DOCUMENTACIÓN PARA EL PROCESO DE INSTALACIÓN DE DSPACE Y FUSEKI EN LINUX (UBUNTU 18.04)

# ¿QUÉ ES LINUX, QUE ES UBUNTU?

Los sistemas operativos (SO) más utilizados como de computador personal como Windows o MacOS, pero no son los únicos, existen otros sistemas operativos basados en LINUX (Es un Kernel, lo que significa que se encarga de intercomunicar los componentes de hardware de nuestro dispositivo o PC con nuestros programas), los Sistemas Operativos basados en LINUX se les conoce como **DISTRIBUCIONES** de las cuales existen miles, cada una desarrollada para fines específicos (servidores, investigación, etc.), y otros para propósitos generales (uso personal). Desde hace tiempo los servidores basados en LINUX han ganado popularidad sobre los de Windows (Windows tiene una versión especial para servidores) cuyo uso se ha reducido notablemente por diversas razones. Para esta ocasión se eligió la distribución UBUNTU en su versión estable 18.04 por su facilidad y experiencia en su uso.

# ¿QUÉ ES UNA TERMINAL Y COMO LA UTILIZAMOS?

Los sistemas operativos para servidores pueden venir con interfaz gráfica, pero esto reduce la velocidad y otras capacidades del servidor para realizar las tareas más importantes, por lo tanto, la distribución de UBUNTU utilizada no dispone de interfaz gráfica y es administrada por medio de una terminal la cual es una consola de comandos por la cual le damos instrucciones y ordenes al servidor

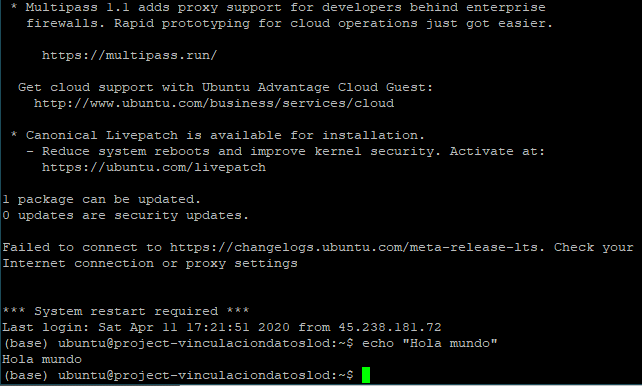


Fig. 1 Terminal UBUNTU del servidor

Por ejemplo, con el comando echo (fig.1) le ordena a nuestro sistema a que debe mostrar en pantalla el mensaje que estamos escribiendo a continuación en este caso el “Hola mundo”, en la línea siguiente nuestro sistema muestra el mensaje.

# ¿DÓNDE Y COMO FUNCIONA EL SERVIDOR, CÓMO ACCEDEMOS A LA TERMINAL DE LA MAQUINA?

El servidor se encuentra en el CECAD (centro de computación de alto desempeño) de la universidad Francisco José de Caldas, para su acceso se realizó una petición en la que definimos las características (8 Gb RAM, 4 Núcleos, 250 Gb almacenamiento y 3 puertos abiertos –ssh, tomcat, pruebas-). En cuanto responden se entregan las claves de acceso, así como la IP y los puertos:

* IP: 200.69.103.29
* ssh: 23380 🡪 Este permite acceder a la terminal del servidor.
* Tomcat: 26237
* Pruebas: 26236

Luego utilizando la herramienta puTTy se accede a la terminal de la máquina de la siguiente manera:

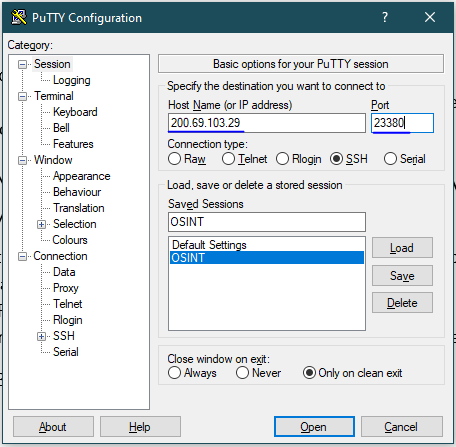


Fig. 2 Configuración de puTTy 1

Ingresamos la IP de la maquina y el puerto para conexión en modo SSH, recordar dejar el tipo de conexión en SSH

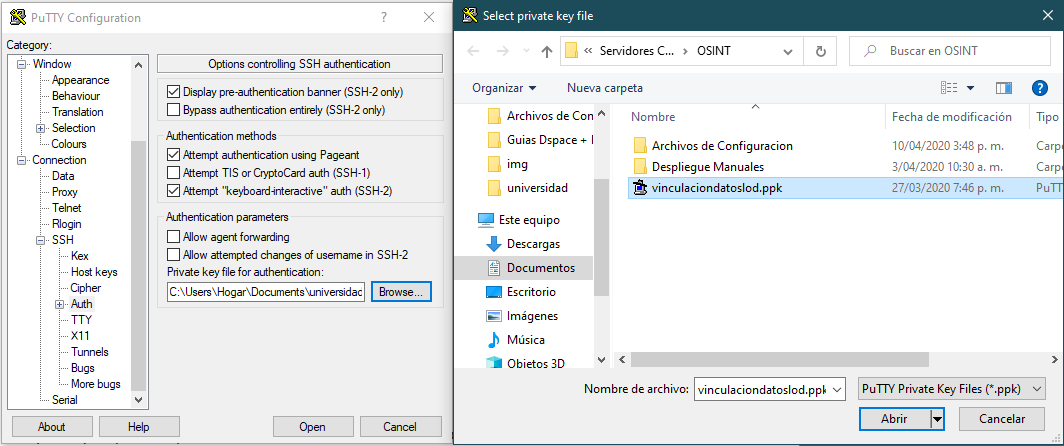


Fig. 3 Configuración de puTTy 2

Dirigirse a la sección auth de puTTy, hacer clic en *Browse…*, seleccionar el archivo de claves entregado, hacer clic en el botón *Open* y esperar a la terminal.

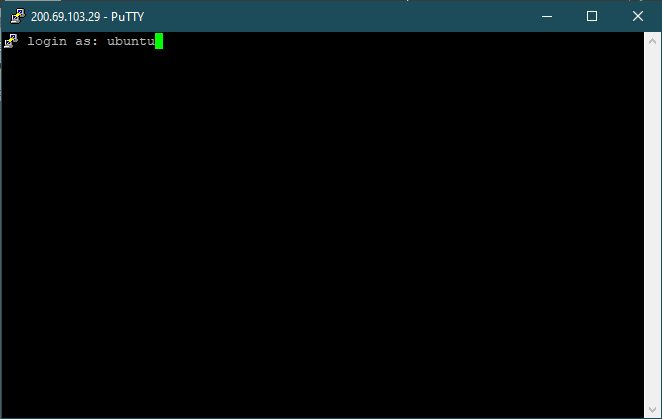


Fig. 4 Configuración de puTTy 3

Se abrirá la ventana como en la Figura 4, solo se añade el usuario al que se conectará y presiona ENTER, en este momento ya está conectado al servidor por medio de su terminal.

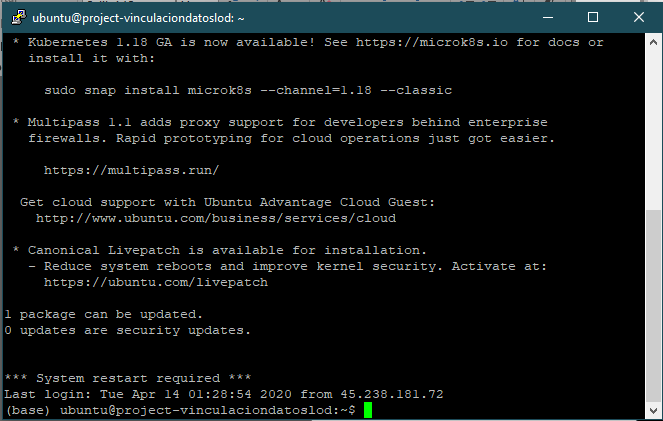


Fig. 5 Terminal del servidor

# ¿QUÉ SON LAS DEPENDENCIAS Y CUALES TIENE DSPACE Y FUSEKI?

Las aplicaciones que se utilizan a diario ya sea en Ubuntu, Windows o cualquier otro sistema operativo por lo general requieren de otras aplicaciones para funcionar de forma correcta o para complementar las funciones que tienen, por ejemplo, Google Chrome necesita de DirectX otro programa para que pueda funcionar (por lo general no se instala DirectX porque Windows ya lo trae instalado en sus últimas versiones). DSPACE y FUSEKI también requieren de otras aplicaciones para funcionar entre ellas están:

* Maven: Un software para la gestión y construcción de proyectos Java
* Tomcat: Es el que permite iniciar DSPACE/FUSEKI y accederlas con las direcciones web.
* Java: Lenguaje de programación y plataforma de ejecución de aplicaciones de este mismo lenguaje
* Apache Ant: Biblioteca Java y herramienta de línea de comandos cuya misión es impulsar los procesos descritos en los archivos de compilación como objetivos y puntos de extensión que dependen uno del otro. Con esta generamos algunos archivos de Dspace.
* PostgreSQL: Es la base datos en la cual Dspace guardara sus datos (Su configuración se mostrará más adelante)

Para instalar cada una de estas aplicaciones se realiza lo siguiente:

* En la terminal del servidor introduzca *sudo apt update* presione Enter e introduzca *sudo apt install openjdk-8-jdk openjdk-8-jre*, esto instalará automáticamente Java en nuestro sistema.

