



UNIVERSIDAD DE HUELVA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

**Solución integral automatizada a los TFG de la
Universidad de Huelva a través del Repositorio
Institucional Arias Montano**

Montserrat Soler Bernal

Título del Proyecto

Solución integral automatizada a los TFG de la Universidad de Huelva a través del Repositorio Institucional Arias Montano

Autor

Montserrat Soler Bernal

Titulación

Master en Tecnologías Informáticas Avanzadas

Director

Jacinto Mata Vázquez

Defensa

Huelva, 28 de Septiembre de 2012

Agradecimientos

A mis compañeros Rafaela y José Carlos, sus ideas y continuas revisiones, son las que han dado a luz y han hecho posible este proyecto. Espero que la puesta en marcha de él, les facilite el buen trabajo que siempre desarrollan.

A mi compañero Ángel, que siempre está presente intentando alisar el camino de mis estudios y mi trabajo.

A mi compañera Inma, por su corrección ortográfica y gramatical, aplicando los conocimientos adquiridos en sus cursos de formación.

A mis familiares y amigos, por el continuo apoyo durante este año de encierro dedicado al master.

A D. Jacinto Mata Vázquez, deseo expresar mi más sincero agradecimiento por la confianza depositada en mí al recibir de gran agrado el proyecto que le presenté, aceptando ser mi director. Gracias por la orientación, el seguimiento y la supervisión continua del mismo durante todos estos meses en los que se ha desarrollado este trabajo fin de master.

Gracias a todos por estar ahí.

ÍNDICE

1. INTRODUCCIÓN

1.1. Descripción de la situación actual	6
1.2. Objetivos	7
1.3. Organización de la memoria	8

2. PROPUESTA DE SOLUCIONES

2.1. Descripción de las propuestas planteadas	9
Propuesta 1	9
Propuesta 2	11
Propuesta 3	12
Propuesta 4	14
Propuesta 5	15
2.2. Diagramas de flujo de las propuestas planteadas	16
2.3. Justificación de la propuesta adoptada	20
2.4. Planificación del desarrollo del proyecto	22

3. DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN ELEGIDA

3.1. Descripción del servidor del Repositorio	27
3.2. Descripción del software Dspace	28
3.3. Preparar el entorno de Dspace para almacenar TFG	30
3.4. Creación de una aplicación web	31
3.5. Integración de la aplicación web con LDAP	33
3.6. Integración de la aplicación web con la base de datos de Dspace	34
3.6.1. Conexión a la BD de Dspace	34
3.6.2. Esquema de la BD de Dspace	34
3.6.3. Extracción de consultas	35
1.- Consulta: tipo de proyecto a subir	36
2.- Consulta: Titulación	38
3.- Consulta: Director/Codirector	39
4.- Consulta: Listado de proyectos pendientes de subir	42
5.- Insertar: datos confirmados por el alumno	45
6.- Actualizar: datos confirmados por el director	45

7.- Retirar el ítem subido al repositorio	46
3.6.4. Crear tabla en Dspace para almacenar TFG	48
3.7. Integración de la aplicación web con Dspace	49
3.7.1. Requerimientos	49
3.7.2. Instrucción y parámetros	50
3.7.3. Programación	52
3.8. Modelo Entidad-Relación	58
3.9. Validación del proyecto: probando la solución propuesta	60
3.9.1. Workflow del alumno	60
3.9.2. Workflow del director	70
3.10. Informe final con cuestiones que deben mencionarse	77
4. CONCLUSIONES Y LÍNEAS DE TRABAJOS FUTURAS	
4.1. Conclusiones	82
4.2. Líneas de Trabajo Futuras	84
4.3. Normativa de entrega de Trabajos Fin de Grado	86
5. BIBLIOGRAFÍA	87
6. ANEXOS	
Anexo 1: Oferta académica en Grados	88
Anexo2: Formulario para los TFG	89
Anexo 3: Conexión del Formulario con las Colecciones	92
Anexo 4: Definir los datos del formulario TFG que deben visualizarse	93
Anexo 5: Crear un grupo de usuarios por cada colección creada	93
Anexo 6: Conexión al LDAP de la Universidad de Huelva	95
Anexo 7: Crear conexión a BD	98
Anexo 8: dublin_core.xml en /dspace/FichImportar/idproyecto/Item	99
Anexo 9: Subir ítem al repositorio.....	101

1. Introducción

Este proyecto intenta ofrecer una solución integral automatizada a la entrega y almacenamiento de los Trabajos Fin de Grado de la Universidad de Huelva, tomando como elemento contenedor de los proyectos, el Repositorio Institucional Universitario Arias Montano.

1.1. Descripción de la situación actual

Actualmente, la Universidad de Huelva ofrece una oferta académica amplia dentro de sus Grados, Ciclos y Posgrados. Los estudios de Grado, se iniciaron hace un par de años, por lo que el próximo año será el primero en el que se finalice un ciclo completo de esta especialidad. Consecuentemente, en 2013 se presentarán los primeros Trabajos Fin de Grado elaborados en el seno de esta Universidad.

La memoria que aquí se expone, intenta ofrecer una forma informatizada de entrega de estos Trabajos y, como aún no existe una normativa elaborada, se realizará una equivalencia con la entrega de los Trabajos Fin de Carrera de la ETSI, partiendo del planteamiento lógico que la normativa a crear será similar.

En la actualidad, el proceso de entrega, defensa y calificación de los Trabajos Fin de Carrera no está informatizado y son varias las fases indicadas en su normativa que deben llevarse a cabo, por lo que pasamos a detallar el proceso que hoy en día se desarrolla sin intervención de un flujo de trabajo telemático.

El proceso comienza con la entrega por parte del alumno y la aceptación por parte del director asignado del Trabajo Fin de Carrera, a partir de ahora TFC. El alumno encuaderna y presenta tres copias al director para distribuir entre los miembros del tribunal. A continuación, el alumno se dirige a la Secretaría, donde entrega una copia en CD para ser remitida a la Biblioteca donde será catalogada y puesta a disposición de los alumnos para su consulta.

El director asignado al TFC solicita a su Secretaría el Acta mediante un impreso denominado "Propuesta de defensa de trabajo fin de carrera" (<http://www.uhu.es/etsi/impresos/propuestaDef.pdf>). Desde la recepción de ese documento, la Secretaría podrá generar el Acta de Evaluación que se rellenará el día de la defensa.

El día de la defensa del TFC ante el tribunal, el Acta será recogida por el director. Tras la defensa, evaluación y calificación del trabajo, el director entregará el Acta nuevamente en la Secretaría para que la calificación figure en el expediente del alumno.

1.2. Objetivos

Como se ha indicado, el trabajo trata de informatizar el proceso tomando como referencia los TFC, ofreciendo así un carácter más oficial a la presentación de los Trabajos Fin de Grado (de aquí en adelante TFG). De esta manera, tanto el alumno como el director ofrecen a la Universidad los datos necesarios mediante un procedimiento electrónico informatizado, sin necesidad de rellenar en la Secretaría ningún formulario y sin presentar copia digitalizada en CD para su envío a la Biblioteca.

Se podría indicar que el **objetivo principal** del trabajo defendido en esta memoria pretende, *“Mejorar el proceso de entrega del TFG, informatizando toda la información que tiene que fluir entre alumno, director, Secretaría y Biblioteca y al mismo tiempo ofreciendo una forma directa de almacenarlo en el Repositorio Institucional de la UHU, quedando así catalogado para su posterior consulta entre los usuarios del Repositorio”*.

Además del objetivo mencionado, si este trabajo fuera incluido como parte de la nueva normativa que se elaborará para los TFG de la UHU, se alcanzarían paralelamente los siguientes **objetivos específicos**:

- Evitar que el alumno entregue su memoria en formato digitalizado en CD a la Secretaría para su posterior distribución a la Biblioteca.
- Optimizar el proceso de distribución del TFG entre alumno, director, la Secretaría y la Biblioteca.
- Ahorro de espacio físico en la Biblioteca, encargada de la custodia de los TFG.
- Evitar el trabajo de catalogación por parte de la Biblioteca, ya que los datos que vayan incorporando alumno y director en este proceso contribuirán al autoarchivo del trabajo.
- Incluir el proceso de entrega de TFG descritos en esta memoria en la nueva normativa a elaborar.
- Conseguir una mejora en la gestión, automatización y control de procesos.
- Impulsar la vía telemática como medio de presentación de TFC, TFG, ...
- Impulsar el Catálogo y/o Repositorio de la UHU, ofreciendo, a través de ellos, un acceso electrónico a los TFG.
- Cultura digital y acceso digital
- Superar cuestiones de espacio, acceso, préstamo, etc.
- Sostenibilidad ambiental: menos soporte digital, ahorro de espacio en la Biblioteca, ahorro económico por parte del alumno,...
- Difusión en recolectores y libre acceso.
- Buenas prácticas en comunicación del trabajo de investigación.
- Buenas prácticas en derechos y propiedad intelectual: Creative Commons.

1.3. Organización de la memoria

La memoria desarrollada en estas páginas, se organiza tal como se indica a continuación:

- el apartado en el que estamos, “**Introducción**”, se describe brevemente la situación actual y los objetivos que quieren alcanzarse con la propuesta de mejora.
- en el segundo apartado, *se describen distintas soluciones* que pueden desarrollarse para mejorar la situación actual. Se valoran las ventajas e inconvenientes de cada una de ellas. Basándose en las especificaciones dadas, se indicará según el autor de este trabajo, cual es bajo su punto de vista la mejor de las propuestas planteadas. Una vez elegida y justificada, se planea el desarrollo de sus fases.
- en el apartado tres de esta memoria, *se pone en marcha la solución elegida*, describiendo las situaciones de hardware y software necesarias para realizarse el planteamiento desarrollado. Se describen cada una de las fases que deben ejecutarse, se implementa el desarrollo, y se ejecuta la programación mostrando los resultados obtenidos.
- en el apartado cuarto, se muestran las *conclusiones y las líneas de trabajo futuras*.
- para concluir, se muestra la *Bibliografía* leída para la elaboración de este trabajo. Finalizando con el desarrollo de los *Anexos* elaborados durante la realización de esta memoria.

2. PROPUESTA DE SOLUCIONES.

2.1. Descripción de las propuestas planteadas

Son varias las soluciones planteadas durante el trabajo de investigación realizado en la elaboración de esta memoria. De todas las planteadas, se concluyó que el enfoque que se debería dar, tendría que abarcar, al menos, las siguientes expectativas:

- La creación de un proceso telemático que permita presentar los TFG.
- Incluir en ese proceso un WorkFlow (flujo de trabajo) que permita el paso de información entre alumno, director, tribunal, Secretaría y Biblioteca.
- Incluir un método de autoarchivo, que realice una catalogación directa sin la intervención de la Biblioteca.
- Elaborar un método, que permita ocultar el TFG catalogado, poniéndolo a disposición tan solo del tribunal que lo va a corregir.

Pasemos a detallar algunas de las propuestas, describiendo en qué hubieran consistido y ofreciendo al mismo tiempo las ventajas e inconvenientes encontrados durante sus implementaciones.

Propuesta 1

En esta propuesta se plantea usar un servidor Citrix Metaframe donde almacenar los TFG y crear un cliente ICA por cada TFG. Este cliente permite el acceso a la memoria del trabajo al director y a los miembros del tribunal.

¿En qué consiste el cliente ICA?

Las bases de datos, ficheros, ..., accesibles a través de web funcionan con diferentes sistemas técnicos. Algunos de estos sistemas requieren que el usuario que las quiera consultar se instale en su ordenador un pequeño programa llamado cliente, sin el que no se puede consultar la base de datos, el fichero, ...

Todas las bases de datos, ficheros, ..., que requieren la utilización de un software de consulta propio son instaladas en un servidor con el Sistema Metaframe de Citrix Systems.

Metaframe utiliza tecnología cliente/servidor, lo que hace que las bases de datos, ficheros, ..., sean accesibles desde diversas plataformas, Windows 95/98/NT/XP, MacIntosh, Unix, etc. Para hacer la conexión sólo es necesario instalarse en el ordenador un pequeño programa llamado Cliente ICA.

La filosofía de MetaFrame permite el uso de aplicaciones estándares en cualquier dispositivo. Estas aplicaciones se ejecutan en el servidor Citrix pero se visualizan en la pantalla del cliente.

Para tener acceso a las aplicaciones instaladas en Metaframe, se crea un cliente ICA por cada aplicación instalada, que en nuestro caso será uno por cada TFG almacenado en el servidor, y se ofrece un enlace de este cliente en una aplicación web para que el usuario pueda acceder a través de una conexión al servidor Citrix.

El proceso completo de esta propuesta se detalla a continuación:

- El alumno presenta al director del proyecto la memoria en formato digital. El director la remite a la Biblioteca para que sea alojado en un servidor Metaframe licenciado y localizado en la sala de servidores de la Biblioteca. La Biblioteca asociará a cada TFG almacenado en el servidor, para su referencia, un cliente ICA.
- Se creará una aplicación web que muestre un listado con todos los TFG publicados en el servidor Metaframe y enlazados al cliente ICA que permitirá la conexión al servidor Citrix. En este proceso, los componentes del tribunal de ese TFG tendrán acceso de lectura. Dicha identificación se realizará a través del LDAP de la UHU.
- Tras la identificación se mostrarán, únicamente, los proyectos que deben evaluar.
- El director se presentará en la Secretaría de su Centro como lo hace actualmente, para rellenar la "Propuesta de defensa de trabajo fin de grado". Con ella, la Secretaría generará el Acta de Calificación, que será recogida por el director el día de la defensa.
- Tras la defensa del TFG, el director informará a la Biblioteca su finalización y le indicará si el alumno permite el acceso público a su proyecto. Si es así, el documento subido al servidor Metaframe será catalogado en Columbus (catálogo de la Biblioteca Universitaria) y/o Repositorio y puesto a disposición de toda la comunidad universitaria para su consulta. Esta comunicación se llevará a cabo mediante un correo electrónico.

Al mismo tiempo, el director devolverá el Acta de Calificación a la Secretaría.

- Una vez concluida la defensa del trabajo, si el alumno no ha permitido su publicación en abierto, la memoria será eliminada del servidor Metaframe.

En este proceso intervienen el director de proyecto y el personal de la Biblioteca. Se debe tener claro a quien dirigirse ante cualquier problema.

Se exponen a continuación los inconvenientes encontrados en el desarrollo de esta propuesta:

- Se necesita un servidor con capacidad suficiente para alojar todos los TFG.
- Se necesita un sistema de backup para las copias diarias.
- Se necesita adquirir la licencia de Metaframe y resto de software para el servidor.

- Todo esto implica un gasto adicional tanto en las adquisiciones como en el mantenimiento anual.
- El director debe presentarse en la Secretaría para presentar la "Propuesta de defensa del trabajo fin de grado".

