

# Manual Técnico y Guía de Instalación: DSpace 7+

Prof. César Rodríguez - UDO/SUCRE

7 de febrero de 2026

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Arquitectura Técnica (DSpace 7+)</b>	<b>2</b>
2.1. Backend (DSpace-Server) . . . . .	2
2.2. Frontend (DSpace-UI) . . . . .	2
<b>3. Estructura de Datos</b>	<b>2</b>
<b>4. Guía de Instalación con Docker</b>	<b>2</b>
4.1. Prerrequisitos . . . . .	2
4.2. Instalación y Despliegue . . . . .	3
4.2.1. 1. Descarga del Código Fuente . . . . .	3
4.2.2. 2. Despliegue de Contenedores . . . . .	3
4.2.3. 3. Inicialización de la Base de Datos . . . . .	3
4.2.4. 4. Creación del Administrador . . . . .	3
4.3. Verificación . . . . .	3
4.4. Comandos Útiles . . . . .	3
4.4.1. Ver logs . . . . .	3
4.4.2. Detener el sistema . . . . .	4
4.5. Puesta en Producción (IP Pública) . . . . .	4
4.5.1. Ajustes Básicos . . . . .	4
4.5.2. Proxy Reverso (Recomendado) . . . . .	4
4.6. Resumen de la Instalación . . . . .	4
<b>5. Ejemplos de Código y Configuración</b>	<b>4</b>
5.1. Configuración Básica (local.cfg) . . . . .	4
5.2. Desarrollo Backend (Java API) . . . . .	5
<b>6. Implementación de Metadatos UDO (Análisis de PDFs)</b>	<b>5</b>
6.1. 1. Mapeo de Campos (Mapping) . . . . .	5
6.2. 2. Configuración del Formulario (input-forms.xml) . . . . .	6
<b>7. Áreas de Asistencia</b>	<b>6</b>

## 1. Introducción

DSpace es el software de código abierto más utilizado a nivel mundial para la creación de repositorios institucionales y archivos digitales. Las versiones modernas (DSpace 7 y 8) representan un cambio arquitectónico significativo respecto a las versiones anteriores (4, 5, 6).

## 2. Arquitectura Técnica (DSpace 7+)

A partir de la versión 7, DSpace se dividió en una arquitectura **Headless** (Backend y Frontend separados):

### 2.1. Backend (DSpace-Server)

- Basado en **Java** y **Spring Boot**.
- Expone una **REST API** completa (HAL browser compliant).
- Utiliza **Hibernate** para la persistencia de datos.
- Utiliza **Apache Solr** para la indexación y búsqueda.
- Base de datos: Generalmente **PostgreSQL** (o Oracle).

### 2.2. Frontend (DSpace-UI)

- Basado en **Angular** (TypeScript).
- Utiliza **Node.js** para el Server-Side Rendering (SSR) para mejorar el SEO.
- Se comunica exclusivamente con el backend a través de la REST API.

## 3. Estructura de Datos

DSpace sigue una jerarquía estricta basada en el modelo Dublin Core:

1. **Comunidades** (Communities)
2. **Colecciones** (Collections)
3. **Items** (Artículos/Tesis)
4. **Bitstreams** (Archivos físicos como PDFs)

## 4. Guía de Instalación con Docker

Esta sección describe el procedimiento para desplegar DSpace 7 utilizando contenedores Docker. Este método simplifica drásticamente la gestión de dependencias (Java, Maven, PostgreSQL, Solr) ya que todo se ejecuta dentro de contenedores aislados y preconfigurados.

### 4.1. Prerrequisitos

- **Docker Engine** (v19.03 o superior).
- **Docker Compose** (v1.25 o superior).
- **Git**.
- Al menos 4GB de RAM dedicados a Docker (recomendado 8GB).

## 4.2. Instalación y Despliegue

### 4.2.1. 1. Descarga del Código Fuente

Clonamos el repositorio principal de DSpace. Este repositorio incluye los archivos `docker-compose.yml` necesarios para levantar todo el stack (Backend, Frontend, BD y Solr).

```
1 mkdir /dspace-docker
2 cd /dspace-docker
3 git clone https://github.com/DSpace/DSpace.git -b dspace-7.6
4 cd DSpace
```

### 4.2.2. 2. Despliegue de Contenedores

Utilizamos Docker Compose para descargar las imágenes oficiales y levantar los servicios.

```
1 # Descargar las imágenes más recientes
2 docker compose pull
3
4 # Levantar los servicios en segundo plano (detached mode)
5 docker compose -p dspace up -d
```

Esto iniciará los siguientes servicios automáticamente:

- **dspace-angular**: Frontend (Puerto 4000).
- **dspace-server**: Backend API (Puerto 8080).
- **dspace-db**: Base de datos PostgreSQL.
- **dspace-solr**: Motor de búsqueda Solr.