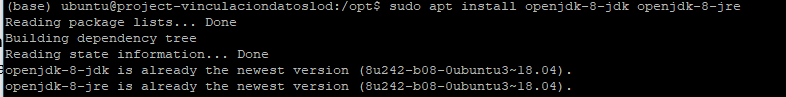


Fig. 6 Instalación de Java, en este caso la maquina ya lo tiene instalado

* Por su parte Maven, Tomcat, dspace y fuseki vienen en archivos comprimidos lo cual reduce su peso, por lo tanto, hay que descomprimirlos. Con el comando *wget* descargamos es archivo desde un sitio web y con el comando *tar* los descomprimimos, con *cd* se cambia de carpeta para instalar todo en una ruta más cómoda, ingrese cada línea presionando Enter luego de cada una:
* cd /opt
* wget <https://downloads.apache.org/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz>
* wget <https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.19/bin/apache-tomcat-9.0.19.tar.gz>
* wget <https://github.com/DSpace/DSpace/releases/download/dspace-6.3/dspace-6.3-src-release.tar.gz>
* wget <https://archive.apache.org/dist/ant/binaries/apache-ant-1.10.5-bin.tar.gz>
* wget <http://archive.apache.org/dist/jena/binaries/apache-jena-fuseki-3.10.0.tar.gz>
* tar -xvzf apache-jena-fuseki-3.10.0.tar.gz
* tar -xvzf apache-maven-3.6.3-bin.tar.gz
* tar -xvzf apache-tomcat-9.0.19.tar.gz
* tar -xvzf dspace-6.3-src-r
* tar -xvzf apache-ant-1.10.5-bin.tar.gz
* Luego de ejecutar cada comando introduzca el comando *ls* y debería ver el contenido de la figura 7



Fig. 7 Archivos descargados y descomprimidos (azules directorios)

* Una vez descomprimidos los datos están en carpetas (las carpetas tienen el mismo nombre del archivo sin incluir tar.gz) y ya no necesitamos los datos comprimidos, por este motivo utilizamos *rm* para borrar los comprimidos y *mv* (en general se usa para mover archivos, pero puede ser usado para renombrar) para renombrar algunas las carpetas que se utilizan con mayor frecuencia con nombres más cortos.
  + rm apache-maven-3.6.3-bin.tar.gz
  + rm apache-tomcat-9.0.19.tar.gz
  + rm dspace-6.3-src-release.tar.gz
  + rm apache-jena-fuseki-3.10.0.tar.gz
  + rm apache-ant-1.10.5-bin.tar.gz
  + mv apache-maven-3.6.3 maven
  + mv apache-tomcat-9.0.19 tomcat
  + mv apache-ant-1.10.5 ant

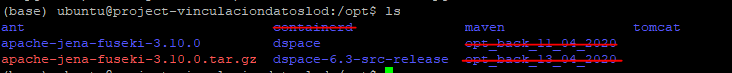


Fig. 8 Resultado luego de borrar paquetes y cambiar de nombre (ignorar tachados)

Hasta este punto están descargados y descomprimidos los archivos necesarios para que Fuseki y Dspace funcionen, pero requiere que se configuren las variables de entorno lo cual se verá en el siguiente punto.

# ¿QUÉ SON LAS VARIABLES DE ENTORNO Y CUALES UTILIZAMOS?

En este punto todos los archivos que necesitan Dspace y Fuseki esta descargados o instalados (Java), pero aún no se pueden utilizar, la razón es que, aunque las carpetas de cada dependencia de estas aplicaciones estén en el sistema, el sistema no sabe dónde están o como se llaman a pesar de que estén en el disco duro por lo tanto hay que buscar la forma de decirle al sistema donde las puede encontrar, para esto se utilizan las variables de entorno que son variables que dentro de la terminal almacenan los valores que queramos.



Fig. 9 Ejemplo Variable de entorno

En la fig. 9 se muestra como a una variable a la que nombramos VARIABLE (puede llevar cualquier nombre) y le asignamos el valor Hola, con echo podemos imprimir el valor de la variable para lo cual debemos añadir *$* antes del nombre de la variable.

Para que DSPACE y FUSEKI funcionen de forma correcta debemos crear las siguientes variables de entorno:

* JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64
* JRE\_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
* PATH=$JAVA\_HOME/bin:$PATH
* PATH=$JRE\_HOME/bin:$PATH
* MAVEN\_HOME=/opt/maven
* ANT\_HOME=/opt/ant
* CATALINA\_HOME=/opt/tomcat
* PATH="${MAVEN\_HOME}/bin:${PATH}"
* PATH="${ANT\_HOME}/bin:${PATH}"
* PATH="${CATALINA\_HOME}/bin:${PATH}"

Donde el texto separado por “/” indica donde está la carpeta de nuestros archivos. Para insertar estas variables de entorno ingresamos al *.bashrc* introduciendo el comando *nano ~/.bashrc*

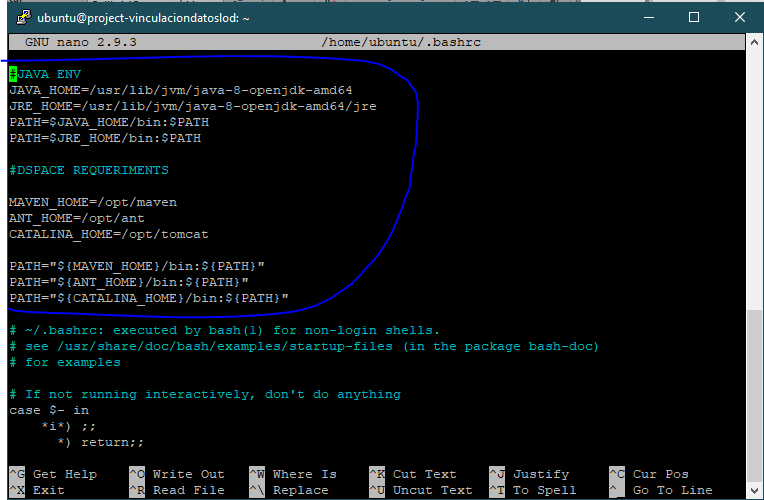


Fig. 10 Añadiendo variables de entrono al bashrc

Ingresamos las variables mencionadas, presionar *Ctrl+s* para guardar y *Ctrl+x* para salir. Por introducimos el comando *source ~/.bashrc* el cual actualizara la terminal actual. **Este proceso no requiere repetirse cada vez que se vuelva a iniciar la terminal, solo deberá hacerlo una vez**.

# ¿CÓMO CONFIGURAMOS NUESTRA BASE DE DATOS POSTGRES?

DSPACE requiere de un base de datos POSTRGESQL, para evitar el largo proceso de instalación en UBUNTU simplemente utilizamos una herramienta conocida como DOCKER, el cual nos permite crear un contenedor con la base de datos ya instalada y que requiere muy pocos pasos para configurarla, para ello realizamos los siguientes pasos:

* En la terminal se introducen las siguientes instrucciones seguido de ENTER en cada una
* sudo apt-get update

apt-get install \

apt-transport-https \

ca-certificates \

curl \

gnupg-agent \

software-properties-common

* sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –
* sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

* sudo apt-get update
* sudo apt-get install docker-ce docker-ce-cli containerd.io
* sudo curl -L "https://github.com/docker/compose/releases/download/1.25.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
* sudo chmod +x /usr/local/bin/docker-compose
* sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
* Se verifica que Docker y Docker-compose estén instalados con los comandos *docker --version* y *docker-compose --version*

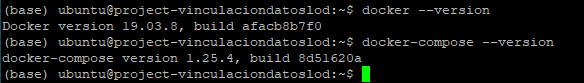


Fig. 11 Docker y Docker-compose instalados exitosamente

* De esta forma Docker ya está instalado solo nos queda crear dos archivos con los que Docker iniciara nuestra base de datos
  + Se crea un directorio para los archivos de Docker con *mkdir*, se introduce en la terminal *mkdir ~/bddspace*, se ingresa al directorio introduciendo *cd ~/bddspace* y con *nano* se crea un documento de texto, se introduce