Se indican del mismo modo, las ventajas que podrían conseguirse si el modelo descrito se implantara:

- La Biblioteca ahorraría espacio físico de almacenamiento.
- Se evitaría la entrega del CD de la memoria en la Secretaría y su posterior traspaso a la Biblioteca, ya que ésta se transmite del alumno al director y del director a la Biblioteca de forma telemática.
- La consulta al proyecto por parte del tribunal y por parte de cualquier usuario si el TFG fuera publicado en abierto, sería más eficaz.

Propuesta 2

Esta propuesta es similar a la descrita en el apartado anterior, pero esta vez los TFG serán almacenados en un servidor del Servicio de Informática y Comunicaciones, el cual, poseerá una estructura específica para alojar los distintos trabajos generados en el seno de la Universidad de Huelva.

El proceso completo de esta propuesta se detalla a continuación:

- El alumno presenta en formato digital la memoria del TFG al director del proyecto.
- El director lo subirá a un servidor del SIC destinado a dicho fin donde tendrá, previa autorización, permiso específico para almacenarlo. Este servidor será licenciado, su estructura permitirá el almacenamiento organizado de los trabajos generados en la UHU y se ubicará en la sala de servidores del SIC.
- El servidor poseería la estructura lógica necesaria para que cada director pudiera subir el TFG. Para ello, una posible estructura sería crear una división por Facultad/Escuela, y dentro de éstas, todas las Titulaciones existentes en el Centro.

Todos los docentes pertenecientes a esa titulación, tendrían los permisos necesarios para poder alojar en esa ubicación los proyectos de sus alumnos.

- El director tras subir el trabajo presentado al servidor, comunicará al Tribunal la url de acceso del trabajo subido para que puedan consultarlo.
- El director, tal y como lo hace actualmente, se presentará en la Secretaría de su Centro para rellenar la "Propuesta de defensa de trabajo fin de grado". Con ella, la Secretaría generará el Acta de Calificación, que será recogida por el director el día de la defensa.
- Tras la defensa del TFG, el director informará a la Biblioteca que la defensa ha concluido y le indicará si el alumno permite el acceso público a su proyecto. Si es así, indicará a la Biblioteca la url del documento subido al servidor para ser catalogado en Columbus (catálogo de la Biblioteca Universitaria) y/o Repositorio y

puesto a disposición de toda la comunidad universitaria para su consulta. Esta comunicación se llevará a cabo mediante un correo electrónico.

Al mismo tiempo, el director devolverá el Acta de Calificación a la Secretaría.

- Si el alumno no diera su permiso, la memoria sería eliminada del servidor una vez finalizada la defensa del trabajo.

En este proceso intervendrían el director del TFG, el personal del SIC y el personal de la Biblioteca. Es necesario dejar claro a quien dirigirse ante cualquier problema.

Se exponen a continuación los inconvenientes encontrados en el desarrollo de esta propuesta:

- Se necesita un servidor con capacidad suficiente para alojar todos los TFG.
- Se necesita un sistema de backup para las copias diarias.
- Se necesita adquirir las licencias de software para el servidor.
- Todo esto implica un gasto adicional tanto en la adquisición como en el mantenimiento anual.
- El director debe presentarse en la Secretaría para presentar la "Propuesta de defensa del trabajo fin de grado".

Se indican del mismo modo, las ventajas que podrían conseguirse si el modelo descrito se implantara:

- La Biblioteca ahorraría espacio físico de almacenamiento.
- Se evitaría la entrega de la memoria en CD en la Secretaría y su posterior traspaso a la Biblioteca, ya que todos los envíos serían telemáticos.
- La consulta al proyecto por parte del tribunal y por parte de cualquier usuario, si el TFG fuera publicado en abierto, sería más eficaz.

Propuesta 3

Para esta solución, se propone implementar una aplicación web en lenguaje php, utilizando como Gestor de Bases de Datos MySQL.

El desarrollo de esta aplicación web, permitirá un flujo de trabajo (Workflow) entre el alumno, el director del TFG, el tribunal y la Secretaría.

El proceso completo de esta propuesta se detalla a continuación:

- El flujo de esta aplicación se inicia en la entidad alumno. Éste rellenará un formulario con los metadatos necesarios para la catalogación del TFG en el Repositorio Institucional de la Universidad de Huelva, Arias Montano, y al

mismo tiempo, subirá a un servidor su TFG. Tras enviar el formulario, el director del proyecto recibirá un correo indicando que tiene una tarea pendiente.

- Tras la intervención del alumno, el flujo de la aplicación se dirige al director, quien con la información proporcionada por el alumno, catalogará manualmente el proyecto en el Repositorio, entrando como usuario administrador en Dspace y dando permiso de lectura solo al tribunal.

El tribunal será informado por correo de la url donde podrá consultar el proyecto.

Tras la aceptación del director, el flujo de la aplicación se dirige a la Secretaría, quien quedará informada de la entrega para realizar el Acta.

- Tras la defensa y evaluación del proyecto, si el alumno permite su publicación en abierto, el director informará a la Biblioteca para que sea publicado.

En esta propuesta, quedaría por determinar qué servidor sería utilizado para ubicar la aplicación creada que permitiera el flujo del trabajo anteriormente mencionado. Una opción válida podría consistir en utilizar el servidor web de la Universidad y realizar la programación de esta aplicación en php y MySQL, con autenticación ante el LDAP de la Universidad, tanto por parte del alumno, como por parte del docente o el personal de la Biblioteca.

En este proceso intervendrían los directores de proyecto y el personal de la Biblioteca. Es necesario dejar claro a quien dirigirse ante cualquier problema

Se exponen a continuación los inconvenientes encontrados en el desarrollo de esta propuesta:

- Habría que determinar un servidor provisional, donde sería alojado el proyecto durante su entrega. En el momento que el proyecto sea subido por el director a Arias Montano, éste podría ser desalojado del servidor provisional.
- El director, entrando en Dspace como usuario administrador, subirá de forma manual el ítem al Repositorio, con los datos introducidos por el alumno en el formulario.

Se indican del mismo modo, las ventajas que podrían conseguirse si el modelo descrito se implantara:

- La Biblioteca ahorraría espacio físico de almacenamiento.
- Se evitaría la entrega del CD de la memoria en la Secretaría y su posterior traspaso a la Biblioteca, pues es subida por el alumno a un servidor dentro de su procedimiento telemático.
- La consulta al proyecto por parte del tribunal y por parte de cualquier usuario si el TFG fuera publicado en abierto, sería más eficaz.
- No habría gasto adicional de servidores ni de software pues el Repositorio ya está funcionando. Otro tema sería el servidor de almacenamiento provisional,

que no requeriría mucha capacidad y que podría utilizarse alguno existente en el SIC para no tener que adquirir uno nuevo.

- El director ya no tiene que presentarse en la Secretaría para presentar la "Propuesta de defensa de trabajo fin de grado".

Propuesta 4

Esta propuesta es similar a la descrita en el apartado anterior, pero esta vez los TFG serán almacenados directamente en el Repositorio institucional de la Universidad de Huelva Arias Montano.

Con esta propuesta, la aplicación permite un workflow entre alumno, director, tribunal, Secretaría y Biblioteca, en el entorno del servidor de aplicaciones Tomcat instalado en el servidor del Repositorio.

El proceso completo de esta propuesta se detalla a continuación:

- Al igual que en la propuesta anterior, el flujo de esta aplicación se inicia en la entidad alumno. Éste rellenará un formulario con los metadatos necesarios para la catalogación del TFG en el Repositorio. Como este formulario estaría alojado en el servidor del Repositorio, los archivos que constituyen el trabajo realizado por el alumno serían alojados directamente en el servidor Institucional del Repositorio.

Tras enviar los datos a través del formulario, el director asignado recibiría un correo indicando que tiene una tarea pendiente.

- El flujo de la aplicación se dirige en estos momentos hacia al director, quien, tras identificarse en la plataforma, puede comprobar los trabajos enviados por sus alumnos pendientes de subir al Repositorio. Tras seleccionar uno que contendrá toda la información proporcionada por el alumno, catalogará automáticamente el proyecto en el Repositorio dando permiso de lectura solo al tribunal. Éste será informado por correo de la url donde podrán consultar el proyecto. Tras la aceptación del director, el flujo de la aplicación se dirigiría a la Secretaría, quien quedará informada de la entrega para realizar el Acta.
- Tras ser catalogado el TFG en el repositorio, desaparecerá como tarea pendiente dentro del listado que le aparece al director tras identificarse.

Esta catalogación por parte del docente, sería totalmente automática, pues los datos proporcionados por el alumno y los archivos que constituyen el proyecto, ya están alojados en el servidor donde se procederá la catalogación.

- Tras la defensa y evaluación del proyecto, si el alumno permite su publicación, el director informará a la Biblioteca para que sea publicado en abierto.

Esta propuesta enfoca la programación en permitir acceso directo a la base de datos de Dspace (se hará referencia de Dspace en el apartado 3.2.), con la finalidad de hacer todo directamente sobre el Repositorio Institucional de la UHU.

En este proceso intervendrían los directores del trabajo y el personal de la Biblioteca. Es necesario dejar claro a quien dirigirse ante cualquier problema

Las ventajas de este desarrollo, son:

- La Biblioteca ahorraría espacio físico de almacenamiento.
- Se evitaría la entrega del CD de la memoria en la Secretaría y su posterior traspaso a la Biblioteca, ya que es subida por el alumno a un servidor dentro de su procedimiento telemático
- La consulta al proyecto por parte del tribunal y por parte de cualquier usuario si el TFG fuera publicado en abierto, sería más eficaz.
- No habría gasto adicional de servidores ni de software pues el Repositorio ya está funcionando.
- El director no tendría que cargar manualmente el ítem en el Repositorio, pues los metadatos y los ficheros subidos por el alumno ya estarían alojados en el servidor del Repositorio.
- El director ya no tiene que presentarse en la Secretaría para presentar la "Propuesta de defensa de trabajo fin de grado".

Propuesta 5

En esta última exposición, se propone la creación de un procedimiento en Administración Electrónica que desarrolle todo el flujo de trabajo explicado en las propuestas anteriores entre alumno, director del TFG, Secretaría y Biblioteca.

Un detalle de esta propuesta se expone a continuación:

- El alumno accedería al procedimiento de Administración Electrónica creado para este fin y rellenará los datos solicitados en el formulario, metadatos necesarios para la catalogación del TFG en el Repositorio. Tras enviar el formulario, el director del proyecto recibirá un correo indicando que tiene una tarea pendiente en un proceso de Administración Electrónica.
- El director accede al proceso pendiente, revisa y corrige los datos si es necesario. Posteriormente, realizará la carga automática de los metadatos y los ficheros que constituyen el TFG en el Repositorio implementándose una pasarela para dicho fin.
- Tras la catalogación en el Repositorio, el director accederá al ítem para dar permiso de lectura solo al tribunal, e informará a éste por correo de la url donde podrán consultar el proyecto. Tras la aceptación del director, el flujo del procedimiento se dirige a la Secretaría, quien quedará informada de la entrega para realizar el Acta.
- Tras la defensa y evaluación del proyecto, el director accederá al proyecto catalogado y lo publicará en abierto, si el alumno ha dado su consentimiento.

En este proceso intervendrían el director del trabajo, el personal del SIC que tramita la Administración Electrónica y el personal de la Biblioteca. Es necesario dejar claro a quien dirigirse ante cualquier problema

Se exponen a continuación los inconvenientes encontrados en el desarrollo de esta propuesta:

- El coste económico que supone la creación de un nuevo procedimiento en Administración Electrónica.
- Habría que determinar un servidor provisional que podría ser el mismo de Administración Electrónica, donde sería alojado el proyecto durante su entrega. En el momento que el proyecto sea subido por el director a Arias Montano, éste podría ser desalojado del servidor provisional.
- Es más complicado programar un interfaz de integración de información entre dos aplicaciones realizadas por entidades distintas como lo son Administración Electrónica y Repositorio Dspace.

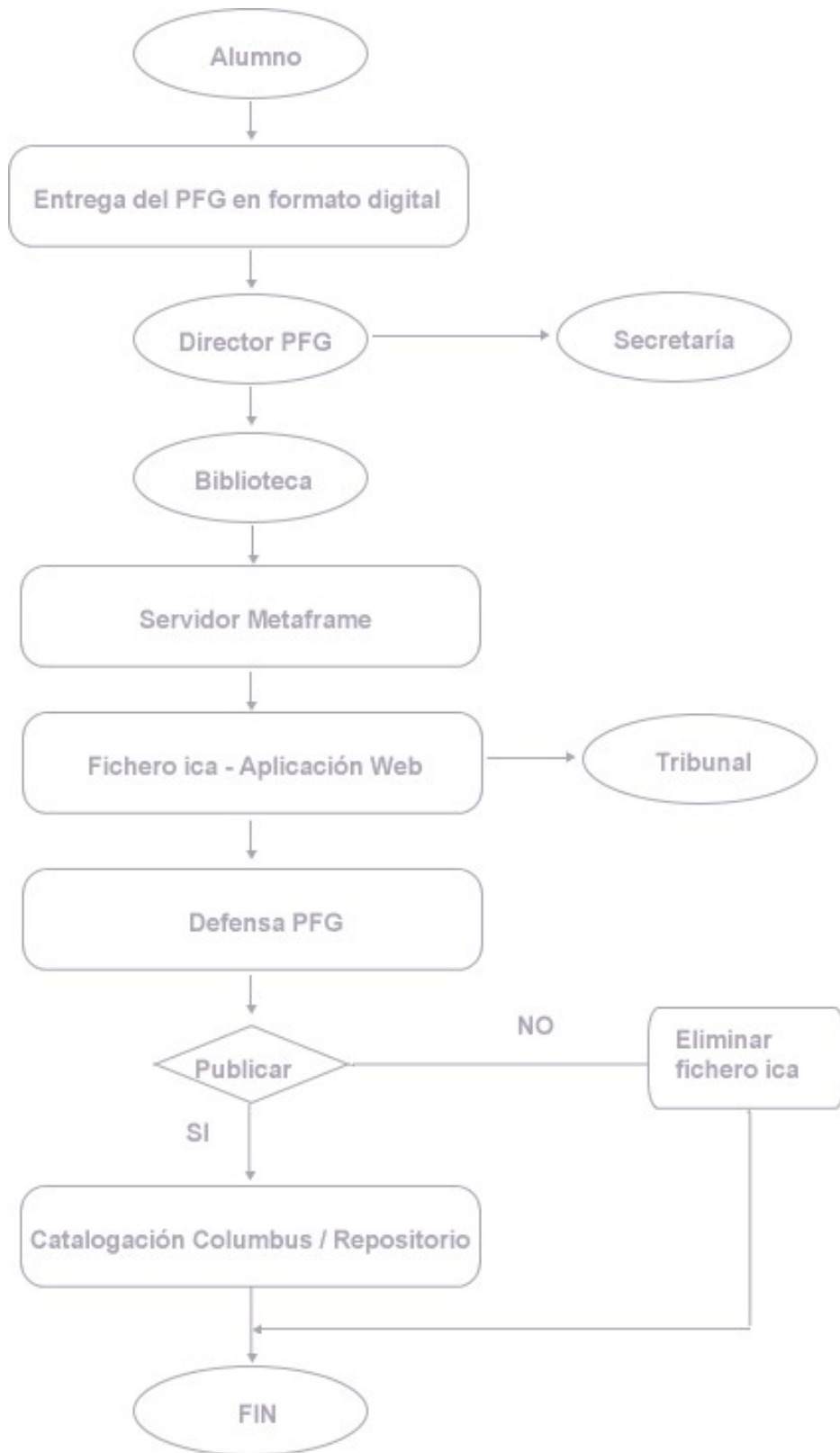
Se indican del mismo modo, las ventajas que podrían conseguirse si el modelo descrito se implantara:

- La Biblioteca ahorraría espacio físico de almacenamiento.
- Se evitaría la entrega del CD de la memoria en la Secretaría y su posterior traspaso a la Biblioteca, ya que ésta es subida por el alumno a un servidor dentro de su procedimiento telemático.
- La consulta al proyecto por parte del tribunal y por parte de cualquier usuario si el TFG fuera publicado en abierto, sería más eficaz.
- No habría gasto adicional de servidores ni de software pues el Repositorio ya está funcionando y el servidor de Administración Electrónica también.
- El director catalogaría de una forma fácil y eficaz el recurso, con la creación de un interfaz que permitiera la conexión entre la Administración Electrónica y el Repositorio.
- La Administración Electrónica ofrece un carácter más oficial a la entrega de TFG.
- El director ya no tiene que presentarse en la Secretaría para presentar la "Propuesta de defensa del trabajo fin de grado".

