatos

DSpace permite definir **perfiles de metadatos** a partir de

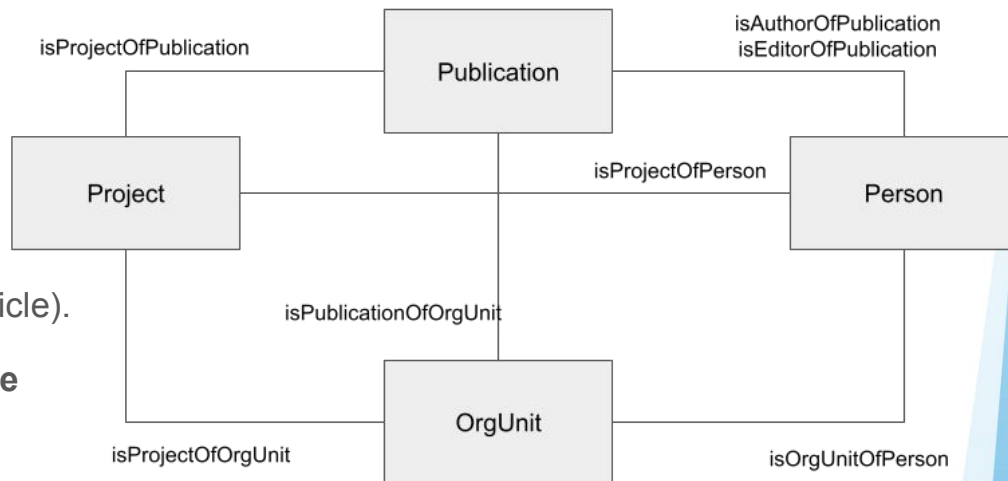
1. combinación de elementos de diferentes schemas
2. usa de calificadores:
 - schema.element.qualifier

DC Simple (estándar DCES)	DC simple (perfil en DSpace)
dc.date	dc.date.accessioned dc.date.issued dc.date.created dc.date.available dc.date.submitted



Modelo de contenidos - DSpace Configurable Entities

- Dspace 7 incorpora conceptos de DSpace-CRIS como entidades configurables.
- Las entidades provistas son:
 - Publication, Person, OrgUnit, Project
 - Journal Entities include Journal, Journal Volume, Journal Issue and Publication (article).
- Los ítems tienen un metadato **dspace.entity.type** que indica su entidad
- Cada ítem/entidad puede relacionarse con otros ítems/entidades a partir de relaciones pre-existentes o definidas ad-hoc.
- <https://wiki.lyrasis.org/display/DSDOC7x/Configurable+Entities>



Modelo de contenidos - DSpace Configurable Entities

- Cada relación tiene:
 - Nombre de la relación
 - Entidades involucradas
 - Cardinalidad
- Entidades y relaciones se configuran en un XML
 - relationship-types.xml
 - La información de este xml se vuelca en la base de datos a través de un script



Modelo de contenidos - DSpace Configurable Entities

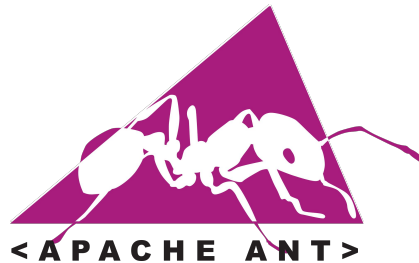
- relationship-types.xml

```
<relationships>
  <type>
    <leftType>JournalIssue</leftType>
    <rightType>Publication</rightType>
    <leftwardType>isPublicationOfJournalIssue</leftwardType>
    <rightwardType>isJournalIssueOfPublication</rightwardType>
    <leftCardinality>
      <min>0</min>
    </leftCardinality>
    <rightCardinality>
      <min>0</min>
      <max>1</max>
    </rightCardinality>
  </type>
</relationships>
```



DEMO con SEDICI y CIC-DIGITAL





Proceso de compilación e instalación



Maven

Es un software para gestión de proyectos Java que permite controlar **actividades recurrentes** como:

- Construcción del árbol de dependencias
- Descarga automática de dependencias
- Filtrado de archivos de configuración
- Compilación
- Test
- Empaquetado (jar, war, pom)
- Infinidad de acciones más como deploy, generación de documentación, etc.