Fig. 12 Creando e ingresando al directorio para los archivos de docker de la base de datos

* + Introducimos el comando *nano docker-compose.yml* e ingresamos el contenido de la figura 13

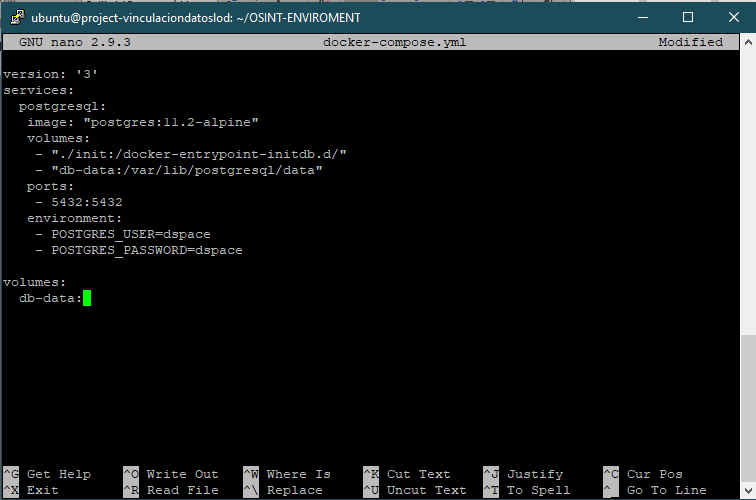


Fig. 13 Creando archivo docker-compose.yml

* + Presione *Ctrl+s* para guardar y *Ctrl+x* para salir. Cree otro directorio con nombre *init* introduciendo el comando *mkdir init* en la terminal, luego ingrese a dicho directorio con el comando *cd init*, con el comando *nano init\_pgcrypto.sh* creará un nuevo archivo en el cual debería adicionar el contenido de la figura 14

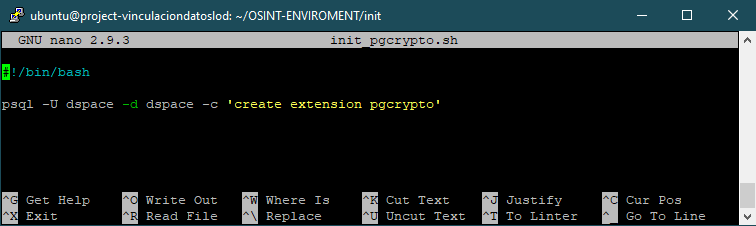


Fig. 14 Contenido del archivo init\_pgcrypto.sh

* + Presione *Ctrl+s* para guardar y *Ctrl+x* para salir, por ultimo introduzca el comando *cd ..* para ir al directorio anterior.
* Inicie la base de datos escribiendo en la terminal *sudo docker-compose up -d.*



Fig. 14 Inicio exitoso de la base de datos para dspace

# ¿CÓMO CONFIGURAMOS DSPACE Y FUSEKI?

* Ejecutamos el comando *sed -i 's+dspace.dir = /dspace+dspace.dir = /opt/dspace+g' dspace-6.3-src-release/dspace/config/dspace.cfg* este comando se utiliza para buscar líneas de un archivo y reemplazarlas por otras, en este caso se busca la línea *dspace.dir = /dspace* que viene por defecto y se reemplaza por *dspace.dir = /opt/dspace* en el archivo *dspace-6.3-src-release/dspace/config/dspace.cfg*. Aquí se le dice a Dspace donde deberá generar los archivos que se generarán más adelante. Para este punto en adelante si un comando le presenta problemas por permisos ejecute *sudo su* solo una vez, pero tenga precaución de ingresar todo correctamente.

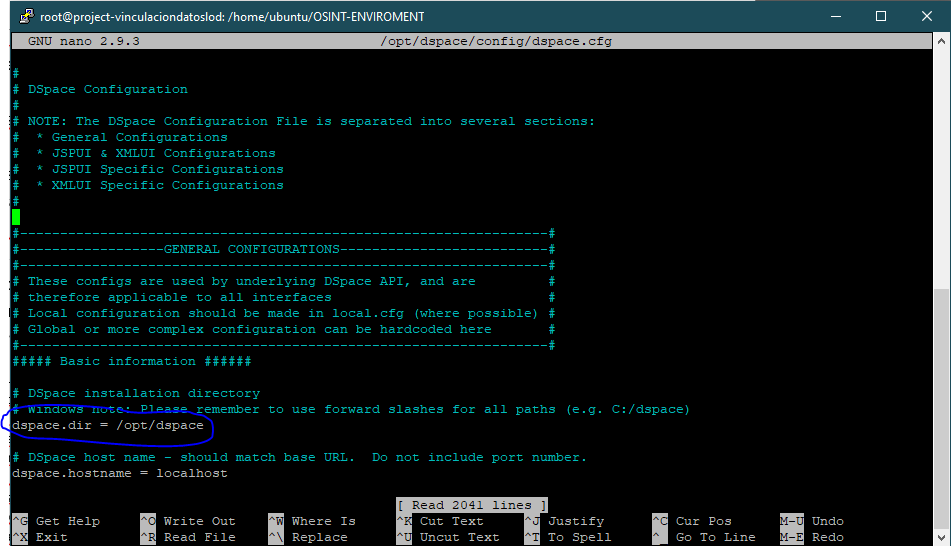


Fig. 15 Resultado de la ejecución del comando sed

* Ejecutar *mkdir /opt/dspace*. El comando *mkdir* nos permite crear carpetas, en este caso dentro de la carpeta opt estamos creando otra carpeta llamada dspace.

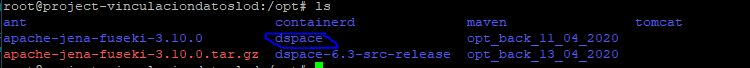


Fig. 16 Directorio dspace creado

* Ejecutamos *cd /opt/dspace-6.3-src-release/dspace/target/dspace-installer* con lo que nos ubicaremos en la carpeta donde podemos iniciar la generación de archivos



Fig. 17 Ubicados sobre dspace-installer

* Se introduce el comando *ant fresh\_install* lo cual podría llevar un largo tiempo dependiendo de la capacidad de su máquina, luego nuevos archivos aparecerán en la carpeta */opt/dspace* los cuales podemos ver ingresando al directorio con *cd /opt/dspace* e introduciendo *ls*

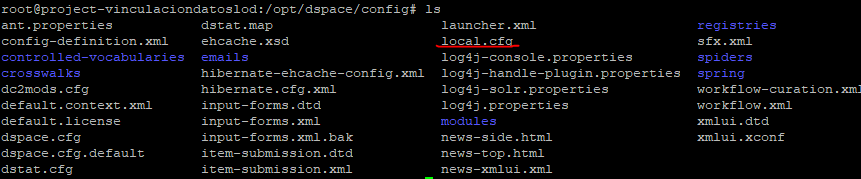


Fig. 18 Contenido generado en la carpeta dspace

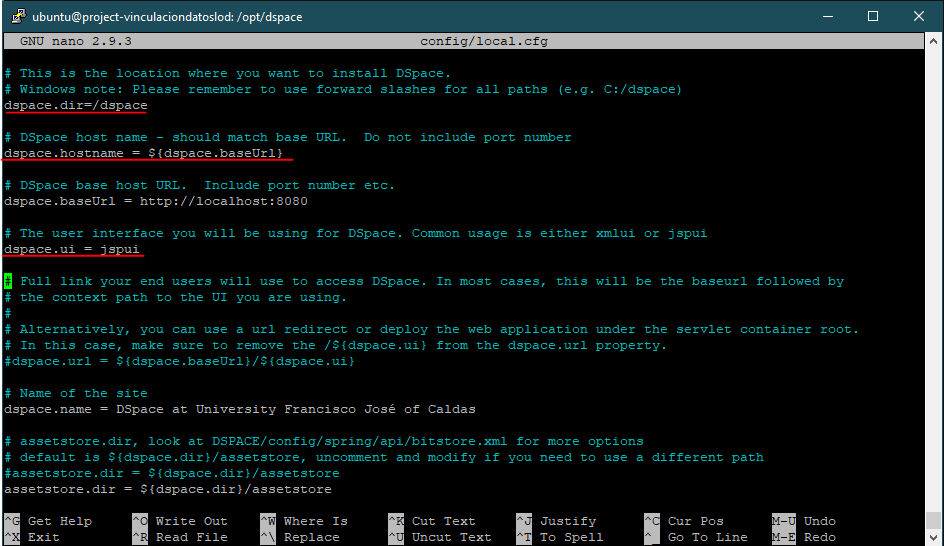
* Dspace tiene varios archivos de configuración que deben ser ajustados, estos ajustes se mostraran paso a paso
  + Se ejecuta *ln -s /opt/dspace/webapps/\* /opt/tomcat/webapps/* esto creará un enlace entre las carpetas dentro de webapps en Dspace a la carpeta webapps en tomcat, lo cual se puede comprobar de acuerdo a la figura 19 (*cd /opt/tomcat/webapps* luego *ls*)