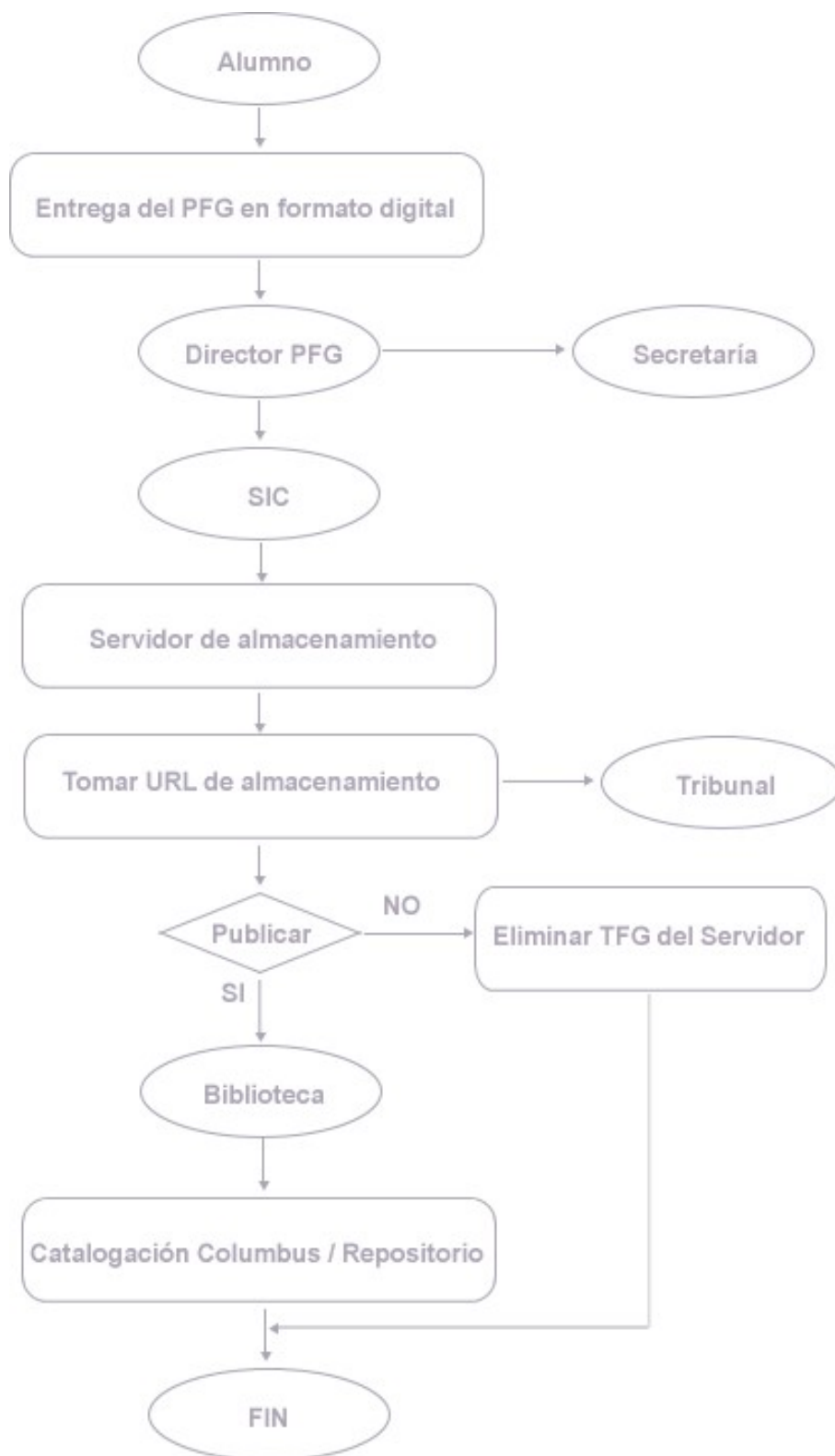
Las propuestas mostradas, han ido sufriendo distintas variaciones a medida que se han ido desarrollando y que se han ido debatiendo, intentando desarrollar siempre, la solución más óptima que beneficie al alumno y que minimice el trabajo de todo el personal involucrado en la presentación y defensa de un TFG.

2.2. Diagramas de flujo de las propuestas planteadas

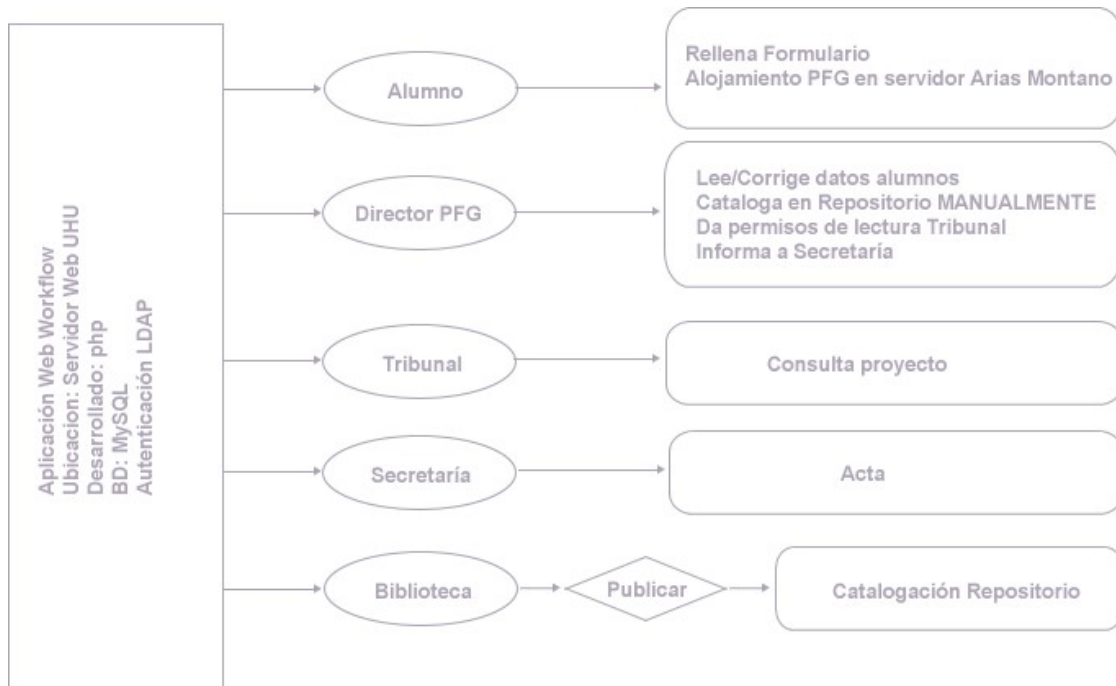
Propuesta 1



Propuesta 2



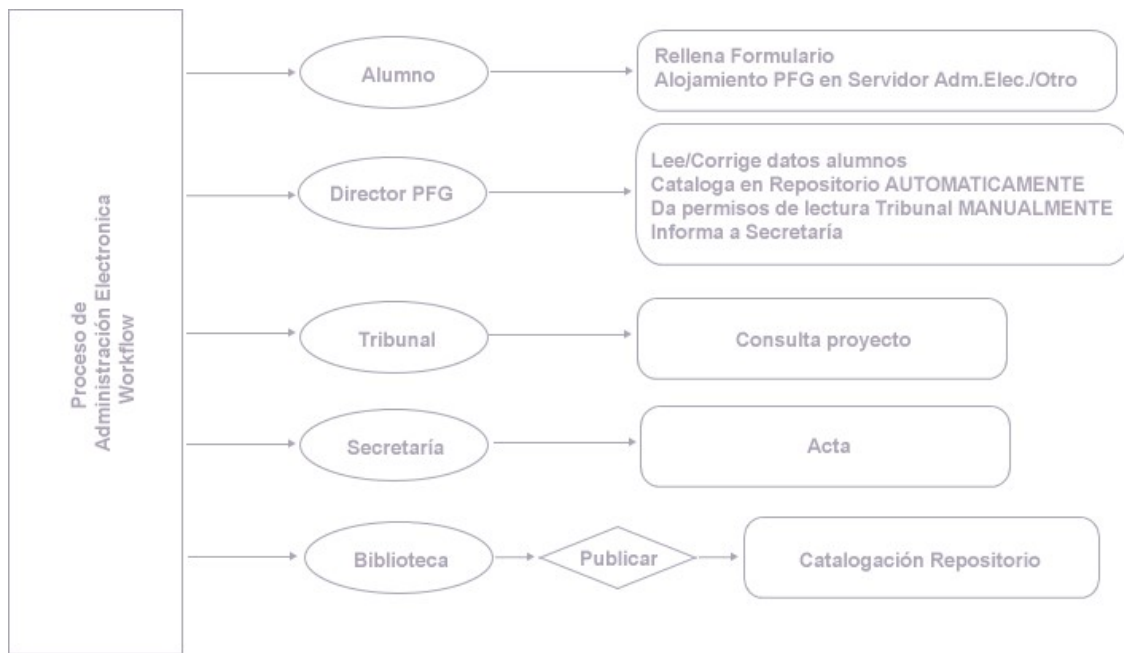
Propuesta 3



Propuesta 4



Propuesta 5



2.3. Justificación de la propuesta adoptada.

De todas las soluciones planteadas, se podría indicar que utilizar Administración Electrónica como medio de entrega de un TFG, podría aportar un carácter algo más oficial y relevante a todo proceso de entrega de documentos.

Esa solución, si bien no se ha llegado a desarrollar principalmente por el coste económico adicional que implicaría que una empresa externa a la Universidad elaborara un nuevo procedimiento, también debe indicarse el gran coste en horas trabajadas que implicaría ese desarrollo para la Universidad.

Un programador de la Universidad, si la empresa externa hubiera creado un proceso de Administración Electrónica, hubiera tenido que elaborar un software específico que permitiera la *Integración de la Información entre dos Aplicaciones* totalmente independientes y propietaria una de ellas de una empresa privada, y la otra open access (Dspace).

Debe mencionarse que “*integrar información entre aplicaciones*”, es uno de los trabajos más caros existentes en el mundo de la programación. Recordemos que las estadísticas indican que “*integrar información entre aplicaciones*”, es entre 5 y 20 veces más caro que elaborar un nuevo proyecto:

Las soluciones software, orientadas a la Enterprise Application Integration -EAI- y a la Enterprise Information Integration -EII-, suponen un gran reto tecnológico, sobre todo si las aplicaciones a integrar son aplicaciones web. Además, en la actualidad, se trata de una de las mayores preocupaciones para

las empresas, ya que informes expertos demuestran que los gastos de integración superan en una proporción de entre 5 y 20 al resto de desarrollos.

El motivo del alto coste viene dado por la inexistencia de métodos de trabajos, notaciones y conjunto de buenas prácticas que permitan obtener soluciones ingenieriles en vez de “soluciones artesanales”.

(<http://www.i2factory.com/servicios/integracion>)

Teniendo en cuenta la apreciación indicada en los párrafos anteriores sobre la propuesta número 5, se elige como solución óptima la propuesta 4.

La solución indicada en la propuesta 4, implica el desarrollo de una aplicación que genere un flujo de trabajo entre el alumno, director, tribunal, la Secretaría y la Biblioteca dentro del mismo entorno del repositorio. En el mismo servidor, será alojado directamente el TFG, metadatos y ficheros, y es ahí donde posteriormente será publicado para su consulta, si el alumno lo permite.

El coste económico de este proyecto sería nulo:

- en cuanto a hardware, servidor y sistema de copias de seguridad necesarias para almacenar los trabajos, se almacenarían directamente en el servidor del repositorio ya que actualmente, éste tiene capacidad suficiente para el alojamiento de los proyectos generados en el seno de la Universidad. En un futuro, habría que ir adquiriendo nuevos dispositivos de almacenamiento.
- respecto al software, el repositorio utiliza Dspace, software gratuito de libre acceso.

Las horas invertidas en programación han sido muchas debido al tiempo dedicado a investigar cómo desarrollar una aplicación web en el entorno de Dspace, que permita la integración de información, entre los datos proporcionados por el alumno y el tratamiento de estos para ser transformados en un formato aceptado por Dspace.

A pesar de lo laborioso y arduo que ha sido con frecuencia la elaboración de este proyecto, se puede indicar que ha sido perfectamente asumible por un programador que es dueño de la totalidad del entorno, pero en el caso de la Administración Electrónica, los entornos eran distintos.

Por último, el flujo de trabajo estaría controlado previa identificación del personal que accede a la aplicación web ante el LDAP de la Universidad, por lo que los datos de conexión estarían también a disposición del programador.

La solución adoptada ha requerido de un largo proceso de investigación para poder:

- Crear una aplicación web que genere el flujo del trabajo.
- Conectar esa aplicación con el LDAP para permitir la autenticación antes de entrar en el proceso.