The logo for Apache Maven, featuring the word "Maven" in a bold, black, sans-serif font. The letter "v" is replaced by a stylized feather with a gradient of colors from purple to orange. A small "TM" trademark symbol is positioned to the upper right of the "n".

Maven™



Maven - Repositorio local

- Por default se ubica en `~/.m2`.
- Por cada dependencia descargada, mantiene copia del paquete descargado:

```
<dependency>  
  <groupId>org.hibernate</groupId>  
  <artifactId>hibernate-core</artifactId>  
  <version>4.2.21.Final</version>  
</dependency>
```

```
.m2/  
|--org/  
|----hibernate/  
|-----hibernate-core/  
|-----4.2.21.Final/  
|-----hibernate-core-4.2.21.Final.jar  
|-----hibernate-core-4.2.21.Final.pom
```

- Cuando Maven construye y compila un proyecto, antes de ir a buscar alguna dependencia a un repositorio externo, verifica que no se encuentre en el repositorio local.
- El repositorio local `.m2` puede llegar a crecer varios GBs.



Maven - Comandos frecuentes

`mvn package`

Realiza una compilación de todo el proyecto y crea los paquetes necesarios (jar, war, etc), dejándolos en un subdirectorio llamado `target/`.

`mvn install`

Realiza un ‘`mvn package`’ y copia los paquetes generados en el repositorio local `.m2`

`mvn clean`

Elimina el directorio de trabajo `target/` creado durante la compilación de un proyecto.

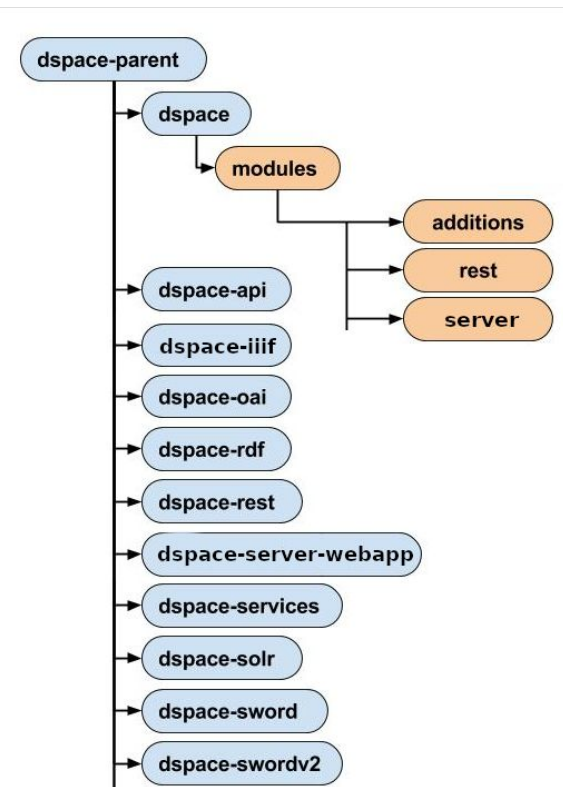
Una lista más completa de comandos ver en [documentación](#).



Maven - Estructura de proyectos

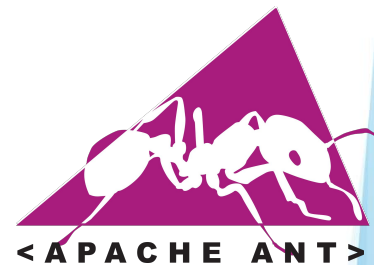
1. se divide en un proyecto padre y submódulos
2. el módulo ***dspace***
 - a. filtra los archivos de configuración
 - b. reúne las personalizaciones con el código original de dspace.
 - c. genera el paquete de instalación

IMPORTANTE: si estamos trabajando sobre una versión de desarrollo (no es release, ej 7.0) o si hicimos cambios en el código de dspace (directorios azules) se debe ejecutar el mvn package en `dspace-parent`. Caso contrario se puede ejecutar en `dspace-parent/dspace`



Ant

Apache Ant es una herramienta que permite automatizar tareas.