### 4.2.3. 3. Inicialización de la Base de Datos

Una vez que los contenedores estén corriendo, es necesario ejecutar las migraciones iniciales para crear el esquema de la base de datos.

```
1 # Ejecutar migración de base de datos dentro del contenedor
2 docker compose -p dspace exec dspace-server dspace database migrate
```

### 4.2.4. 4. Creación del Administrador

Para acceder al sistema y gestionar contenidos, creamos una cuenta de administrador (e-person).

```
1 docker compose -p dspace exec dspace-server dspace create-administrator
```

Siga las instrucciones en pantalla para ingresar email, nombre, apellido y contraseña.

## 4.3. Verificación

Acceda a las siguientes URLs desde su navegador:

- **Frontend (UI)**: <http://localhost:4000>
- **Backend (API)**: <http://localhost:8080/server>
- **HAL Browser**: <http://localhost:8080/server/api>

## 4.4. Comandos Útiles

### 4.4.1. Ver logs

```
1 # Ver logs del backend en tiempo real
2 docker compose -p dspace logs -f dspace-server
3
4 # Ver logs del frontend
5 docker compose -p dspace logs -f dspace-angular
```

#### 4.4.2. Detener el sistema

```
1 docker compose -p dspace down
```

### 4.5. Puesta en Producción (IP Pública)

Cuando disponga de la IP pública asignada por Teleinformática, deberá actualizar la configuración para permitir el acceso externo.

#### 4.5.1. Ajustes Básicos

- **Backend:** Actualizar `dspace.server.url` y `dspace.ui.url` en `local.cfg`.
- **Frontend:** Configurar el cliente Angular para que apunte a la IP pública de la API (usualmente mediante variables de entorno `DSPACE_REST_HOST` en Docker).

#### 4.5.2. Proxy Reverso (Recomendado)

Para un entorno de producción seguro, no exponga los puertos 8080/4000 directamente. Configure **Nginx** o **Apache** como proxy reverso en el servidor host para:

1. Recibir tráfico en puertos estándar (80/443).
2. Gestionar certificados SSL (HTTPS).
3. Redirigir internamente a los contenedores Docker.

### 4.6. Resumen de la Instalación

Al completar estos pasos, usted tendrá efectivamente **DSpace 7.6** ejecutándose en su PC GNU/Linux bajo la siguiente arquitectura:

- **Contenerización Total:** Backend, Frontend, Base de Datos y Solr corren aislados en Docker.
- **Persistencia:** Los datos se guardan en volúmenes de Docker, no se pierden al apagar el container.
- **Versión:** Se garantiza la versión 7.6 al haber seleccionado la rama específica en el paso de clonación.

## 5. Ejemplos de Código y Configuración

### 5.1. Configuración Básica (`local.cfg`)

En lugar de editar el `dspace.cfg` principal, las mejores prácticas de ingeniería dictan que debes sobreescibir las configuraciones en `local.cfg`.

```
1 # dspace/config/local.cfg
2
3 # Configuración del directorio de instalación
4 dspace.dir = /dspace
5
6 # Configuración de la URL pública (Frontend Angular)
7 dspace.ui.url = https://repositorio.miuniversidad.edu.co
8
9 # Configuración de la REST API (Backend Java)
10 dspace.server.url = https://api.repositorio.miuniversidad.edu.co/server
11
12 # Conexión a Base de Datos (PostgreSQL)
13 db.url = jdbc:postgresql://localhost:5432/dspace
14 db.username = dspace
```

```

15 db.password = mi_password_seguro
16 db.schema = public
17
18 # Configuración de Solr (Búsqueda)
19 solr.server = http://localhost:8983/solr