- Conectar esa aplicación con la base de datos interna de Dspace para tomar los datos necesarios para catalogar el TFG.
- Crear tablas adicionales en Dspace, para el almacenamiento temporal de los metadatos proporcionados por el alumno antes de ser revisados y subidos al repositorio por el director.
- Integrar los datos proporcionados por el alumno en Dspace.
- Permitir que la aplicación ejecute un comando interno de Dspace que solo puede ejecutar el root desde un tty.

Todo este trabajo previo, realizado con la finalidad de conocer el entorno en el que debía desarrollarse la programación, ha requerido gran esfuerzo, dedicación e investigación hasta conseguir los resultados buscados.

Tras todo este análisis, finalmente se concluye que la solución 4 es la más adecuada por:

- Poseer un coste económico nulo.
- Por desarrollarse en un entorno propio de la UHU, sin tener la necesidad de acudir a una empresa externa.

2.4. Planificación del desarrollo del proyecto

Si se analiza con detenimiento este proyecto, podemos indicar en él varias fases, a saber:

- Fase I: Preparar Arias Montano para recepcionar TFG.
- Fase II: Crear una aplicación web que permita el flujo de trabajo entre las personas/entidades que intervienen en la entrega de un TFG.
- Fase III: Conectar la aplicación web al LDAP de la UHU.
- Fase IV: Conectar la aplicación web a la BD de Dspace.
- Fase V: Integrar la aplicación web en el entorno de Dspace.

A modo de resumen:

- **FASE I:** preparación del entorno de Arias Montano para recepcionar todos los TFG subidos a través de la aplicación web creada.

En Dspace todo se almacena en comunidades, que pueden ser divididas a su vez en sub-comunidades, las cuales reflejan la estructura lógica de la producción científica de la entidad a la que pertenece. Toda sub-comunidad contiene colecciones, que son agrupaciones con contenido relacionado.

Basándonos en esta clasificación, se ha creado dentro de la **Comunidad Aprendizaje y Docencia** las **subcomunidades** necesarias para subir los distintos proyectos que pueda presentar cualquier alumno de esta Universidad (TFG, TFC, TFM,...). Aunque esta memoria se enfoca en los TFG, debe tenerse

presente generalizar al máximo la aplicación, para poder contemplar en un futuro otros tipos de trabajos.

Se han creado, dentro de la subcomunidad *Trabajo Fin de Grado* todas las **colecciones** necesarias correspondientes a todas las titulaciones existentes en los estudios de Grado.

Se han creado los **grupos de usuarios que puedan administrar** cada colección. El director, codirector y tribunal, deben tener acceso al ítem subido para leerlo antes de su publicación; el director debe tener permiso para catalogar el ítem.

- **FASE 2:** se ha creado una aplicación web que permite el flujo de trabajo entre todas las personas y servicios que interactúan a lo largo de la entrega, defensa y calificación del TFG.

Puesto que Dspace está desarrollado en java, se ha utilizado este lenguaje para la realización del proyecto.

El interfaz de usuario de esta nueva aplicación web, está totalmente integrado en Dspace.

- **FASE 3:** conexión de la nueva aplicación con el LDAP de la Universidad. Dependiendo de quien se autentique, alumno ó director de proyecto, la aplicación web mostrará una u otra información.

Se aprovecha esta conexión, para extraer datos del LDAP, nombre, apellidos, y DNI de quien se autentifique. De esta forma, estos datos se rellenarán automáticamente en el formulario, facilitando la tarea a quien debe rellenarlo.

- **FASE 4:** conexión de la nueva aplicación con la base de datos interna de Dspace. Esto permite que el alumno elija el tipo de proyecto a subir, dentro de una lista desplegable que se mostrará con los datos extraídos tras consultar la/s tabla/s de Dspace que contiene dicha información. El elegir un valor de una lista y no teclearlo, además de facilitar la tarea a quien rellena el formulario, permite evitar errores mecanográficos.

Se aprovecha también las consultas para extraer los datos de la titulación, Director y Codirector si este existiera.

- **FASE 5:** permitir la conexión e interacción de la nueva aplicación web con el entorno Dspace. Eso permite subir directamente el TFG, metadatos y archivos al servidor del Repositorio.

A continuación, se analiza en detalle cada una de las fases mencionadas.

La **primera fase** consiste en **preparar el repositorio Arias Montano**, con la estructura necesaria para albergar todos los TFG (véase Figura 1):

- A nivel de Comunidades:

- dentro de la comunidad *Aprendizaje y Docencia* se ha creado en principio, la subcomunidad “Trabajos Fin de Grado”.

A medida que vayamos integrando otros tipos de proyecto en Arias Montano (Trabajos Fin de Carrera, Trabajos Fin de Master,...), se irán creando las subcomunidades destinadas a dicho fin.

- se ha creado, dentro de la subcomunidad anterior (Trabajos Fin de Grado), una colección por cada titulación existente dentro de los Grados impartidos en la UHU.

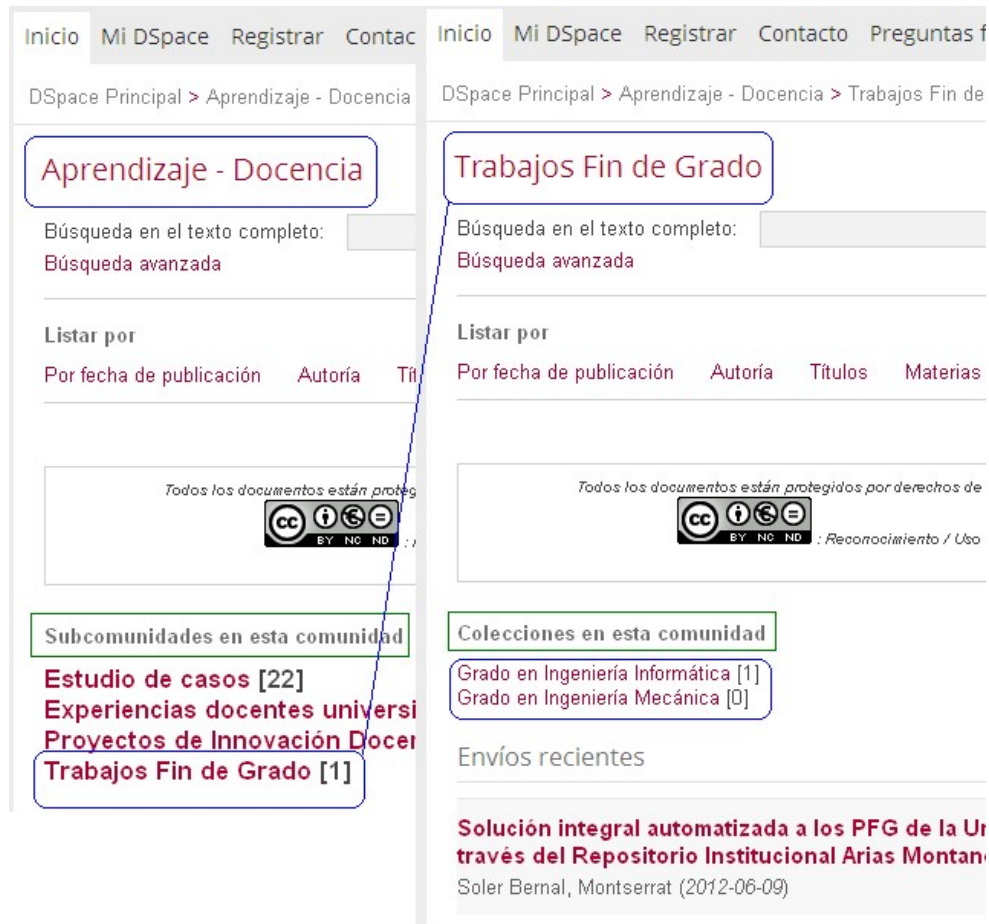


Figura 1

- A nivel de Formularios:
 - se ha creado un nuevo formulario, con los metadatos necesarios para catalogar los TFG.
 - se ha asociado este nuevo formulario, a cada colección creada dentro de la subcomunidad “Trabajos Fin de Grado”. Esto permite que cada proyecto catalogado en Arias Montano, tenga asociado los metadatos definidos en el formulario anterior.
- A nivel de Permisos:
 - por cada colección creada, Dspace ha creado automáticamente un grupo de administrador para esa colección.

La Biblioteca ha creado un grupo por colección con todos los profesores que puedan ser directores o tribunales de proyectos contenidos en esa colección.

A continuación se asocia este grupo, al grupo administrador de la colección creado automáticamente.

- con esta asociación se consigue que cualquier usuario perteneciente a ese grupo, previa identificación ante el LDAD, pueda subir un ítem o visualizarlo mientras esté oculto para su evaluación.
- todo director de proyecto debe enviar un correo a la Biblioteca con las personas del tribunal si no están dadas de alta en el grupo de administrador de la colección donde va a depositarse el ítem.

La **segunda fase** consiste en **crear una aplicación web** con la siguiente funcionalidad:

- presentar un *Formulario Electrónico* al alumno que le permita rellenar una serie de datos necesarios para la presentación del TFG y subir el/los archivo/s que forman parte de su proyecto.
- realizar el envío al director del proyecto para su posterior corrección y extracción de datos para ser catalogados en Arias Montano.
- ocultar el proyecto subido para que solo sea accesible por el grupo administrador de la colección.
- remitir correo a la Secretaría correspondiente, informando de la presentación del proyecto y comunicándole los datos necesarios para que genere el Acta de Calificación.
- remitir el proyecto entregado al tribunal para su lectura antes de la defensa.
- informar del proyecto entregado a la Biblioteca, para tras la defensa, si el alumno lo permite, indicar su estado público en Arias Montano.

Si el alumno permite su publicación en el Repositorio, debe remitir a la Biblioteca el impreso indicado en la aplicación, relleno y firmado por él.

La **tercera fase** consiste en interconectar la aplicación web creada con el LDAP de la Universidad, de tal forma que permite la autenticación de las personas que acceden a la aplicación utilizando la información almacenada en el servidor LDAP.

La **cuarta fase** consiste en interconectar la aplicación web creada con la base de datos de Dspace. De esta forma, tanto el alumno como el director del proyecto, a la hora de rellenar sus formularios correspondientes, toman de la base de datos de Dspace, los datos existentes que necesitan aportar.

El funcionamiento es sencillo. Determinados campos del formulario toman sus valores de las tablas correspondientes a la BD de Dspace. Estos valores se muestran en una lista desplegable desde la cual se elige el valor correspondiente. Esto garantiza la normalización de los valores y agiliza la introducción de datos.

La **quinta fase** consiste en interconectar la aplicación web con el sistema de Dspace. La finalidad es subir el ítem o proyecto con todos los metadatos asociados, incluidos los ficheros, para que el recurso quede catalogado. Para ello, se ha utilizado el comando interno import de Dspace.

Para utilizar este comando, los ficheros a importar deben tener un formato aceptado por Dspace y la ubicación de esa información debe estar en una estructura determinada, que se detallará en el apartado **3.7.1. Requerimientos**.

3. DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN ELEGIDA

3.1. Descripción del servidor del Repositorio

El servidor que alberga el Repositorio Institucional de la Universidad de Huelva, reside en la sala de máquinas de la Biblioteca Universitaria. Para tal fin, se adquirió un Servidor HP Proliant serie DL380 G5 (véase Figura 2):

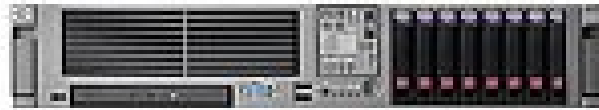


Figura 2

Las *características de este servidor* son:

- 2 procesadores Intel Xeon Quad Core a 2.0 GHz con 8 MB de caché L2 y FSB 1.333 MHz
- Ram: 8GB (4 módulos de 2Gb)
- 2 adaptadores Gigabit Ethernet en cobre
- 2 x discos duros internos hot plug de 72 GB a 10.000 rpm para sistema operativo
- 4 x discos duros internos de 146 GB a 10.000 rpm para datos
- Fuente de alimentación redundante
- DVD

Como *dispositivo de backup*, se adquiere:

- 1 x HP Storage Works 1U Rack Mount Kit
- 1 x Cinta interna HP StorageWorks Ultrium 920 SCSI

La configuración realizada sobre el servidor es la siguiente:

- Sobre los 4 d/d de 146Gb cada uno, (146x4=684Gb), se realiza un Raid5 (146x3=438Gb)
- Como Sistema Operativo se instala Red Hat Enterprise Linux Server, release 5.6 (Tikanga)
- Se crea el siguiente Sistema de Ficheros:

S.ficheros	Bloques de 1K	Usado	Dispon	Uso%	Montado en
/dev/cciss/c0d0p1	4061540	1832796	2019100	48%	/
tmpfs	1683824	0	1683824	0%	/dev/shm
/dev/cciss/c0d0p5	8123168	3630524	4073352	48%	/home
/dev/cciss/c0d0p2	10154020	4915348	4714552	52%	/usr

/dev/cciss/c0d0p3	4061572	1004768	2847156	27%	/var
/dev/cciss/c0d1p1	423276144	188619344	213155612	47%	/dspace

La instalación de Dspace versión 1.6.2 recomienda que el sistema gestor de bases de datos PostgreSQL sea instalado en un servidor distinto. Por tal motivo, se adquiere un *nuevo Servidor* HP Proliant serie DL380 G6 (véase Figura 3) con similares características al servidor anterior.

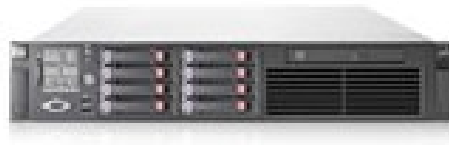


Figura 3

Del mismo modo, se procede a instalar un raid5 (146Gb x 3 = 438Gb) + 1 disco de hostspare.

3.2. Descripción del software Dspace

El Repositorio Institucional de la UHU, nace en el verano de 2007 tras instalar y configurar la versión 1.4.2 de Dspace, utilizando el interfaz JSPUI.

En agosto de 2010, se procede a migrar Dspace a su versión 1.6.2. Se aprovecha esta migración para cambiar al interfaz XMLUI, pues toda la documentación leída sobre la nueva versión apunta a mejoras consistentes en esta nueva interfaz. Se aprovecha esta migración para elaborar un nuevo diseño.

Por último, en el verano de 2011, se desarrolla un nuevo diseño más acorde con las nuevas funcionalidades. Toda la información relacionada con esta nueva versión, puede leerse en: http://www.dspace.org/1_6_2Documentation/DSpace-Manual.pdf

Antes de proseguir, dediquemos unas líneas a describir el software de Dspace así como sus requerimientos básicos.

¿Qué es Dspace?