- Su funcionamiento se rige por el archivo `build.xml`
- provee tareas predefinidas que permiten manipular archivos, compilar, testear y correr aplicaciones Java.
- las acciones principales en DSpace son:
 - **ant fresh_install**
 - Realiza la instalación en el directorio `dspace.dir` (base de datos, índices Solr, assetstore, etc).
 - **ant update**
 - Actualiza una instalación previamente realizada
 - **ant clean_backups**
 - Elimina los directorio `.bak` generados tras un `ant update`

En `dspace/src/main/config/build.xml` están todas las acciones posibles



Proceso de compilación - Maven

1. **mvn clean package** - en el directorio root de DSpace (`/[DSPACE-DIR]/`)
 - a. El compilado se genera en `[DSPACE-DIR]/dspace/target/dspace-installer`
2. **service tomcat stop** - parar la instancia de tomcat.



Proceso de compilación - Ant

3. Posicionarse en el directorio donde se encuentra el compilado
 - a. `/[DSPACE-DIR]/dspace/target/dspace-installer`
4. **ant clean_backups**
 - a. Para limpiar los backups realizados por ant en instalaciones previas
5. **ant update (ant fresh_install para la primera vez)**
 - a. Para instalar o actualizar el código de DSpace
 - b. El código queda en el directorio definido por la variable de config [dspace.dir](#)



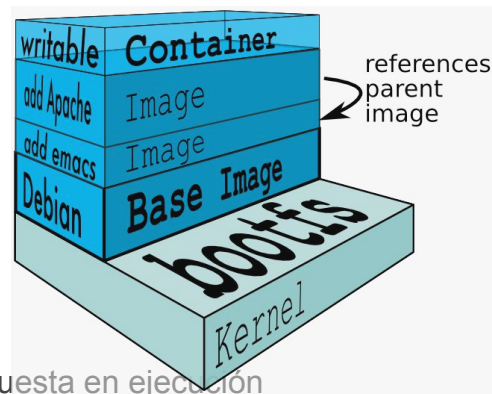
Entorno de desarrollo con Docker



Entorno desarrollo con Docker

Conceptos básicos:

- **Imágenes:**
 - plantilla que contiene instrucciones para crear un contenedor
 - definido en capas a partir de Dockerfiles
 - presentes localmente, en registros públicos y privados
- **Contenedores:**
 - es la instanciación de un entorno de ejecución a partir de una imagen y puesta en ejecución
- **Volúmenes:**
 - permiten montar el almacenamiento del host en el container



Docker - Acciones frecuentes

Creación de una imagen a partir del Dockerfile

- `docker build <dockerfile>`

Crear container a partir de una imagen

- `docker run <imagen> -p <puertos> -v <volúmenes> -e <variables de entorno>`

Reiniciar container con:

- `docker start <nombre_del_container>` (corre en background)

Detener un container:

- `docker stop <nombre_del_container>`

Obtener info de un Docker Object (container, image, network, etc):

- `docker inspect <docker-object>`



Docker - Acciones frecuentes - Docker compose

Docker compose nos facilita la tarea :-D

- Múltiples containers → Una sola aplicación
- A través de archivos yml
 - Por lo general llamados docker-compose.yml
 - Se definen los containers/servicios que tendrá nuestra aplicación
- Comandos para administrar los servicios creados
 - Iniciar, detener y reconstruir servicios
`docker-compose [up|down|build|restart]`
 - Listar los servicios definidos
`docker-compose images`
 - Visualizar la salida y el estado de los servicios en ejecución
`docker-compose [logs|events]`
 - Ejecute una acción en un servicio determinado
`docker-compose [kill|run] [servicio] [comando]`

```
version: "3.7"
services:
  dspacedb:
    image: postgres:9.5-alpine
    environment:
      POSTGRES_DB: "dspace"
      POSTGRES_USER: dspace
      POSTGRES_PASSWORD: dspace
    volumes:
      - /var/lib/postgresql/data
  dspace:
    image: dspace:dspace_latest
```