Fig.19 Contenido webapps, las carpetas en azul aguamarina son aquellas que vienen desde dspace

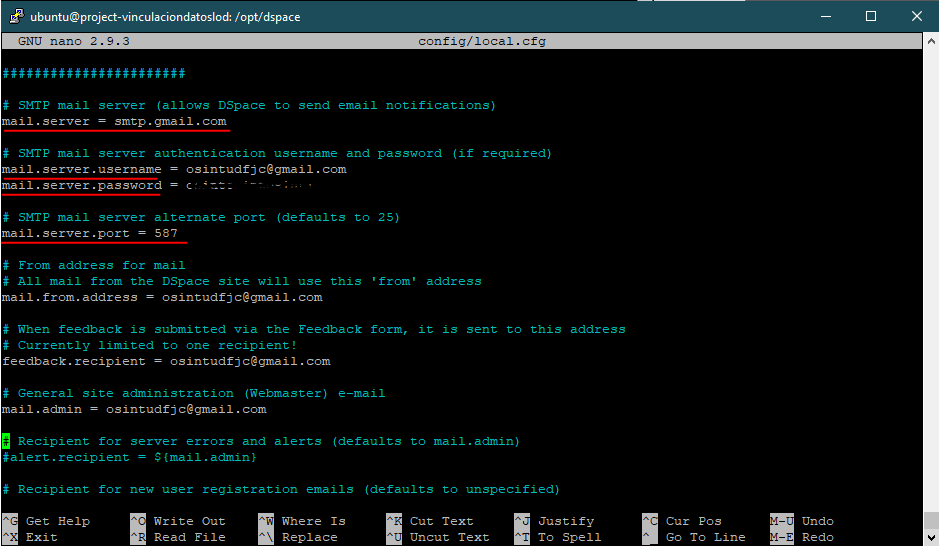
* + Se ejecuta *mv /opt/dspace/config/local.cfg.EXAMPLE /opt/dspace/config/local.cfg* para cambiar de nombre al archivo por defecto de *dspace local.cfg.EXAMPLE* a *local.cfg*

Fig. 20 Resultado luego de cambiar el nombre del archivo

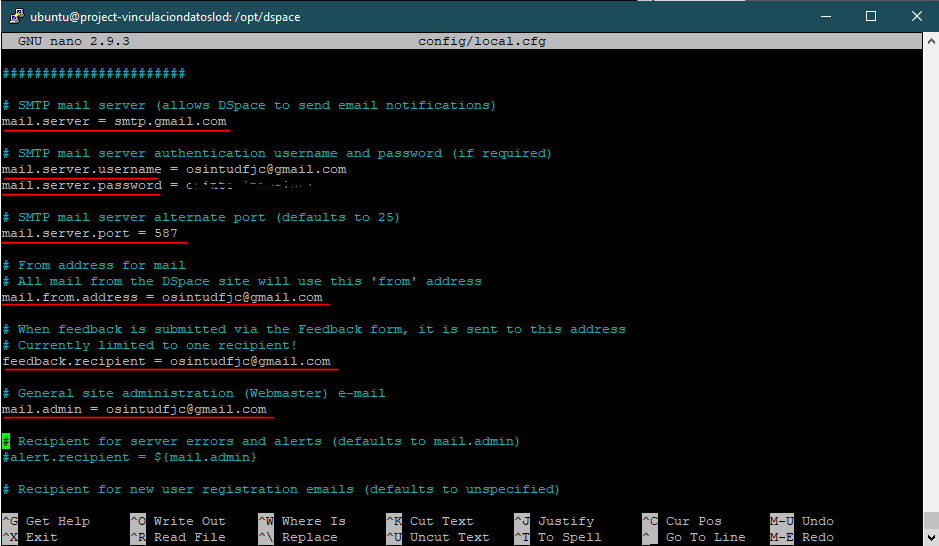
* + Ejecutamos *sed -i 's+dspace.dir = /dspace+dspace.dir = /opt/dspace+g' /opt/dspace/config/local.cfg, sed -i 's+dspace.hostname = localhost+dspace.hostname = ${dspace.baseUrl}+g' /opt/dspace/config/local.cfg*  y *sed -i 's+dspace.ui = xmlui+dspace.ui = jspui+g' /opt/dspace/config/local.cfg* como se mencionó antes *sed* busca líneas y las reemplaza, en este caso se cambia la dirección del directorio de dspace por */opt/dspace*, el nombre del host por *${dspace.baseUrl}* y se cambia la interfaz por defecto de *xmlui* a *jspui.* Puede abrir el archivo introduciendo *nano /opt/dspace/local.cfg* y comprobar los cambios.

Fig. 21 Comprobando cambios

* + *sed -i 's+#mail.server = smtp.example.com+mail.server = smtp.gmail.com+g' /opt/dspace/config/local.cfg, sed -i 's+#mail.server.username = myusername+mail.server.username = osintudfjc@gmail.com+g' /opt/dspace/config/local.cfg, sed -i 's+#mail.server.password = mypassword+mail.server.password = xxxxxxxxxx+g' /opt/dspace/config/local.cfg, sed -i 's+#mail.server.port = 25+mail.server.port = 587+g' /opt/dspace/config/local.cfg*.Esta configuración también se realiza sobre local.cfg en este caso se configura el servidor de correos utilizando SMTP de GMAIL a través del puerto 587 y se introduce el correo que será utilizado para el envío de correos en este caso[*osintudfjc@gmail.com*](mailto:osintudfjc@gmail.com)*.*

Fig. 22 Configuración de coreo

* + Configuramos otras características asociadas al correo con *sed -i 's+#mail.from.address = dspace-noreply@myu.edu+mail.from.address = osintudfjc@gmail.com+g' /opt/dspace/config/local.cfg, sed -i 's+#feedback.recipient = dspace-help@myu.edu+feedback.recipient = osintudfjc@gmail.com+g' /opt/dspace/config/local.cfg, sed -i 's+#mail.admin = dspace-help@myu.edu+mail.admin = osintudfjc@gmail.com+g' /opt/dspace/config/local.cfg*

Fig. 23 Otras configuraciones de correo

* + Cambios el texto de introducción de la página con *sed -i 's+dspace.name = DSpace at My University+dspace.name = DSpace en Universidad Distrital FJC+g' /opt/dspace/config/local.cfg*

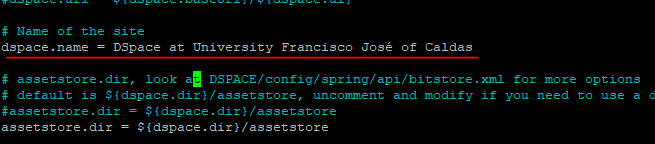


Fig. 24 Cambiar texto de introducción

* + Con *echo* y *>>* agregamos texto al final de dspace.cfg introduciendo echo 'mail.extraproperties = mail.smtp.socketFactory.port=587, \' >> /opt/dspace/config/dspace.cfg, echo ' mail.smtp.starttls.enable = true, \' >> /opt/dspace/config/dspace.cfg, echo ' mail.smtp.socketFactory.fallback=false, \' >> /opt/dspace/config/dspace.cfg, echo ' mail.smtp.starttls.required = true' >> /opt/dspace/config/dspace.cfg. Esto configurará Dspace para que utilice TLS en vez de SSL que es más antiguo, está por defecto y es más inseguro. Hasta este punto se ha termina las configuraciones sobre los archivos principales de configuración de DSPACE.
* Por último, ejecutamos *cp /opt/apache-jena-fuseki-3.10.0/fuseki.war /opt/tomcat/webapps/* con lo cual fuseki también ya estará configurado.



Fig. 25 Agregando fuseki a tomcat, la carpeta fuseki de esta figura se agregará automáticamente la primera vez que se ejecute tomcat

# AÑADIENDO ARCHIVOS DE CONFIGURACIÓN PREAJUSTADOS

Algunos de los archivos dentro de la configuración de Dspace en el momento de la realización de esta guía están siendo continuamente por lo tanto son almacenados en un repositorio para que sean copiados y reemplazados en instalaciones nuevas de Dspace. Diríjase al directorio home del usuario *cd ~/* e introduzca *git clone* [*https://github.com/AlienX456/DSPACE-CONFIGURACION.git*](https://github.com/AlienX456/DSPACE-CONFIGURACION.git)en la terminal esto descargara una carpeta ingrese con *cd* *DSPACE-CONFIGURACION* y con *ls* observe su contenido.