DSpace es una plataforma de software de código abierto que permite a las instituciones:

- captar y describir el material digital existente en la institución, utilizando un módulo que, a través de un flujo de trabajo, permita la catalogación de los ítems con los metadatos requeridos para su definición dentro de Dspace.
- consultar y recuperar los activos digitales de la organización a través de Internet ofreciendo un interfaz web a los usuarios intuitivo y claro.
- preservar los recursos digitales a largo plazo.

Diagrama de modelo de datos

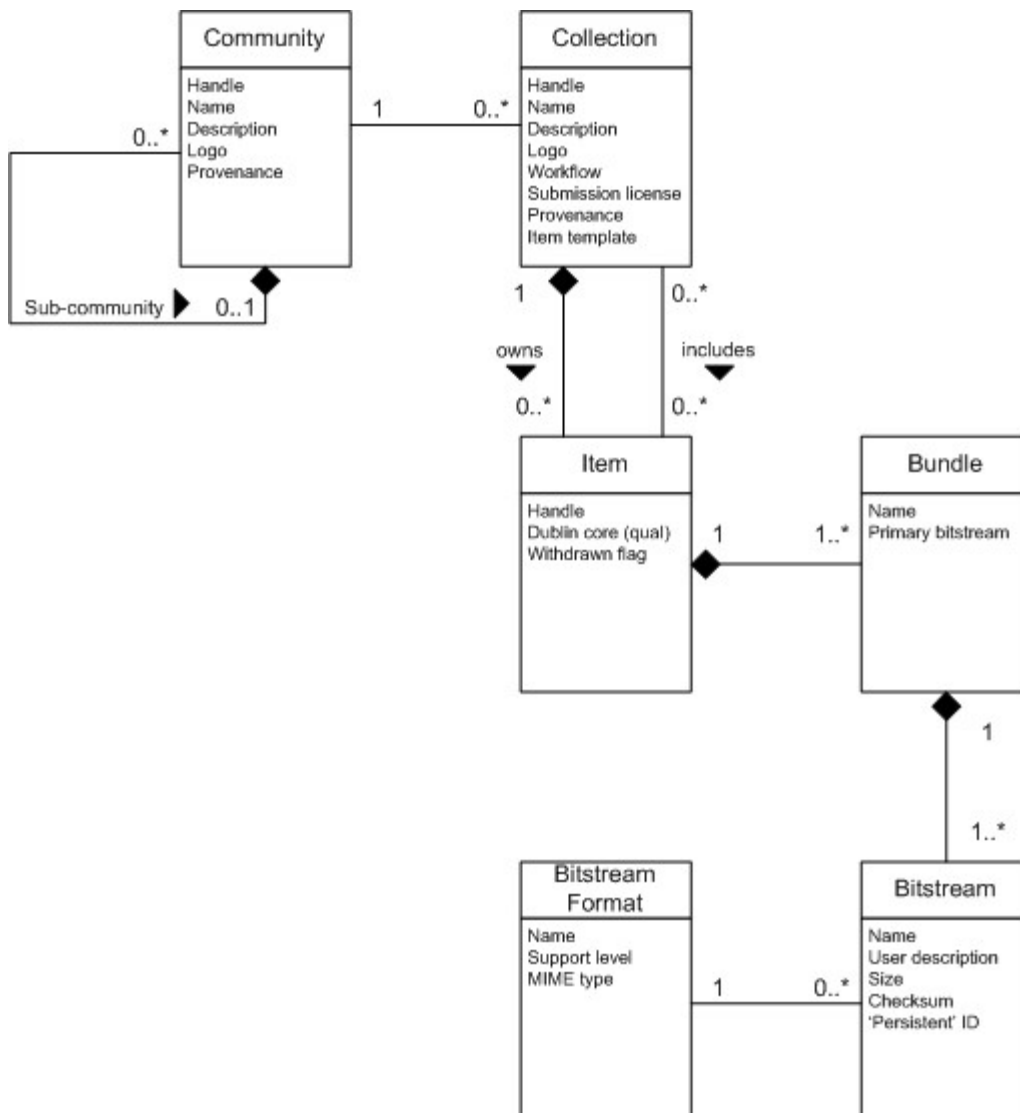


Figura 4

La forma de organizar los datos refleja la estructura de su organización interna (véase Figura 4), y esta organización se muestra perfectamente a los usuarios a través del interfaz de usuario. En Dspace todo se almacena en comunidades, que pueden ser divididas en sub-comunidades, las cuales reflejan la estructura lógica de la producción científica de la entidad a la que pertenece. Toda sub-comunidad contiene colecciones, que son agrupaciones con contenido relacionado.

Cada colección se compone de elementos básicos denominados ítems, que son los elementos catalogados en el repositorio. Cada ítem es propiedad de una colección, aunque tras ser creado y asignado a una colección, puede ser relacionado con más de una. Y por último, cada ítem está definido por una serie de metadatos, que lo identifican y caracterizan.

Prerrequisitos de software

A continuación se muestra una breve descripción del software base que debe existir en el servidor antes de la instalar Dspace:

- **Sun Java JDK 5 o superior:** En la actualidad, DSpace requiere Sun Java 5 o superior, debido al uso de nuevas capacidades del lenguaje introducidas en la versión 5 que presenta una codificación más fácil y clara. Java 5 o posterior puede descargarse desde: <http://java.sun.com/javase/downloads/index.jsp>
- **Apache Maven 2.0.8 o superior:** es necesario para armar el paquete de instalación de la instancia de DSpace. Maven puede descargarse desde: <http://maven.apache.org/download.html>
- **Apache Ant 1.7 o superior:** se utiliza una vez que el paquete de instalación se ha construido. Puede descargarse desde: <http://ant.apache.org>
- **Relational Database: PostgreSQL u Oracle:** se optó por utilizar como base de datos relacional PostgreSQL, al ser su licencia gratuita. PostgreSQL puede descargarse de : <http://www.postgresql.org/>
- **Tomcat 4.x o superior:** Dspace necesita funcionar con el mismo usuario que Tomcat, por lo que Tomcat debe ser instalado y ejecutado con el usuario llamado dspace. Puede ser bajado de <http://tomcat.apache.org/whichversion.html>

3.3. Preparar el entorno de Dspace para almacenar TFG

En Arias Montano se ha creado la estructura necesaria para permitir almacenar todos los TFG presentados en el seno de la UHU. Para ello, se configura:

- **Comunidades, subcomunidades y colecciones**

Todos los TFG serán albergados bajo la comunidad **Aprendizaje y Docencia**. En ella, se ha creado la subcomunidad **Trabajos Fin de Grado**. Dentro de esta subcomunidad, se ha creado una colección por cada **Titulación** existente en la oferta académica de Grados ofrecida por la Universidad. Esta estructura se tomó de la información publicada en la web de la Universidad:

<http://www.uhu.es/estudios/ofertaacademica/grados.html>

Ver [Anexo 1](#)

- **Formulario con los metadatos que definen los TFG**

Se ha creado un formulario específico para los TFG con los metadatos que necesitan identificar un ítem de este tipo de colección. Posteriormente, este formulario ha sido anexoado a todas las colecciones que utilizan ese formulario para la subida de sus ítems.

El formulario se define según [Anexo 2](#) en el archivo

/dspace/enovation/huelva-1173/dspace/config/input-forms.xml

Este tipo de formularios se crean manualmente accediendo al servidor como administrador y generando un fichero xml que contendrá los metadatos que indique el bibliotecario en el orden que indique y con el formato adecuado.

El siguiente paso es relacionar el formulario creado con las colecciones que utilizarán este formulario. Para ello, en el fichero de configuración `input-forms.xml`, se indica el nombre de la colección seguida del nombre de formulario que utilizará. Esta relación se indica en el [Anexo 3](#).

Por último, en `/dspace/enovation/huelva-1173/dspace/config/dspace.cfg` se definen los metadatos que deben visualizarse en pantalla, según se indica en [Anexo 4](#).

- **Grupos con permiso de Administrador de colecciones.**

Se ha creado un grupo por cada colección existente dentro de las subcomunidad *Trabajos Fin de Grado*. Cada grupo está constituido por todos los profesores que imparten clases en esa titulación, pues cualquiera puede ser director de proyecto o formar parte del tribunal. El director del proyecto, debe asegurarse que los docentes que formen parte del tribunal pertenezcan a este grupo, en caso contrario, debe comunicarlo a la Biblioteca para que sean añadidos.

Ese grupo tendrá permiso de Administrador sobre la colección, de tal forma que cualquier docente perteneciente al grupo podrá subir un ítem a la colección o revisarlo mientras que esté oculto.

Para esto, simplemente agregar al grupo de administradores de la colección *Trabajos Fin de Grado*, el grupo creado para esa colección.

La creación de los grupos y la asociación de estos al grupo administrador de la colección, se detalla en las figuras del [Anexo 5](#).

3.4. Creación de la aplicación web

Dentro del entorno de Arias Montano, se ha creado una aplicación web que permite la presentación del TFG, siguiendo un determinado flujo de trabajo en el que intervienen el alumno, el director del proyecto, la Secretaría del centro a la que pertenece el proyecto, el tribunal y la Biblioteca.

Esta aplicación se basa en el estándar de aplicaciones web JEE (Java Enterprise Edition), ya que Tomcat, entorno en el que se ha instalado Dspace, admite este estándar. Por tanto se desarrolla en este ámbito la aplicación web (`dspaceBuh`) que permitirá el flujo de trabajo indicado anteriormente.

En cuanto a modelado o estructura, este programa sigue el patrón de construcción de aplicaciones Modelo-Vista-Controlador, a partir de ahora MVC.

Según el siguiente enlace, http://es.wikipedia.org/wiki/Modelo_Vista_Controlador, podemos definir ese patrón del siguiente modo:

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software
--

que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres **componentes** distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones web, donde:

- **la vista** es la página HTML y el código que provee de datos dinámicos a la página,
- **el modelo** es el Sistema de Gestión de Base de Datos y la Lógica de negocio,
- y **el controlador** es el responsable de recibir los eventos de entrada desde la vista.

Según la definición indicada, podemos concluir que este proyecto describe:

- **la capa vista** con las páginas jsp
- **la capa controladora** con las clases java con sufijo Bean
- **la capa modelo o lógica de empresa** con la clase específica DspaceBuh.java

En cuanto a tecnología usada, este proyecto ha sido elaborado en jsp (Java Server Page), jstl (JavaSever Pages Standard Tag Library), scriptlets, struts y java.

- **JavaServer Pages (JSP)** es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Esta tecnología es un desarrollo de la compañía Sun Microsystems. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 2.1.

Las JSP's permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Bibliotecas de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

http://es.wikipedia.org/wiki/JavaServer_Pages

- La tecnología **JavaServer Pages Standard Tag Library (JSTL)** es un componente de Java EE. Extiende las ya conocidas JavaServer Pages (JSP) proporcionando cuatro bibliotecas de etiquetas (Tag Libraries) con utilidades ampliamente utilizadas en el desarrollo de páginas web dinámicas.

Estas bibliotecas de etiquetas extienden de la especificación de JSP (la cual a su vez extiende de la especificación de Servlet). Su API nos permite además desarrollar nuestras propias bibliotecas de etiquetas.

http://es.wikipedia.org/wiki/JavaServer_Pages_Standard_Tag_Library

- En JavaServer Pages (JSP), un **scriptlet** es un trozo de código Java incrustado en el código HTML.

El scriptlet es todo lo que está dentro de las etiquetas `<%%>`. Entre esas etiquetas, el usuario puede añadir cualquier Scriptlet válido, es decir, cualquier código JAVA válido.

<http://en.wikipedia.org/wiki/Scriptlet>

- **Struts.** El framework web Apache Struts es una solución de código abierto para la creación de aplicaciones web en Java. Las aplicaciones webs ofrecen no sólo páginas estáticas como los sitios webs convencionales, sino también pueden crear una respuesta dinámica. Una aplicación web puede interactuar con bases de datos y motores de lógica de negocios para personalizar la respuesta.

El framework Struts, está diseñado para ayudar a los desarrolladores a crear aplicaciones web que utilizan una arquitectura MVC.

El framework proporciona tres componentes claves:

- Un manejador de peticiones que proporcionado por el desarrollador de la aplicación, es mapeado a una URI estándar.
- Un manejador de respuestas que transfiere el control a otro recurso para que este complete la respuesta.
- Una biblioteca de etiquetas que ayuda a los desarrolladores a crear aplicaciones basadas en formularios interactivos con páginas de servidor.

http://es.wikipedia.org/wiki/Apache_Struts,
<http://osl2.uca.es/wikiIW/index.php/Struts>

- **Java** es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. La memoria es gestionada mediante un recolector de basura.

http://es.wikipedia.org/wiki/Java_%28lenguaje_de_programaci%C3%B3n%29

3.5. Integración de la aplicación web con LDAP

Una vez codificada la aplicación web que permite la interacción entre las distintas entidades que intervienen en la presentación del proyecto (alumno, director, Secretaría, tribunal y Biblioteca) se ha creado en su seno el código necesario para conectarla con el LDAP universitario. De esta forma, la entrada en el flujo del trabajo, será previa identificación, mostrándose una información u otra, dependiendo de quien solicita la entrada a la aplicación (al identificarse contra el LDAP, sabemos si es alumno o no).

La identificación ante el LDAP, no solo permite identificar si el que accede es alumno o no, también se aprovecha esta conexión para tomar datos personales del LDAP (dni,

nombre, apellido1, apellido2,...), que se asocian automáticamente a los campos del formulario donde debe recopilarse dicha información.

Cabe destacar que en el LDAP, no todos los datos tienen acceso de lectura pública. Existen datos privados que necesitamos obtener, tal es el caso del DNI. Esto complica la conexión teniéndose que realizar a través de un usuario con privilegios especiales que tiene permiso para leer datos privados.

La conexión al LDAP de la Universidad, se documenta en el [Anexo 6](#)

3.6. Integración de la aplicación web con la base de datos de Dspace

Los alumnos, en el momento que están registrando su proyecto en el formulario de entrega, deben indicar una serie de datos que serán extraídos de la base de datos de Dspace tras realizar la consulta oportuna.

Asimismo el director del proyecto, antes de subirlo al Repositorio, deberá indicar los miembros del tribunal ante los cuales se producirá la defensa del proyecto. Estos miembros serán seleccionados de una lista cuyos valores serán ofrecidos tras realizar la consulta a la base de datos de Dspace.

A continuación, se analizan los siguientes apartados:

- cómo realizar la conexión con la BD de Dspace
- saber qué tablas constituyen la BD de Dspace
- realizar las consultas necesarias, para extraer los datos que los usuarios (alumno y docente) deben rellenar en los formularios que se les presentan tras la identificación ante el LDAP.

Pasemos a analizar cada uno de estos tres puntos.