<https://docs.docker.com/compose/>



Docker - DSpace 7

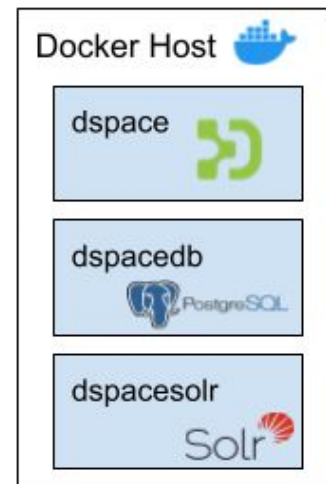
docker@dspace

Pros:

- Fácil deploy del frontend, backend + servicios complementarios (solr, postgres)
- Para desarrollo, prueba de configuraciones, pruebas de comando

Cons:

- No apto para producción
- No tan cómodo para desarrollo
- Ante cada cambio en el código se debe hacer un rebuild de la imagen → lento



Docker - Deploy del backend - Parte 1

1. Descargar fuentes de DSpace:

- `git clone https://github.com/DSpace/DSpace`
- `cd Dspace`

2. Crear una rama basada en dspace 7 para realizar cambios:

- `git branch prueba dspace-7.2.1 # (creo rama "prueba" basada en el tag "dspace-7.0")`
- `git checkout prueba #(me posiciono en la rama "prueba")`

3. Instalación y primer ejecución (paciencia, tarda bastante)

- `docker-compose -f docker-compose.yml -f docker-compose-cli.yml build`
- `docker-compose -f docker-compose.yml up -d`
- Acceder `http://localhost:8080/server` (HAL browser)

Nota: Si aparece un mensaje de error como el siguiente: `ERROR: Pool overlaps with other one on this address space`

Comentar en `docker-composer.yml`

```
networks:
  dspacenet:
#   ipam:
#config:
#- subnet: 172.23.0.0/16
```



Docker - Deploy del backend - Parte 2

4. Creación administrador:

- `docker-compose -f docker-compose-cli.yml run --rm dspace-cli create-administrator -e test@test.edu -f admin -l user -p admin -c en`

5. Ver listado de comandos disponibles:

- `docker-compose -f docker-compose-cli.yml run --rm dspace-cli help`



Docker - Deploy del frontend

- Es una alternativa para cuando se debe desarrollar sobre el **frontend únicamente**
- Se puede **conectar** con un **backend preexistente**, local o externo. Ej api7.dspace.org/server/

- **Instalación y primer ejecución:**

1. `git clone https://github.com/DSpace/dspace-angular`
2. `cd dspace-angular`
3. `git branch prueba dspace-7.2`
4. `git checkout prueba`
5. Definir url al rest en [config.dev.yml](#) (ver [config.example.yml](#))
6. `docker-compose -f docker/docker-compose.yml up -d`
7. Acceder <http://localhost:4000/>

Opcional. Por defecto es <http://localhost:8080/server>, que es la definida en el docker del backend. Podría usarse la demo <https://api7.dspace.org/server/>



Actividad 1

Probar instancia de dspace local

- a. Instalar git, docker y docker-compose
- b. Levantar DSpace backend y Angular de acuerdo a las 2 diapositivas anteriores
- c. Acceder al frontend <http://localhost:4000/>, loguearse con el usuario administrador creado
- d. Probar la creación de comunidades, colecciones, ítems, y cualquier otra cosa que desee.



Referencias

- <https://wiki.lyrasis.org/display/DSDOC7x/Using+DSpace>
- Links a doc de docker por proyecto
 - [Docker DSpace/dspace-angular](#)
 - [Docker DSpace/DSpace](#)



Lo que viene

- Uso de GIT
- Buenas prácticas para desarrollo sobre dspace
- Customización básica de una instalación DSpace
- Configuraciones