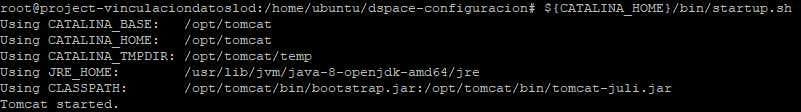
Fig. 26 Contenido de la carpeta

Estos son los datos de configuración o presentación más importantes, todos deben copiarse a diferencia de dspace.cfg y local.cfg que ya en este punto están configurados, para ello introduzca los siguientes comandos, tome en cuenta la ruta donde serán reemplazados. Recuerde que si tiene problemas de permisos debe ejecutar *sudo su* una sola vez y continuar con precaución

* *cp -r controlled-vocabularies /opt/dspace/config/*
* *cp -r rdf /opt/dspace/config/modules/*
* *cp Messages.properties /opt/tomcat/webapps/jspui/WEB-INF/classes/*
* *cp input-forms.xml /opt/dspace/config/*
* *cp news-top.html /opt/dspace/config/*
* *cp news-side.html /opt/dspace/config/*
* *cp dspace-logo-only.png /opt/tomcat/webapps/jspui/image/*
* *cp logo.gif /opt/tomcat/webapps/jspui/image/*

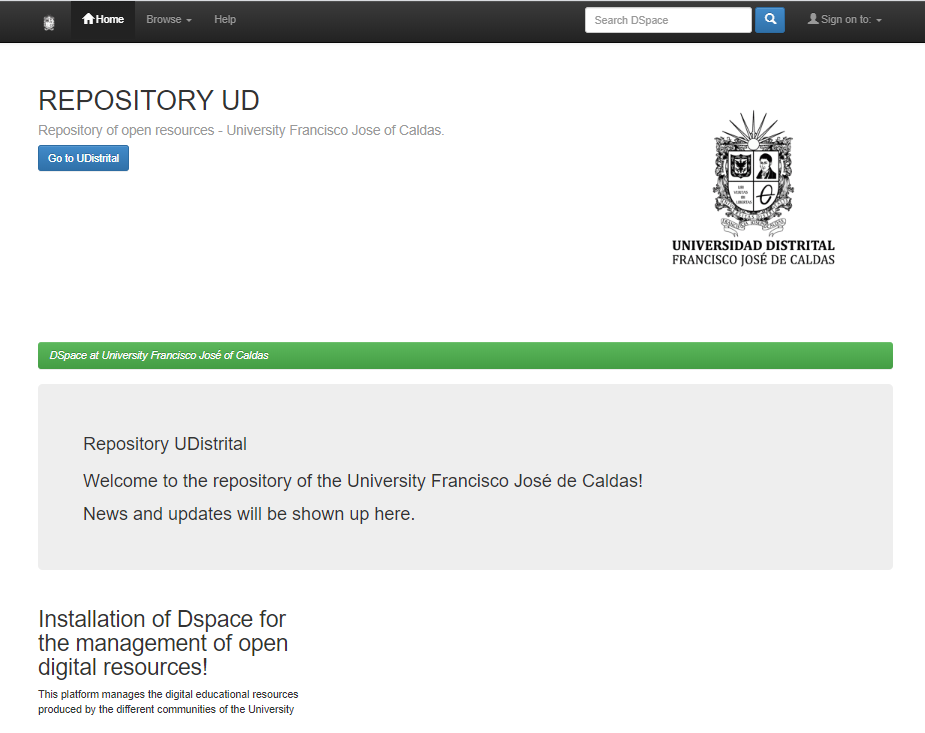
# ¿QUÉ ES TOMCAT Y COMO LO INICIAMOS?

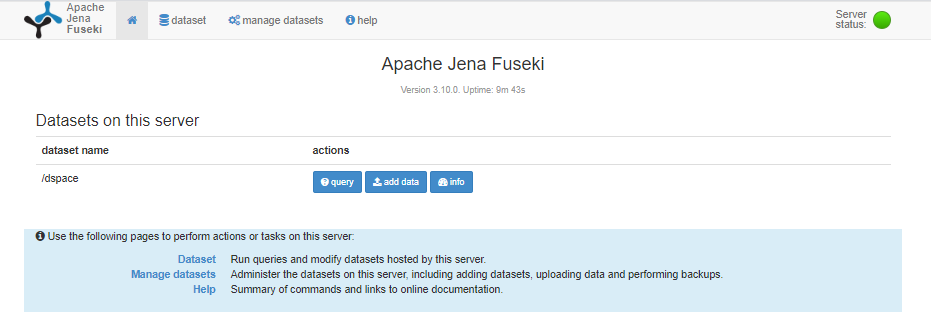
TOMCAT es un software que nos permite iniciar aplicaciones y accederlas desde direcciones web, para iniciar TOMCAT escribimos ${CATALINA\_HOME}/bin/startup.sh, el proceso de inicio puede tardar varios minutos dependiendo de la capacidad de su máquina. Para detener el servicio puede ejecutar ${CATALINA\_HOME}/bin/shutdown.sh. Recuerde crear el usuario administrador de DSPACE ejecutando *sh /opt/dspace/bin/dspace create-administrator* e ingresando los datos requeridos.

Fig. 27 Tomcat iniciado

# ACCEDIENDO A DSPACE Y FUSEKI

Si toda la configuración fue realizada de acuerdo a la guía anterior debería observar una página como la siguiente al acceder a *http://localhost:8080/jspui* si realizo la configuración en su computadoro en el caso del servidor se ingresa por medio de [*http://200.69.103.29:26237/jspui*](http://200.69.103.29:26237/jspui)*.* Para acceder a Fuseki reemplaze *jspui* por *fuseki.*

Fig. 28 DSPACE

Fig. 29 Fuseki

# CONFIGURACIÓN DE DSPACE (EXPANSIÓN PUNTO 8)

En el punto 8 de la guía se observan una serie de archivos ya configurados que pueden reemplazarse en las carpetas de dspace, a continuación, se mostrara con más detalle la configuración realizada y sus efectos.

* **news-top.html**:Se halla en la ruta */opt/dspace/config/* se modifica de la siguiente manera y su resultado se muestra en la figura 31

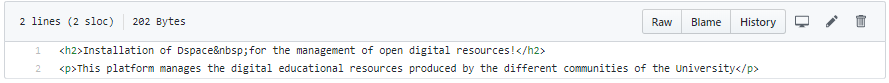


Fig. 30 News-top.html configuración

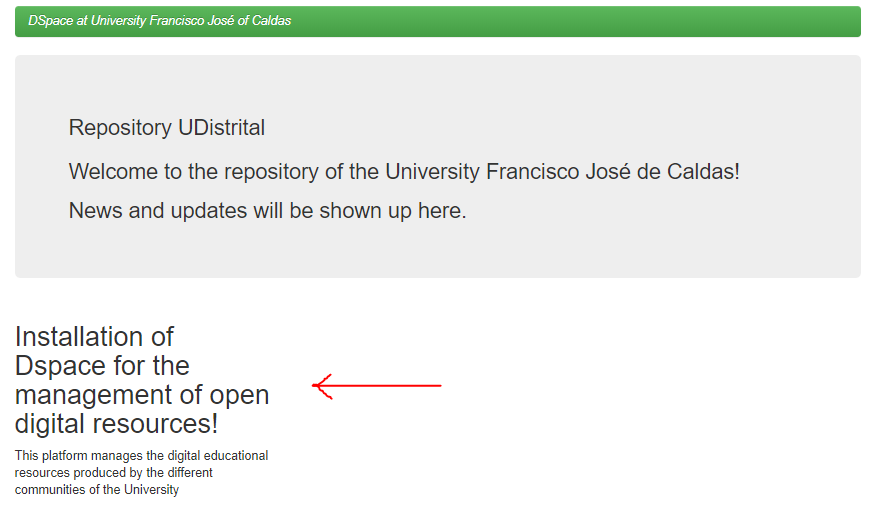


Fig. 31 News-top.html resultado

* **news-side.html:** Se halla en la ruta */opt/dspace/config/* se modifica de la siguiente manera y su resultado se muestra en la figura 33:

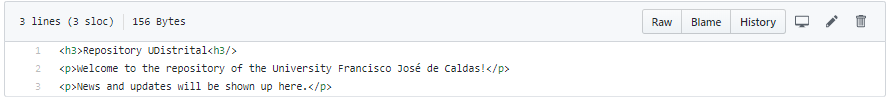


Fig. 32 news-side.html Configuración

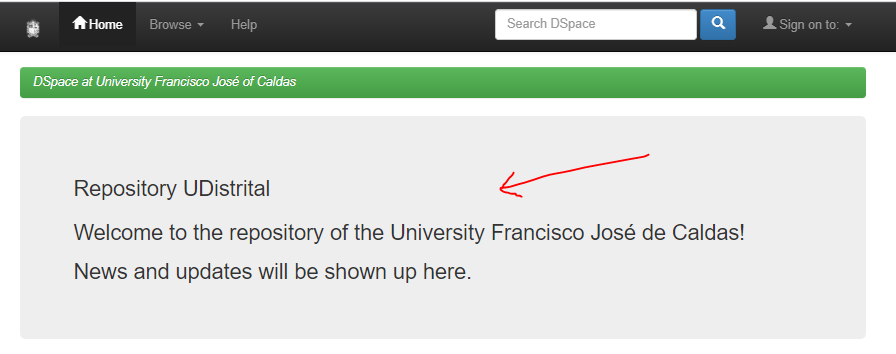


Fig. 33 news-side.html resultado

* **Logos:** En el repositorio existen dos logos asociados a la universidad distrital que reemplazan los logos por defecto de dspace, ambos logos están en la carpeta */opt/tomcat/webapps/jspui/image/* con los nombres *dspace-logo-only.png* y *logo.gif*

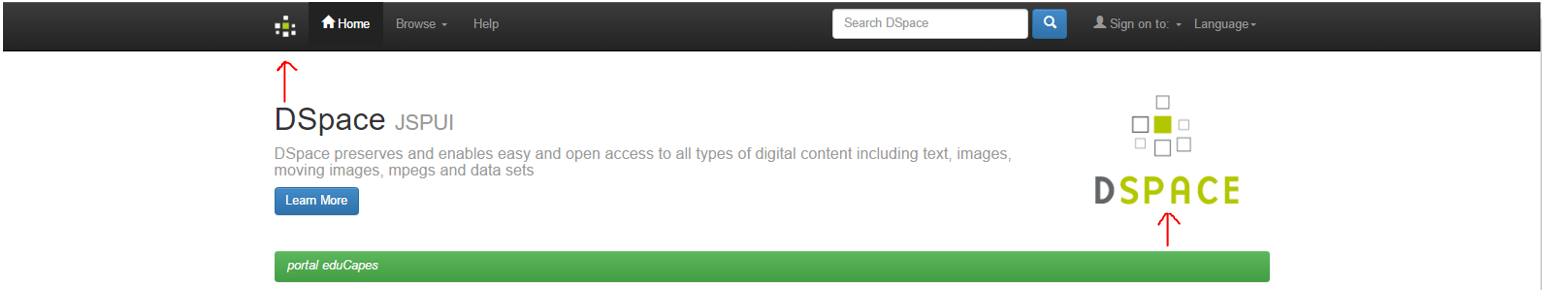


Fig. 34 Logos por defecto

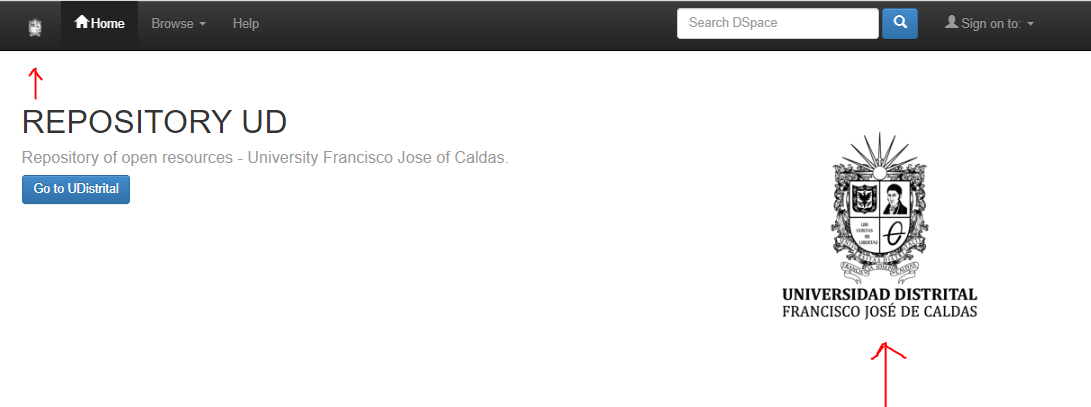


Fig. 35 Logos utilizados

* **Messages.properties:** En este archivo cambiamos la presentación de los metadatos registrados presentes en DSPACE, en este caso se modifica metadata asociada al dublincore:



Fig. 36 Configuración de Messages.properties



Fig. 37 Resultado de configuración Messages.properties

* **input-forms.xml:** Ubicado en */opt/dspace/config/* este archivo permite modificar algunos atributos de la metada añadiendo valores cerrados, descripción, obligatoriedad, ayudas de la misma lo cual será observado al momento de registrar un recurso académico, en la figura 38 pueden observarse tres metadatos del dublincore, en los tags puede observarse que la metadata es obligatoria (<required>), su nombre (<dc-element>), label asociado (<label>), ayuda (<hint>), tipo de presentación (<input-type) entre otros:

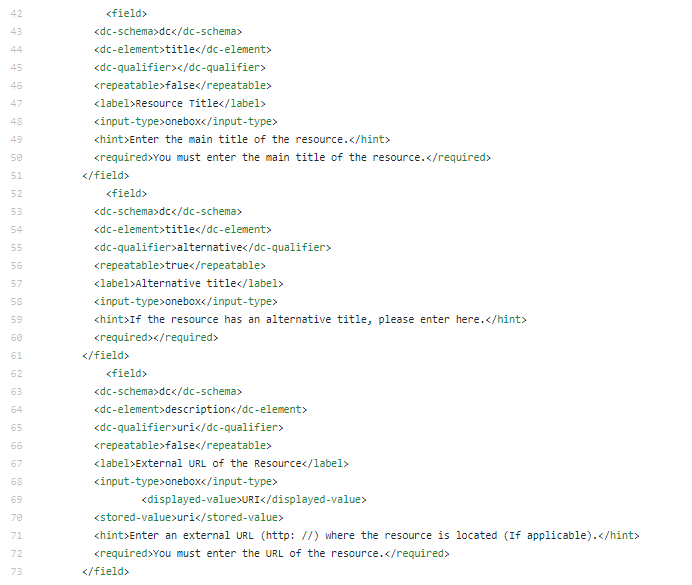


Fig. 38 XML del input-forms

Otros metadatos también incluyen vocabulario como el ejemplo en la figura 39

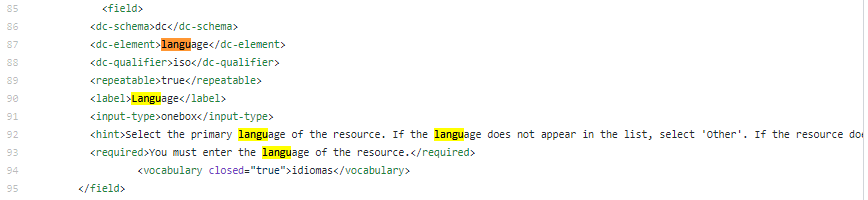


Fig. 39 Metadata subject asociado a un vocabulario idiomas

* **controlled-vocabularies:** Es una carpeta en la cual se encuentran los vocabularios controlados los cuales son llamados desde el input-forms.xml, estos archivos son de formato xml donde se define de forma jerárquica una serie de objetos a los que se podría asociar el recurso, entre ellos se encuentran:
  + audiencia.xml
  + contextoeducativo.xml
  + contribuyentes.xml
  + controlledvocabulary.xsd
  + editores.xml
  + formatos.xml
  + grupo.xml
  + idiomas.xml
  + institucion.xml
  + niveleducativo.xml
  + nivelinteractividad.xml
  + nsi.xml
  + srsc.xml
  + tipointeractividad.xml
  + tiporecursos.xml

En la figura 39 puede observase que el metadato *language* del dublincore está asociado con el vocabulario *idiomas*, que hace parte de los archivos mencionados anteriormente:

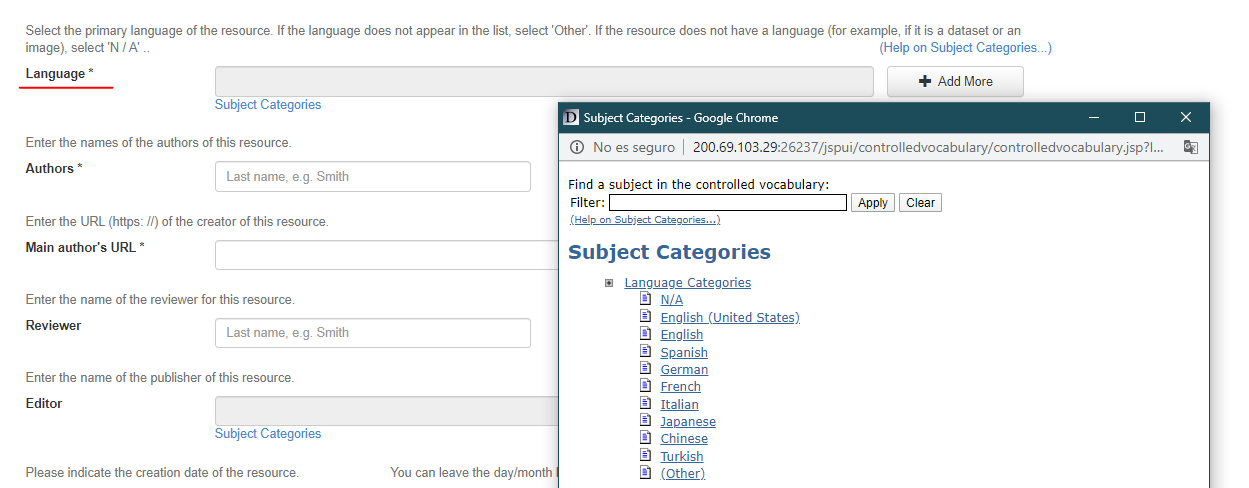


Fig. 40 Ejemplo de un metadato tipo one-box, requerido (\*), junto con su vocabulario definido en *idiomas.xml*

* **rdf:** En una carpeta donde se encuentran los archivos correspondientes al mapeo de elementos dc, en la figura 41 se encuentra un elemento del input-forms llamado *audience*, el cual se mapea con el *metadata-rdf-mapping.xml* que está en la carpeta rdf de acuerdo con la figura 42:

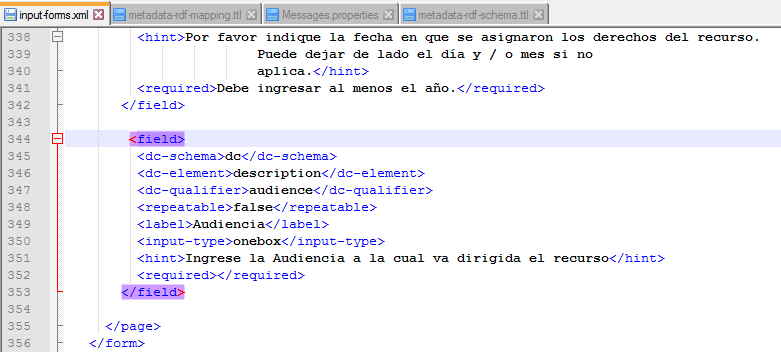


Fig. 41 Audience en input-forms.xml

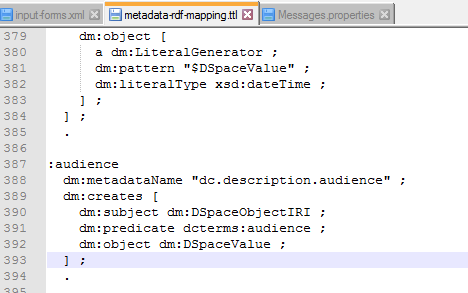


Fig. 42 metadata-rdf-mapping.xml campo audiencia

En la figura 42 se observa cómo se mapeará este campo (audience) al respectivo lenguaje, en la figura 43 el campo es agregado al archivo messages.properties

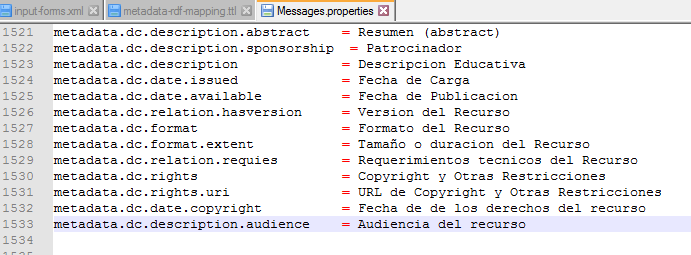


Fig. 43 El campo se añade al messages.properties

# AGREGANDO METADATA A DSPACE

* Una vez iniciada sesión con el administrador vaya a la sección de administrador

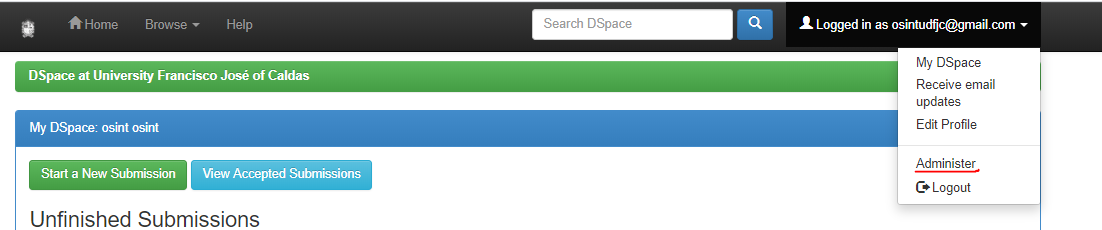


Fig. 44 Paso 1 agregando metadata

* Haga click en General Settings y Metadata Registry

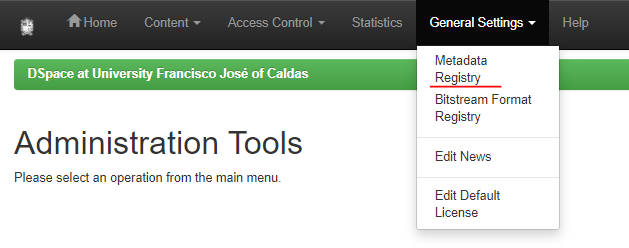


Fig. 45 Paso 2 agregando metadata

* Haga click en dublincore

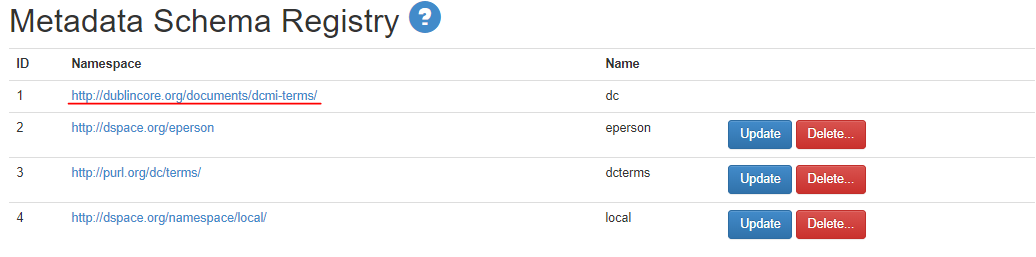


Fig. 46 Paso 3 agregando metadata

* Deslícese hasta el final de la página y llene el formulario con los datos requeridos y haga clic en *Add New*

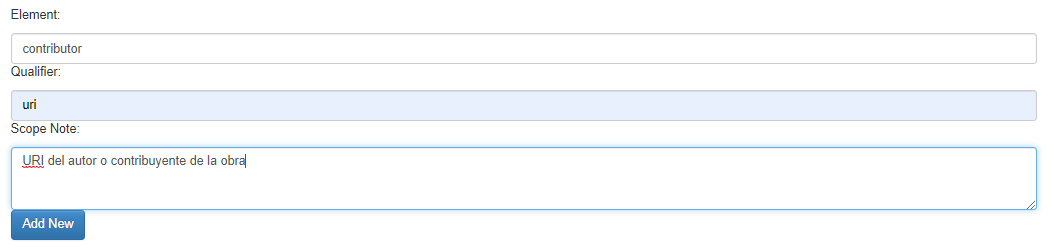


Fig. 47 Paso 4 agregando metadata

* Verifique que el campo haya sido añadido

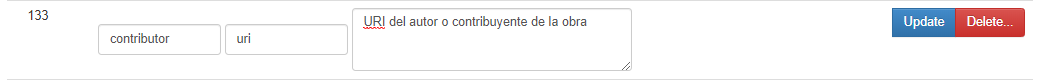


Fig. 48 Paso 5 agregando metadata

# FUSEKI, INTERFAZ Y CONSULTAS

Al ingresar a fuseki podrá observa una interfaz como la siguiente, verifique que el servidor esté funcionando correctamente verificando el server status, si está en color verde todo está correctamente, color rojo indica algún fallo