3.6.1. Conexión a la BD de Dspace

La conexión con la base de datos Dspace se realiza mediante JDBC, utilizando la librería postgresql-9.1-901.jdbc3.jar

En la tabla del [Anexo 7](#) se muestra cómo realizar la conexión a la BD de Dspace para poder realizar las consultas necesarias.

3.6.2. Esquema de la BD de Dspace

Uno de los puntos clave a investigar para realizar la subida automática del ítem y los metadatos asociados, así como para realizar posteriormente su retirada pública, es analizar el esquema de base de datos interna de Dspace. Las tablas que constituyen Dspace son:

SELECT * FROM pg_tables WHERE schemaname='public';

handle	checksum_results	harvested_collection
registrationdata	resourcepolicy	harvested_item
bundle2bitstream	subscription	community2collection
checksum_history	tasklistitem	collection
communities2item	workflowitem	community2community
epersongroup2eperson	eperson	community
epersongroup2workspaceitem	bitstream	collection2item
fileextension	bundle	bi_item
group2group	workspaceitem	bi_withdrawn
group2groupcache	bitstreamformatregistry	bi_2_dmap
item2bundle	collection_item_count	bi_2_dis
metadataschemaregistry	community_item_count	item
metadatafieldregistry	epersongroup	bi_4_dmap
most_recent_checksum	metadatatype	bi_4_dis

Se marca en azul, las tablas que van a consultarse en el desarrollo de este proyecto.

Vistas las tablas que forman parte de la base de datos de Dspace, empezamos a relacionarlas según necesidad de la consulta que vaya a realizarse para extraer los datos necesarios.

3.6.3. Extracción de datos (consultas), inserción y actualización

En el formulario que rellena el alumno, se presenta una serie de campos cuyos posibles valores se ofrecen en una lista desplegable (véase figura 5). Estos datos son extraídos de Dspace a través de una consulta.

Figura 5

Analicemos las consultas realizadas para la extracción de esos datos:

1.- Consulta: Tipo de proyecto a subir

Se mostrará un listado con todas las subcomunidades pertenecientes a la comunidad **Aprendizaje y Docencia** (el identificador de esta comunidad es el 3). Estos valores, están almacenados en la tabla **community**.

El motivo por el que se muestran todas las subcomunidades (véase figura 6) y no sola la subcomunidad **Trabajos Fin de Grado**, es intentar generalizar la aplicación a otro tipo de entregas de trabajos (Trabajos Fin de Carrera, Trabajos Fin de Master,...), para una futura ampliación de este programa.

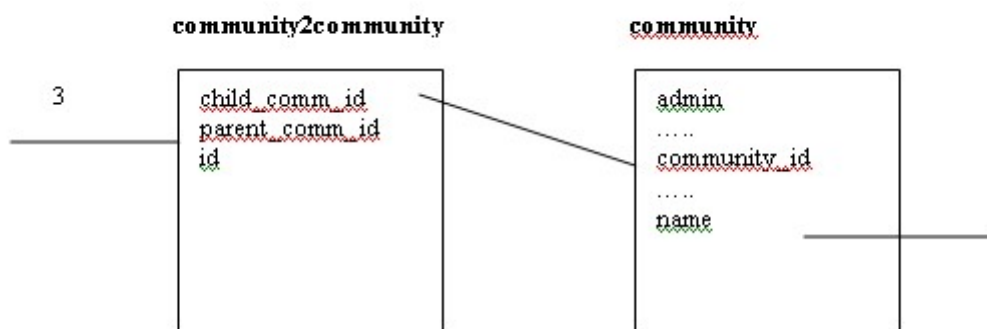
Figura 6

Para obtener ese listado, debe realizarse una búsqueda en dos tablas **community** y **community2community**. La relación entre ellas para conseguir los datos deseados, se muestra a continuación:

```

SELECT  community_id, name
FROM    community
INNER JOIN community2community ON  community.community_id =
community2community .child_comm_id
WHERE   parent_comm_id=3
    
```

En la cláusula **WHERE**, se añade el identificador de la comunidad **Aprendizaje y Docencia**.



La estructura de ambas tablas es:

<pre> SELECT * FROM information_schema.columns WHERE table_name='community'; </pre>	dspace	public	community	admin
	dspace	public	community	logo_bitstream_id
	dspace	public	community	community_id
	dspace	public	community	side_bar_text
	dspace	public	community	copyright_text
	dspace	public	community	introductory_text
	dspace	public	community	short_description
	dspace	public	community	name
<pre> SELECT * FROM information_schema.columns WHERE table_name=' community2community'; </pre>	dspace	public	community2community	child_comm_id
	dspace	public	community2community	parent_comm_id
	dspace	public	community2community	id

2.- Consulta: Titulación

Se mostrará un listado con todas las colecciones pertenecientes a la subcomunidad elegida en el apartado anterior. Estos valores, están almacenados en la tabla **collection**.

Tomemos como ejemplo el siguiente caso, tras elegir como subcomunidad **Trabajos Fin de Grado**, en la lista titulación (véase figura 7) se mostrarán todas las colecciones vinculadas a esa subcomunidad:

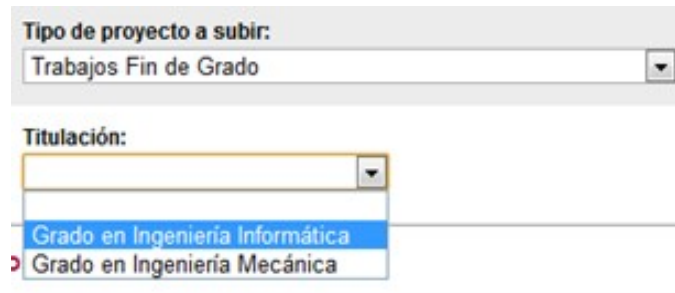


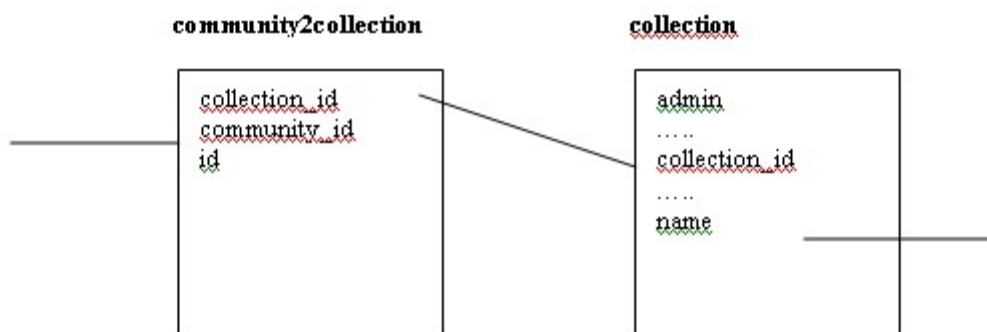
Figura 7

Para obtener ese listado, debe realizarse una búsqueda en dos tablas **collection** y **community2collection**. La relación entre ellas para conseguir los datos deseados, se muestra a continuación:

```

SELECT collection.collection_id, name
FROM collection
INNER JOIN community2collection ON collection.collection_id =
community2collection.collection_id
WHERE community_id = ?;
    
```

En la cláusula **WHERE**, se añade el identificador de la subcomunidad elegida.



La estructura de ambas tablas es:

<pre> SELECT * FROM information_schema.columns WHERE table_name='collection'; </pre>	dspace	public	collection	admin
	dspace	public	collection	submitter
	dspace	public	collection	workflow_step_3

	dspace	public	collection	workflow_step_2
	dspace	public	collection	workflow_step_1
	dspace	public	collection	template_item_id
	dspace	public	collection	logo_bitstream_id
	dspace	public	collection	collection_id
	dspace	public	collection	side_bar_text
	dspace	public	collection	copyright_text
	dspace	public	collection	license
	dspace	public	collection	provenance_description
	dspace	public	collection	introductory_text
	dspace	public	collection	short_description
	dspace	public	collection	name
SELECT * FROM information_schema.columns WHERE table_name='community2collection';	dspace	public	community2collection	collection_id
	dspace	public	community2collection	community_id
	dspace	public	community2collection	id

3.- Consulta: Director /Codirector

Se mostrará un listado con todos los docentes que pueden formar parte del tribunal de la colección elegida. Estos valores, están almacenados en la tabla **eperson**.

Tomemos como ejemplo el siguiente caso, tras elegir como subcomunidad **Trabajos Fin de Grado**, y como colección **Grado en Ingeniería Informática**, en la lista de director y codirector, se mostrarán todos los docentes (véase figura 8) vinculados a esa colección (como administradores de ella):

Alumno: Entrega Trabajo Fin de Grado

Datos personales:

DNI: Nombre:

Primer Apellido: Segundo Apellido:

Tipo de proyecto a subir:
 84

Titulación:
 360

Datos del director, codirector:

Email Director:

Nombre:

Nombre:

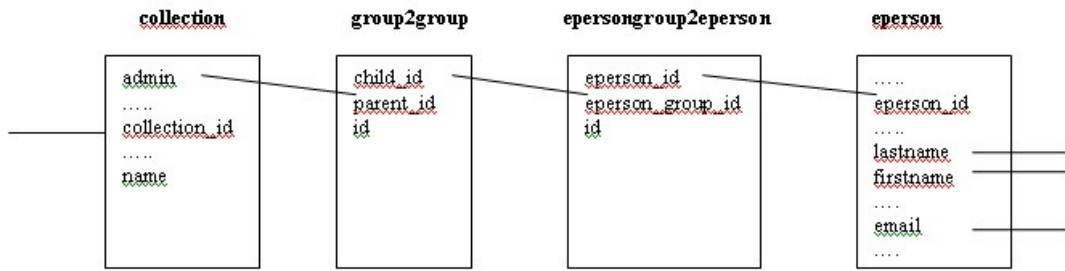
Figura 8

Para obtener ese listado, debe realizarse una búsqueda en cuatro tablas, **collection**, **group2group**, **epersongroup2eperson** y **eperson**. La relación entre ellas para conseguir los datos deseados, se muestra a continuación:

```

SELECT eperson.eperson_id, eperson.firstname, eperson.lastname, eperson.email
FROM ((collection INNER JOIN group2group
ON collection.admin = group2group.parent_id)
INNER JOIN epersongroup2eperson
ON group2group.child_id = epersongroup2eperson.eperson_group_id)
INNER JOIN eperson
ON epersongroup2eperson.eperson_id = eperson.eperson_id
WHERE collection_id = ?;
    
```

En la cláusula **WHERE**, se añade el identificador de la colección elegida.



La estructura de las tablas es:

<pre> SELECT * FROM information_schema.columns WHERE table_name='collection'; </pre>	<table border="1"> <tr><td>dspace</td><td>public</td><td>collection</td><td>admin</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>submitter</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>workflow_step_3</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>workflow_step_2</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>workflow_step_1</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>template_item_id</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>logo_bitstream_id</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>collection_id</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>side_bar_text</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>copyright_text</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>license</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>provenance_description</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>introductory_text</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>short_description</td></tr> <tr><td>dspace</td><td>public</td><td>collection</td><td>name</td></tr> </table>	dspace	public	collection	admin	dspace	public	collection	submitter	dspace	public	collection	workflow_step_3	dspace	public	collection	workflow_step_2	dspace	public	collection	workflow_step_1	dspace	public	collection	template_item_id	dspace	public	collection	logo_bitstream_id	dspace	public	collection	collection_id	dspace	public	collection	side_bar_text	dspace	public	collection	copyright_text	dspace	public	collection	license	dspace	public	collection	provenance_description	dspace	public	collection	introductory_text	dspace	public	collection	short_description	dspace	public	collection	name
dspace	public	collection	admin																																																										
dspace	public	collection	submitter																																																										
dspace	public	collection	workflow_step_3																																																										
dspace	public	collection	workflow_step_2																																																										
dspace	public	collection	workflow_step_1																																																										
dspace	public	collection	template_item_id																																																										
dspace	public	collection	logo_bitstream_id																																																										
dspace	public	collection	collection_id																																																										
dspace	public	collection	side_bar_text																																																										
dspace	public	collection	copyright_text																																																										
dspace	public	collection	license																																																										
dspace	public	collection	provenance_description																																																										
dspace	public	collection	introductory_text																																																										
dspace	public	collection	short_description																																																										
dspace	public	collection	name																																																										
<pre> SELECT * FROM information_schema.columns WHERE table_name=' group2group'; </pre>	<table border="1"> <tr><td>dspace</td><td>public</td><td>group2group</td><td>child_id</td></tr> <tr><td>dspace</td><td>public</td><td>group2group</td><td>eperson_group_id</td></tr> <tr><td>dspace</td><td>public</td><td>group2group</td><td>id</td></tr> </table>	dspace	public	group2group	child_id	dspace	public	group2group	eperson_group_id	dspace	public	group2group	id																																																
dspace	public	group2group	child_id																																																										
dspace	public	group2group	eperson_group_id																																																										
dspace	public	group2group	id																																																										
<pre> SELECT * FROM </pre>	<table border="1"> <tr><td>dspace</td><td>public</td><td>epersongroup2eperson</td><td>eperson_id</td></tr> <tr><td>dspace</td><td>public</td><td>epersongroup2eperson</td><td>parent_id</td></tr> </table>	dspace	public	epersongroup2eperson	eperson_id	dspace	public	epersongroup2eperson	parent_id																																																				
dspace	public	epersongroup2eperson	eperson_id																																																										
dspace	public	epersongroup2eperson	parent_id																																																										

<pre>information_schema.columns WHERE table_name=' epersongroup2eperson';</pre>	<table border="1"> <tr> <td>dspace</td> <td>public</td> <td>epersongroup2eperson</td> <td>id</td> </tr> </table>	dspace	public	epersongroup2eperson	id																																																
dspace	public	epersongroup2eperson	id																																																		
<pre>SELECT * FROM information_schema.columns WHERE table_name=' eperson';</pre>	<table border="1"> <tr><td>dspace</td><td>public</td><td>eperson</td><td>self_registered</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>require_certificate</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>can_log_in</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>sub_frequency</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>eperson_id</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>language</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>netid</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>phone</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>lastname</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>firstname</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>password</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>email</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>last_active</td></tr> </table>	dspace	public	eperson	self_registered	dspace	public	eperson	require_certificate	dspace	public	eperson	can_log_in	dspace	public	eperson	sub_frequency	dspace	public	eperson	eperson_id	dspace	public	eperson	language	dspace	public	eperson	netid	dspace	public	eperson	phone	dspace	public	eperson	lastname	dspace	public	eperson	firstname	dspace	public	eperson	password	dspace	public	eperson	email	dspace	public	eperson	last_active
dspace	public	eperson	self_registered																																																		
dspace	public	eperson	require_certificate																																																		
dspace	public	eperson	can_log_in																																																		
dspace	public	eperson	sub_frequency																																																		
dspace	public	eperson	eperson_id																																																		
dspace	public	eperson	language																																																		
dspace	public	eperson	netid																																																		
dspace	public	eperson	phone																																																		
dspace	public	eperson	lastname																																																		
dspace	public	eperson	firstname																																																		
dspace	public	eperson	password																																																		
dspace	public	eperson	email																																																		
dspace	public	eperson	last_active																																																		

4.- Consulta: Listado de proyectos correspondientes a un director, pendientes de subir al Repositorio

El formulario que abre el director tras identificarse (véase figura 9), muestra todos los proyectos que tiene a su nombre (presentado por sus alumnos), pendientes de subir al Repositorio:



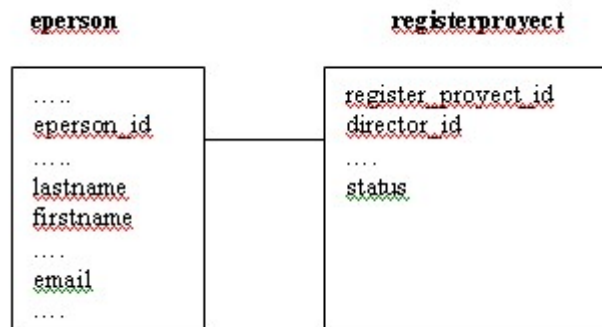
Figura 9

Para este fin, se realiza la siguiente consulta:

```

SELECT register_proyect_id,
dni_alumno, nombre_alumno, apellido1_alumno, apellido2_alumno,
community_id, collection_id,
director_id, codirector_id,
miembro1_tribunal_id, miembro2_tribunal_id, miembro3_tribunal_id,
titulo, palabras_claves_spi, palabras_claves_eng, resumenSpi, resumenEng,
fechaLectura, idioma, fichero1, fichero2, accesoAbierto, fecha_hora_entrega, status,
firstname AS nombreDirector, lastname AS apellidosDirector
FROM registerproyect
INNER JOIN eperson ON registerproyect.director_id = eperson.eperson_id
WHERE status = ? AND eperson.email = ?;
    
```

En la cláusula **WHERE**, se añade el email de docente, y el estado 0 (proyectos no subidos al repositorio).