```

Listing 1: Configuración en local.cfg

## 5.2. Desarrollo Backend (Java API)

Si necesitas crear una tarea curada (Curation Task) o un script personalizado para manipular ítems programáticamente, interactuarás con los servicios del Kernel de DSpace.

```

1 package org.dspace.example;
2
3 import java.sql.SQLException;
4 import java.util.UUID;
5 import org.dspace.content.Item;
6 import org.dspace.content.service.ItemService;
7 import org.dspace.core.Context;
8 import org.dspace.services.factory.DSpaceServicesFactory;
9
10 public class ItemProcessor {
11
12     // Obtenemos el servicio de Items a través de la factoría de servicios
13     protected ItemService itemService = DSpaceServicesFactory.getInstance().
14         getItemService();
15
16     public void processItem(Context context, String uuidString) {
17         try {
18             // Convertir String a UUID
19             UUID uuid = UUID.fromString(uuidString);
20
21             // Buscar el item usando el contexto actual
22             Item item = itemService.find(context, uuid);
23
24             if (item != null) {
25                 // Ejemplo: Leer metadatos (Dublin Core Title)
26                 String title = itemService.getMetadataFirstValue(
27                     item, "dc", "title", null, Item.ANY
28                 );
29
30                 System.out.println("Procesando Item: " + title);
31
32                 // Aquí podrías añadir lógica para modificar metadatos o bitstreams
33             }
34         } catch (SQLException e) {
35             e.printStackTrace();
36         }
37     }
}

```

Listing 2: Procesamiento de Items en Java

## 6. Implementación de Metadatos UDO (Análisis de PDFs)

Basado en los documentos “Hoja de Metadatos” e “Instructivo 2024” de la Universidad de Oriente, se ha diseñado la siguiente estrategia de configuración para DSpace 7.6.

### 6.1. 1. Mapeo de Campos (Mapping)

Se propone la siguiente correspondencia entre el formato UDO y el esquema Dublin Core (dc) o esquemas específicos de tesis (thesis).

Campo UDO	Campo DSpace	Notas
Título / Subtítulo	dc.title / .alternative	Estándar
Autor(es)	dc.contributor.author	Incluir ORCID/Email
Palabras Claves	dc.subject	
Área / Línea Inv.	dc.subject.classification	O crear esquema local ‘udo’
Resumen	dc.description.abstract	
Tutor	dc.contributor.advisor	Rol ‘TU’ en PDF
Jurado	dc.contributor.committeeMember	Rol ‘JU’ (Requiere registro)
Fecha Aprobación	dc.date.issued	
Alcance Espacial	dc.coverage.spatial	
Alcance Temporal	dc.coverage.temporal	
Grado Asociado	thesis.degree.name	Requiere esquema ‘thesis’
Nivel Asociado	thesis.degree.level	Ej. Licenciatura, Maestría
Institución	dc.publisher	Universidad de Oriente

## 6.2. 2. Configuración del Formulario (input-forms.xml)

Ejemplo de configuración para los campos de “Alcance” y “Jurado” definidos en la página 4 del formato UDO.

```

1 <row>
2   <!-- Campo: Alcance Espacial -->
3   <field>
4     <dc-schema>dc</dc-schema>
5     <dc-element>coverage</dc-element>
6     <dc-qualifier>spatial</dc-qualifier>
7     <repeatable>true</repeatable>
8     <label>Alcance Espacial</label>
9     <input-type>onebox</input-type>
10    <hint>Ej. Universal, Inespacial, o lugar geográfico</hint>
11    <required>Requerido</required>
12  </field>
13  <!-- Campo: Jurado (Requiere haber registrado 'committeeMember' previamente) -->
14  <field>
15    <dc-schema>dc</dc-schema>
16    <dc-element>contributor</dc-element>
17    <dc-qualifier>committeeMember</dc-qualifier>
18    <repeatable>true</repeatable>
19    <label>Jurado</label>
20    <input-type>name</input-type>
21    <hint>Apellidos y Nombres del Jurado</hint>
22    <required></required>
23  </field>
24 </row>
```

Listing 3: Bloque XML para Alcance y Jurado

**Nota sobre Despliegue:** Al modificar archivos XML de configuración (`submission-forms.xml`, `input-forms.xml`), es necesario reiniciar el contenedor del backend para recargar los cambios:

```
1 docker compose restart dspace-server
```

## 7. Áreas de Asistencia

Puedo asistirte en áreas como:

- **Instalación:** Despliegue con Docker o instalación manual (Maven/Ant).
- **Personalización:** Modificar el tema de Angular o crear nuevos componentes.
- **REST API:** Cómo consumir la API para integraciones externas.

- **Migración:** Pasar de DSpace 6 (XMLUI/JSPUI) a DSpace 7/8.
- **OAI-PMH:** Configuración para la recolección de metadatos.