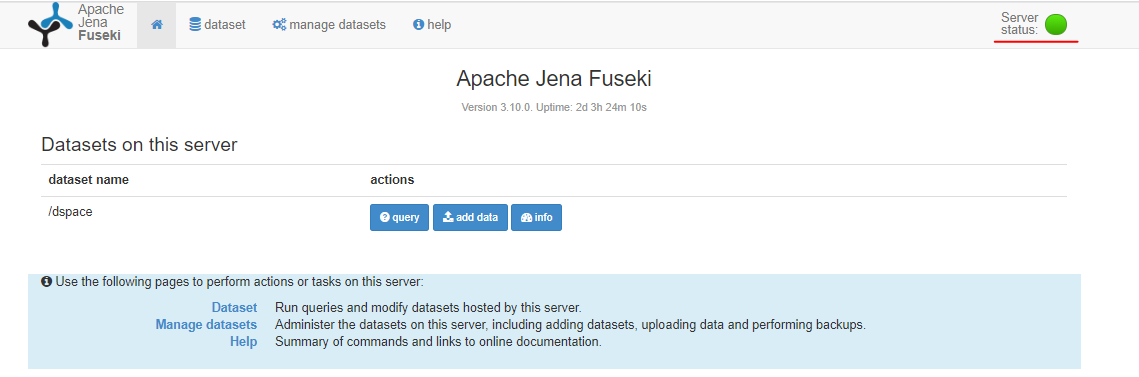


Fig. 49 Interfaz principal de fuseki, server status OK

En la interfaz como se indica en la figura 49 existe una tabla con los nombres de los datasets, usted puede crear un nuevo dataset en *manage datasets* en este caso ya hemos creado uno con el nombre de dspace, a continuación, haga clic en el botón info del dataset /dspace, aparecerá la ventana de la figura 50

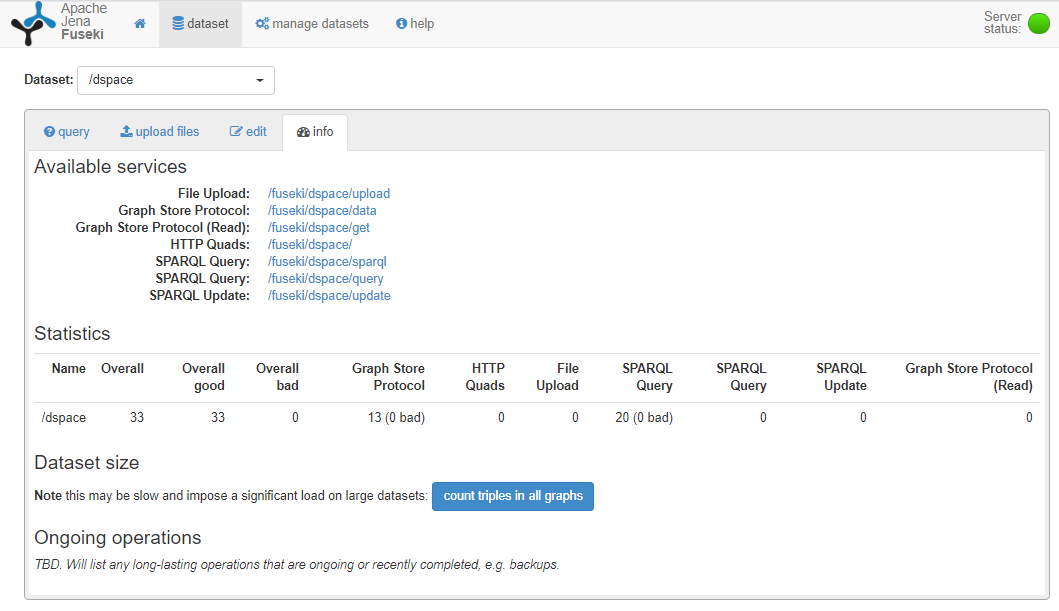


Fig 50. Fuseki información del dataset

Haga clic en *count triple in all graphs* esto verificara el número de datos por grafo, en caso de salir en 0 deberá ejecutar *cd /opt/dspace/bin* y *sudo dspace rdfizer –c –v*, RDFizer se puede utilizar para eliminar todos los contenidos, para reconvertir todos los contenidos y actualizar los contenidos. Debería salir un resultado donde indique que existen grafos

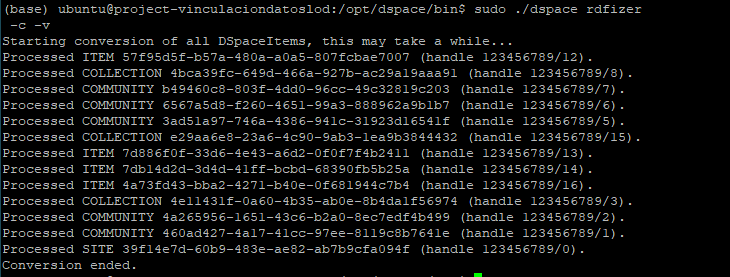


Fig. 51 RDFizer

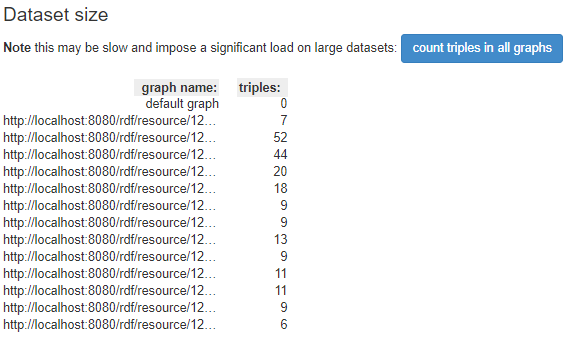


Fig. 52 Número de datos por grafo del dataset

Diríjase a la pestaña query

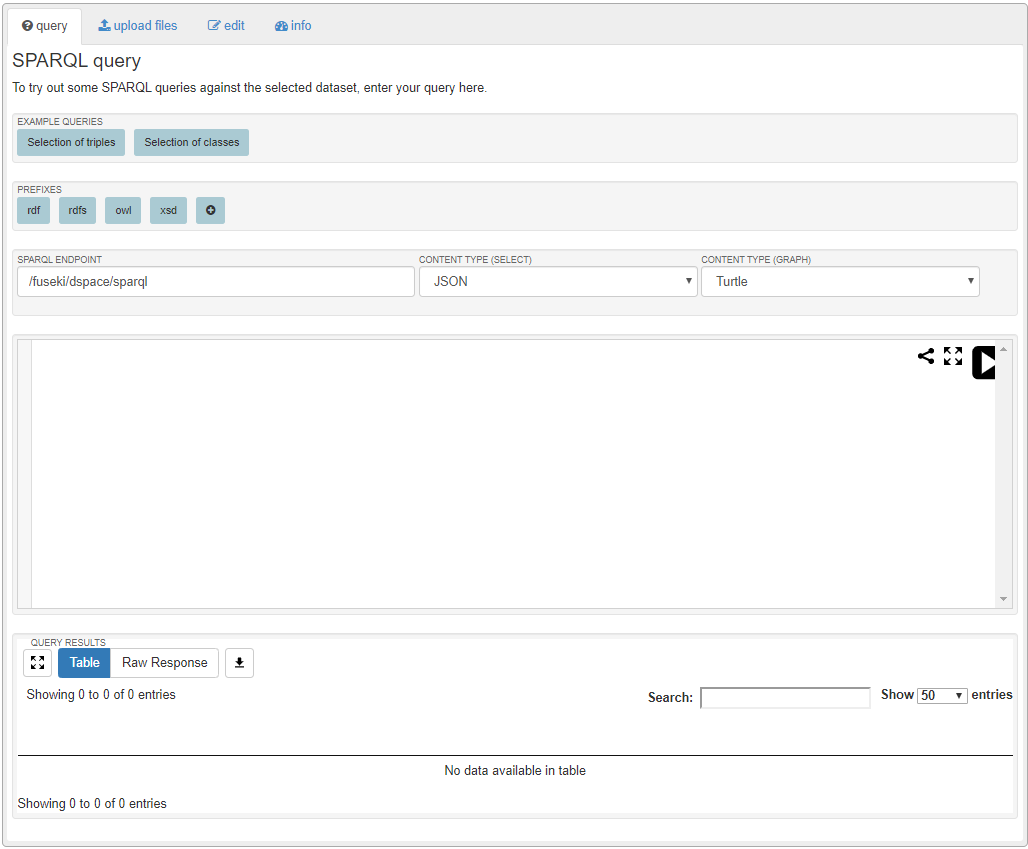


Fig. 53 Interfaz de query SPARQL de fuseki

Aquí podrá ejecutar consultas tipo SPARQL sobre el dataset seleccionado como la



Fig. 53 Consulta de ejemplo sobre Dspace (Todos los títulos del grafo)

La consulta de la figura 53 en este caso es los nombres de todos los títulos almacenados en la base de datos de dspace como se muestran en la figura 54

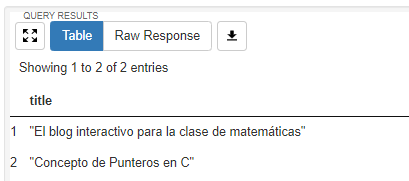


Fig 54 Resultados de la consulta SPARQL en fuseki