La estructura de las tablas es:

<p>SELECT * FROM information_schema.columns WHERE table_name='registerproyect';</p>	<table border="1"> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>status</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>miembro3_tribunal_id</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>miembro2_tribunal_id</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>miembro1_tribunal_id</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>codirector_id</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>director_id</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>collection_id</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>community_id</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>register_proyect_id</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>accesoabierto</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>idioma</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>apellido2_alumno</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>apellido1_alumno</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>nombre_alumno</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>dni_alumno</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>fichero2</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>fichero1</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>resumeneng</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>resumenspi</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>palabras_claves_eng</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>palabras_claves_spi</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>titulo</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>fechalectura</td></tr> <tr><td>dspace</td><td>public</td><td>registerproyect</td><td>fecha_hora_entrega</td></tr> </table>	dspace	public	registerproyect	status	dspace	public	registerproyect	miembro3_tribunal_id	dspace	public	registerproyect	miembro2_tribunal_id	dspace	public	registerproyect	miembro1_tribunal_id	dspace	public	registerproyect	codirector_id	dspace	public	registerproyect	director_id	dspace	public	registerproyect	collection_id	dspace	public	registerproyect	community_id	dspace	public	registerproyect	register_proyect_id	dspace	public	registerproyect	accesoabierto	dspace	public	registerproyect	idioma	dspace	public	registerproyect	apellido2_alumno	dspace	public	registerproyect	apellido1_alumno	dspace	public	registerproyect	nombre_alumno	dspace	public	registerproyect	dni_alumno	dspace	public	registerproyect	fichero2	dspace	public	registerproyect	fichero1	dspace	public	registerproyect	resumeneng	dspace	public	registerproyect	resumenspi	dspace	public	registerproyect	palabras_claves_eng	dspace	public	registerproyect	palabras_claves_spi	dspace	public	registerproyect	titulo	dspace	public	registerproyect	fechalectura	dspace	public	registerproyect	fecha_hora_entrega
dspace	public	registerproyect	status																																																																																														
dspace	public	registerproyect	miembro3_tribunal_id																																																																																														
dspace	public	registerproyect	miembro2_tribunal_id																																																																																														
dspace	public	registerproyect	miembro1_tribunal_id																																																																																														
dspace	public	registerproyect	codirector_id																																																																																														
dspace	public	registerproyect	director_id																																																																																														
dspace	public	registerproyect	collection_id																																																																																														
dspace	public	registerproyect	community_id																																																																																														
dspace	public	registerproyect	register_proyect_id																																																																																														
dspace	public	registerproyect	accesoabierto																																																																																														
dspace	public	registerproyect	idioma																																																																																														
dspace	public	registerproyect	apellido2_alumno																																																																																														
dspace	public	registerproyect	apellido1_alumno																																																																																														
dspace	public	registerproyect	nombre_alumno																																																																																														
dspace	public	registerproyect	dni_alumno																																																																																														
dspace	public	registerproyect	fichero2																																																																																														
dspace	public	registerproyect	fichero1																																																																																														
dspace	public	registerproyect	resumeneng																																																																																														
dspace	public	registerproyect	resumenspi																																																																																														
dspace	public	registerproyect	palabras_claves_eng																																																																																														
dspace	public	registerproyect	palabras_claves_spi																																																																																														
dspace	public	registerproyect	titulo																																																																																														
dspace	public	registerproyect	fechalectura																																																																																														
dspace	public	registerproyect	fecha_hora_entrega																																																																																														
<p>SELECT * FROM information_schema.columns WHERE table_name='eperson';</p>	<table border="1"> <tr><td>dspace</td><td>public</td><td>eperson</td><td>self_registered</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>require_certificate</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>can_log_in</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>sub_frequency</td></tr> <tr><td>dspace</td><td>public</td><td>eperson</td><td>eperson_id</td></tr> </table>	dspace	public	eperson	self_registered	dspace	public	eperson	require_certificate	dspace	public	eperson	can_log_in	dspace	public	eperson	sub_frequency	dspace	public	eperson	eperson_id																																																																												
dspace	public	eperson	self_registered																																																																																														
dspace	public	eperson	require_certificate																																																																																														
dspace	public	eperson	can_log_in																																																																																														
dspace	public	eperson	sub_frequency																																																																																														
dspace	public	eperson	eperson_id																																																																																														

dspace	public	eperson	language
dspace	public	eperson	netid
dspace	public	eperson	phone
dspace	public	eperson	lastname
dspace	public	eperson	firstname
dspace	public	eperson	password
dspace	public	eperson	email
dspace	public	eperson	last_active

5.- Insertar: Datos confirmados por el alumno en la tabla ‘registerproyect’

La siguiente instrucción, graba los datos introducidos y confirmados por el alumno en la tabla creada a tal fin.

```

INSERT INTO registerproyect
(register_proyect_id,
dni_alumno, nombre_alumno, apellido1_alumno, apellido2_alumno,
community_id, collection_id,
director_id, codirector_id,
miembro1_tribunal_id, miembro2_tribunal_id, miembro3_tribunal_id,
titulo, palabras_claves_spi, palabras_claves_eng, resumenSpi, resumenEng,
fechaLectura, idioma, fichero1, fichero2, accesoAbierto, fecha_hora_entrega, status)
VALUES (
(SELECT COALESCE(MAX(register_proyect_id)+1, 1)
FROM registerproyect),
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
    
```

6.- Actualizar: Datos confirmados por el director en la tabla ‘registerproyect’

La siguiente instrucción, actualiza el proyecto cuyo identificador se le pasa como parámetro (register_proyect_id = ?), con los datos corregidos por el director del proyecto.

```

UPDATE registerproyect
SET community_id = ?, collection_id = ?,
miembro1_tribunal_id = ?, miembro2_tribunal_id = ?, miembro3_tribunal_id = ?,
    
```

```

titulo = ?, palabras_claves_spi = ?, palabras_claves_eng = ?, resumenSpi = ?,
resumenEng = ?, fechaLectura = ?, idioma = ?, fichero1 = ?, fichero2 = ?, accesoAbierto
= ?, status = ? " +
WHERE register_project_id = ?;
    
```

7.- Retirar el ítem subido al repositorio

Tras subir el ítem al repositorio, este debe retirarse (ocultarse) para que no sea visible para todos y solo sea accesible por el grupo administrador.

Las instrucciones que se ejecutan para realizar este proceso son:

```

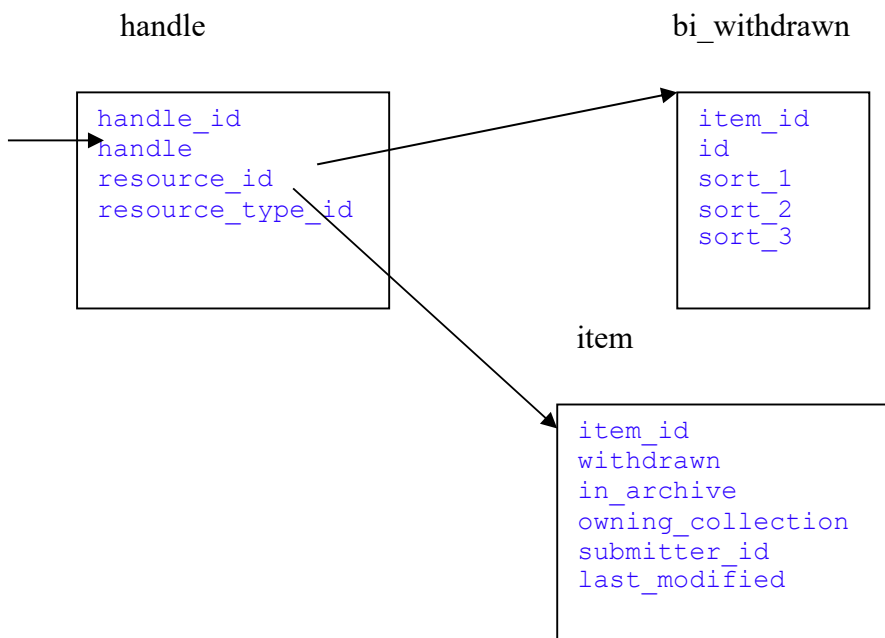
SELECT resource_id
FROM handle
WHERE handle_id = ?;
    
```

```

UPDATE item
SET withdrawn = 't', last_modified = current_timestamp
WHERE item_id = ?;
    
```

```

INSERT INTO bi_withdrawn (id, item_id, sort_1, sort_2, sort_3)
VALUES (nextval('bi_withdrawn_seq'), ?, ?, current_date, current_date);
    
```



La estructura de las tablas es:

SELECT *	<code>item_id</code>	– integer
	<code>id</code>	– integer

<p>FROM information_schema.columns</p> <p>WHERE table_name= 'bi_withdrawn';</p>	<p>sort_1 - text sort_2 - text sort_3 - text</p>
<p>SELECT id, item_id, sort_1, sort_2, sort_3</p> <p>FROM bi_withdrawn;</p>	<p>3 1569 titulo1 2009-09-18T10:37:13Z 2009-09-18T10:37:13Z 4 1568 titulo2 2009-09-18T10:32:02Z 2009-09-18T10:32:02Z 5 2433 titulo3 2010-01-13T09:23:54Z 2010-01-13T09:23:54Z 6 2809 titulo4 2010-02-17T13:30:27Z 2010-02-17T13:30:27Z 28 5620 prueba 2012-04-25 2012-04-25T07:47:49Z 31 5715 titulo5 2012-06-02 2012-06-02T07:03:39Z 32 5719 soluci 2012-06-03 2012-06-03T20:59:06Z 34 5717 soluci 2012-06-03 2012-06-03T09:39:22Z 40 5718 soluci 2012-06-03 2012-06-03T19:58:06Z 41 5720 titulo 2012-06-05 2012-06-05T07:14:59Z</p>
<p>SELECT *</p> <p>FROM information_schema.columns</p> <p>WHERE table_name='handle';</p>	<p>resource_id resource_type_id handle_id handle</p>
<p>SELECT handle_id, handle, resource_id, resource_type_id</p> <p>FROM handle;</p>	<p>..... 5690 10272/5684 5715 2 5692 10272/5683 5717 2 5693 10272/5693 5718 2 5694 10272/5694 5719 2 5695 10272/5695 5720 2</p>
<p>SELECT *</p> <p>FROM information_schema.columns</p> <p>WHERE table_name='item';</p>	<p>withdrawn in_archive owning_collection submitter_id item_id last_modified</p>
<p>SELECT item_id, withdrawn,</p>	<p>5715 t f 360 1</p>

in_archive, owning_collection, submitter_id, last_modified FROM item;	5719	t	f	2012-06-03 20:41:39.511+02
				360 1
	5717	t	f	2012-06-03 23:46:53.404+02
				360 1
	5718	t	t	2012-06-05 09:33:27.914+02
				360 1
	5720	t	f	2012-06-06 18:35:43.36388+02
				360 1
				2012-06-08 13:30:11.044+02

3.6.4. Crear una tabla en Dspace, para almacenar TFG

Una vez que el alumno ha introducido y validado el formulario que debe rellenar para presentar su trabajo, estos datos no son grabados en el Repositorio hasta ser revisados por el director del TFG.

Para su almacenamiento temporal, se crea una tabla, donde serán almacenados todos los datos tras la validación del alumno y de donde serán recuperados para ser mostrados al director que los subirá posteriormente al Repositorio. La estructura de la tabla es la siguiente:

```

CREATE TABLE registerproyect (
    register_proyect_id    INTEGER           NOT NULL,
    dni_alumno            CHAR(25)           NOT NULL,
    nombre_alumno        CHAR(50)           NULL,
    apellido1_alumno     CHAR(50)           NULL,
    apellido2_alumno     CHAR(50)           NULL,
    community_id         INTEGER           NOT NULL,
    collection_id        INTEGER           NOT NULL,
    director_id          INTEGER           NOT NULL,
    codirector_id        INTEGER           NULL,
    miembro1_tribunal_id INTEGER           NULL,
    miembro2_tribunal_id INTEGER           NULL,
    miembro3_tribunal_id INTEGER           NULL,
    titulo               VARCHAR(100)       NOT NULL,
    palabras_claves_spi  VARCHAR(250)       NOT NULL,
    palabras_claves_eng  VARCHAR(250)       NOT NULL,
    resumenSpi           VARCHAR(1000)      NOT NULL,
    resumenEng           VARCHAR(1000)      NOT NULL,
    fechaLectura        DATE               NOT NULL,
    idioma               CHAR(3)           NOT NULL,

```

fichero1	VARCHAR(100)	NOT NULL,
fichero2	VARCHAR(100)	NULL,
accesoAbierto	BOOLEAN	NOT NULL,
fecha_hora_entrega	TIMESTAMP	NOT NULL,
status	BOOLEAN	NOT NULL,
PRIMARY KEY (register_proyect_id)		

3.7. Integración de la aplicación web con Dspace

El docente deberá subir al Repositorio el TFG entregado. Todos los metadatos necesarios y el archivo que contiene la memoria, han sido indicados en el formulario por el alumno. Tras la revisión de estos datos por el director, se genera automáticamente un archivo xml con todos los metadatos mencionados.

Este archivo xml junto a la memoria, serán importados a la colección indicada por el alumno en el Repositorio Institucional.

Para realizar la importación, se hace uso del comando import que posee Dspace, y se generará la estructura necesaria para almacenar el archivo xml y los archivos subidos por el alumno para que dicho comando realice su función.

A continuación se explica este comando, su funcionalidad y sus parámetros, así como la estructura necesaria que hay que crear para que funcione.

3.7.1. Requerimientos

Antes de proceder a la importación del item, debe existir una estructura determinada, donde son almacenados el fichero xml y los ficheros subidos por el alumno.

Para dicho fin se crea, en el momento que el alumno sube los archivos de su memoria, en el directorio /dspace/FichImportar, un directorio cuyo nombre es el identificador del proyecto, seguido del directorio item.

En ese directorio quedan almacenados:

- Los ficheros subidos por el alumno.
- El fichero xml (denominado **dublin_core.xml**) generado automáticamente. Este fichero contendrá los metadatos con formato Dublín Core. Dublin Core es el modelo estándar de metadatos utilizado por la mayoría de las Bibliotecas (http://es.wikipedia.org/wiki/Dublin_Core).
- Adicionalmente se creará un fichero denominado **contents** que contendrá una línea por fichero que sube el alumno, con el nombre del fichero subido.

Resumiendo, la estructura de directorio sería:

```
/dspace/FichImportar/1/item
    dublin_core.xml
    contents
    fichero1.pdf
    fichero2.zip
```

y el contenido de los ficheros dublin_core.xml y contents será:

dublin_core.xml	contents
<pre><dublin_core> <dcvalue element="contributor" qualifier="advisor">ApellidosDir, NombreDir</dcvalue> <dcvalue element="contributor" qualifier="author">ApellAutor, NombreAutor</dcvalue> <dcvalue element="contributor" qualifier="other">UNIVERSIDAD DE HUELVA.Facultad-Escuela.Titulacion</dcvalue> <dcvalue element="date" qualifier="created">aaaa-mm-dd</dcvalue> <dcvalue element="description" qualifier="abstract">Resumen- spa</dcvalue> <dcvalue element="description" qualifier="abstract">Resumen- eng</dcvalue> <dcvalue element="language" qualifier="iso">idioma</dcvalue> <dcvalue element="subject" qualifier="other">PClave1-spa</dcvalue> <dcvalue element="subject" qualifier="other">PClave2-spa</dcvalue> <dcvalue element="subject" qualifier="other">PClave3-spa</dcvalue> <dcvalue element="subject" qualifier="other">PClave4-spa</dcvalue> <dcvalue element="subject" qualifier="other">PClave5-spa</dcvalue> <dcvalue element="subject" qualifier="other">PClave1-eng</dcvalue> <dcvalue element="subject" qualifier="other">PClave2- eng </dcvalue> <dcvalue element="subject" qualifier="other">PClave3- eng </dcvalue> <dcvalue element="subject" qualifier="other">PClave4- eng </dcvalue> <dcvalue element="subject" qualifier="other">PClave5- eng </dcvalue> <dcvalue element="title" qualifier="none">titulo</dcvalue> <dcvalue element="type">Trabajo Fin de Grado</dcvalue> </dublin_core></pre>	<pre>fichero1.pdf fichero2.zip</pre>

El método que genera este fichero, puede verse en el [Anexo 8](#)

3.7.2. Instrucción y parámetros

La instrucción que permite la importación es la que se indica a continuación:

```
[root@rabida bin]# pwd
/dspace/bin
[root@rabida bin]# ./import -a -e montserrat.soler@sic.uhu.es -c 10272/5616 -
s /dspace/FichImportar/1 -m /dspace/FichImportar/1/mapfile
```

Hay que tener en cuenta las siguientes notas antes de utilizar los parámetros con el comando *import*

- el correo debe existir en la tabla eperson
- la colección debe indicarse del siguiente modo: 10272/id_coleccion

El primer parámetro **10272**, corresponde al indentificador único asignado a la Universidad de Huelva por Handle System (<http://www.handle.net/>). Handle System ofrece un servicio eficiente y seguro para los identificadores únicos.

El segundo parámetro **id_colección**, corresponde al identificador de la colección dónde se subirá el ítem.

- mapfile debe ser borrado tras terminar la importación. Si no, dará error en la siguiente importación. Para evitar este problema, creamos este fichero en /dspace/FichImportar/1/, asegurándonos de este modo que en la siguiente importación se ubicará en un directorio distinto
- el fichero dublin_core.xml que se genera, no admite tildes. Debe ser compatible 100% con UTF8.
- no se admite el formato:

schema=dc, type other

y se sustituye por:

schema=dc, type

Analicemos uno a uno los parámetros utilizados:

Command used:	<i>[dspace] /bin/dspace import</i>
Java class:	org.dspace.app.itemimport.ItemImport
Arguments short and (long) forms:	Description
-a or --add	Add items to DSpace
-e or --eperson	Email of eperson doing the importing
-c or --collection	Destination Collection by their Handle
-m or --mapfile	Where the mapfile for items can be found (name and directory)

Tras la importación, el fichero mapfile contendrá la URI del ítem importado (véase figura 10):

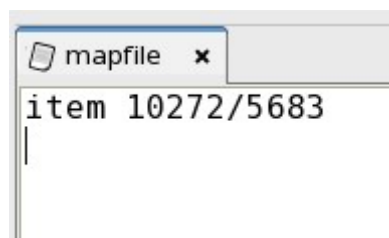


Figura 10

Este fichero debe almacenarse por cada importación en una ruta distinta, pues si existiera en la ruta donde intenta crearse, la importación fallaría.

La importancia de este archivo en todo este proceso es esencial, pues de él se toma el handle, dato imprescindible para los siguientes procesos que van a implementarse:

- retirada del ítem y
- generación de la url que se envía por correo a los miembros del tribunal para permitir la lectura del documento oculto.

3.7.3. Programación

Como ya se ha mencionado en el apartado **2.4. Planificación del desarrollo del proyecto**, existe un comando en Dspace que permite realizar la carga automática de un ítem completo. Ese comando es *import* y los parámetros que se pueden configurar se detallan a continuación:

```
"sudo /dspace/bin/import
-a
-e montserrat.soler@sic.uhu.es
-c 10272/5616
-s /dspace/FichImportar/item1
-m /dspace/FichImportar/item1/mapfile"
```

Dentro del fichero ***CofirmarRegistroProyectoDirectorAction.java***, se realiza la llamada al procedimiento *subirRepositorioItem* con la siguiente instrucción:

```
resultado = dspaceBuh.subirRepositorioItem(Integer.parseInt(idProyecto),
Integer.parseInt(idColeccion), importarSiNo);
```

El procedimiento se crea en el archivo *DspaceBuh.java* con el código indicado en el [Anexo 8](#).

Para que dicho código funcione, debe permitirse al usuario *dspace* (que es el usuario de *tomcat*) ejecutar el comando “*import*” sin necesidad de solicitar la contraseña. Por tal motivo se modifica el fichero */etc/sudoers* (véase figura 11) añadiendo las siguientes líneas:

```

Archivo Editar Ver Terminal Solapas Ayuda
GNU nano 1.3.12 Fichero: /etc/sudoers

#
# Disable "ssh hostname sudo <cmd>", because it will show the password in clear.
#     You have to run "ssh -t hostname sudo <cmd>".
#
# Comentado para evitar pedir tty 02/06/2012
#Defaults    requiretty

#
# Refuse to run if unable to disable echo on the tty. This setting should also be
# changed in order to be able to use sudo without a tty. See requiretty above.
#
Defaults    !visiblepw

Defaults    env_reset
Defaults    env_keep = "COLORS DISPLAY HOSTNAME HISTSIZE INPUTRC KDEDIR \
                        LS_COLORS MAIL PS1 PS2 QDIR USERNAME \
                        LANG LC_ADDRESS LC_CTYPE LC_COLLATE LC_IDENTIFICATION \
                        LC_MEASUREMENT LC_MESSAGES LC_MONETARY LC_NAME LC_NUMERIC \
                        LC_PAPER LC_TELEPHONE LC_TIME LC_ALL LANGUAGE LINGUAS \
                        _XKB_CHARSET XAUTHORITY"

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##     user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
# Se elimina la restriccion de pedir password al usuario dspace 02/06/2012
dspace  ALL=(ALL) NOPASSWD: ALL
    
```

Figura 11

Ocultar el ítem una vez publicado en el repositorio de forma automática.

Cuando un ítem es retirado (ocultado), aparece una línea referente a dicho elemento en la tabla *bi_withdrawn*.

Analicemos la estructura y el contenido de dicha tabla:

- estructura: para mostrar la estructura de esa tabla, usamos la siguiente consulta:

```
sql = "SELECT * FROM information_schema.columns WHERE table_name='bi_withdrawn';"
```

cuyo resultado nos ofrece las siguientes columnas:

- item_id
- id
- sort_2
- sort_1
- sort_3

- contenido: si mostramos el contenido actual de esta tabla, usamos la siguiente consulta:

```
sql = "SELECT item_id, id, sort_1, sort_2, sort_3 FROM bi_withdrawn;";
```

cuyo resultado es:

1569	3	chronique of the conquest of granada	2009-09-18T10:37:13Z	2009-09-18T10:37:13Z
1568	4	tales of the alhambra	2009-09-18T10:32:02Z	2009-09-18T10:32:02Z
2433	5	la encuesta de poblaci?n activa : aspectos metodolo?gicos	2010-01-13T09:23:54Z	2010-01-13T09:23:54Z
2809	6	ambigu?edad y persuasio?n en el dinero de los dioses : el caso de juliano	2010-02-17T13:30:27Z	2010-02-17T13:30:27Z
3130	8	la mineria en la provincia de huelva	2010-04-15T11:48:49Z	2010-04-15T11:48:49Z
2214	9	huelva. plaza de san pedro.	2009-12-18T11:02:42Z	2009-12-18T11:02:42Z
3242	17	castillo de aracena	2010-04-21T10:28:59Z	2010-04-21T10:28:59Z
3246	18	castillo de aracena nevado	2010-04-21T10:33:17Z	2010-04-21T10:33:17Z
3241	19	cera?mica tradicional	2010-04-21T10:27:44Z	2010-04-21T10:27:44Z
3244	20	ermita de san sebastian	2010-04-21T10:30:39Z	2010-04-21T10:30:39Z
4639	23	official masters degree in emergency and critical care nursing	2009	2011-03-30T10:56:30Z
5426	25	prueba openaire	2011-09-09	2011-11-29T08:27:02Z
5620	28	prueba	2012-04-25	2012-04-25T07:47:49Z
5715	29	lo solido a lo liquido: las nuevas alfabetizaciones ante los cambios culturales de la web 2.0	2012-06-02	2012-06-02T07:03:39Z

Como podemos apreciar, el campo *item_id* es el handle del ítem, pero el campo *id*, es un nº secuencial que toma de un contador. Tanto en Oracle como en PostgreSQL, el valor de un contador se toma con la instrucción `nextval('bi_withdrawn_seq')`

El resto de tablas relacionadas en este proceso, se han descrito en el apartado “7.- Retirar el ítem subido al repositorio”

Retirar el ítem una vez publicado en el repositorio de forma manual.

Si el proceso de retirar ítem no se hubiera implementado automáticamente, la forma manual de hacerlo implicaría los siguientes pasos:

1. Acceder al repositorio identificándose en My DSpace (véase figura 12) mediante autenticación en el LDAP.



Figura 12

2. Acceder al ítem y editarlo (véase figura 13):



Figura 13

3. Retirar el ítem (véase figura 14), para que deje de visualizarse:

Inicio Perfil:Montserrat Envíos / Depósitos Contacto Preguntas frecuentes Ayuda

DSpace Principal > Ítems > Estado del ítem >

Editar ítem

Estado del ítem
Archivos del ítem
Metadatos del ítem
Visualizar ítem

Bienvenido a la página de gestión del ítem. Desde aquí podrá retirar, readmitir o borrar el ítem. También podrá modificarlo o añadir nuevos metadatos/archivos en las otras pestañas.

ID interno del ítem:	5719
Handle:	10272/5694
Última modificación:	2012-06-03 22:59:06.802
Página del ítem:	http://rabida.uhu.es/dspace/handle/10272/5694
Autorizaciones del ítem:	Editar autorizaciones
Retirar el ítem del repositorio:	Retirar...
Move item to another collection:	Move...
Borrar completamente el ítem:	Borrar permanentemente

[Volver](#)

Figura 14

4. Tras retirarlo, se confirma el proceso (véase figura 15) y se muestra la siguiente pantalla:



Aviso
The item has been withdrawn.

Editar ítem

Estado del ítem
Archivos del ítem
Metadatos del ítem
Visualizar ítem

Bienvenido a la página de gestión del ítem. Desde aquí podrá retirar, readmitir o borrar el ítem. También podrá modificarlo o añadir nuevos metadatos/archivos en las otras pestañas.

ID interno del ítem:	5719
Handle:	10272/5694
Última modificación:	Sun Jun 03 23:46:53 CEST 2012
Página del ítem:	http://rabida.uhu.es/dspace/handle/10272/5694
Autorizaciones del ítem:	<input type="button" value="Editar autorizaciones"/>
Readmitir el ítem en el repositorio:	<input type="button" value="Readmitir..."/>
Move item to another collection:	<input type="button" value="Move..."/>
Borrar completamente el ítem:	<input type="button" value="Borrar permanentemente"/>

Figura 15

5. En estos momentos, el ítem ha sido retirado. Para volverlo a mostrar, rehacer el mismo proceso y esta vez pulsar sobre *Readmitir*

Readmitir el ítem en el repositorio una vez finalizada la defensa si el alumno lo solicita.

Si el alumno solicita que sea publicado su proyecto en acceso abierto, remitirá a la Biblioteca un escrito que puede bajarse de la página web a la que accede el alumno. Debe firmarlo para así ceder los derechos.

Tras recibir este impreso y una vez terminada la defensa, la Biblioteca deberá readmitir el ítem retirado. Para ello procederá de forma inversa a la mencionada en el apartado anterior. Una vez accedido al ítem, simplemente pulsar el botón destinado a dicho fin (véase figura 16):

Editar ítem

Estado del ítem
Archivos del ítem
Metadatos del ítem
Visualizar ítem

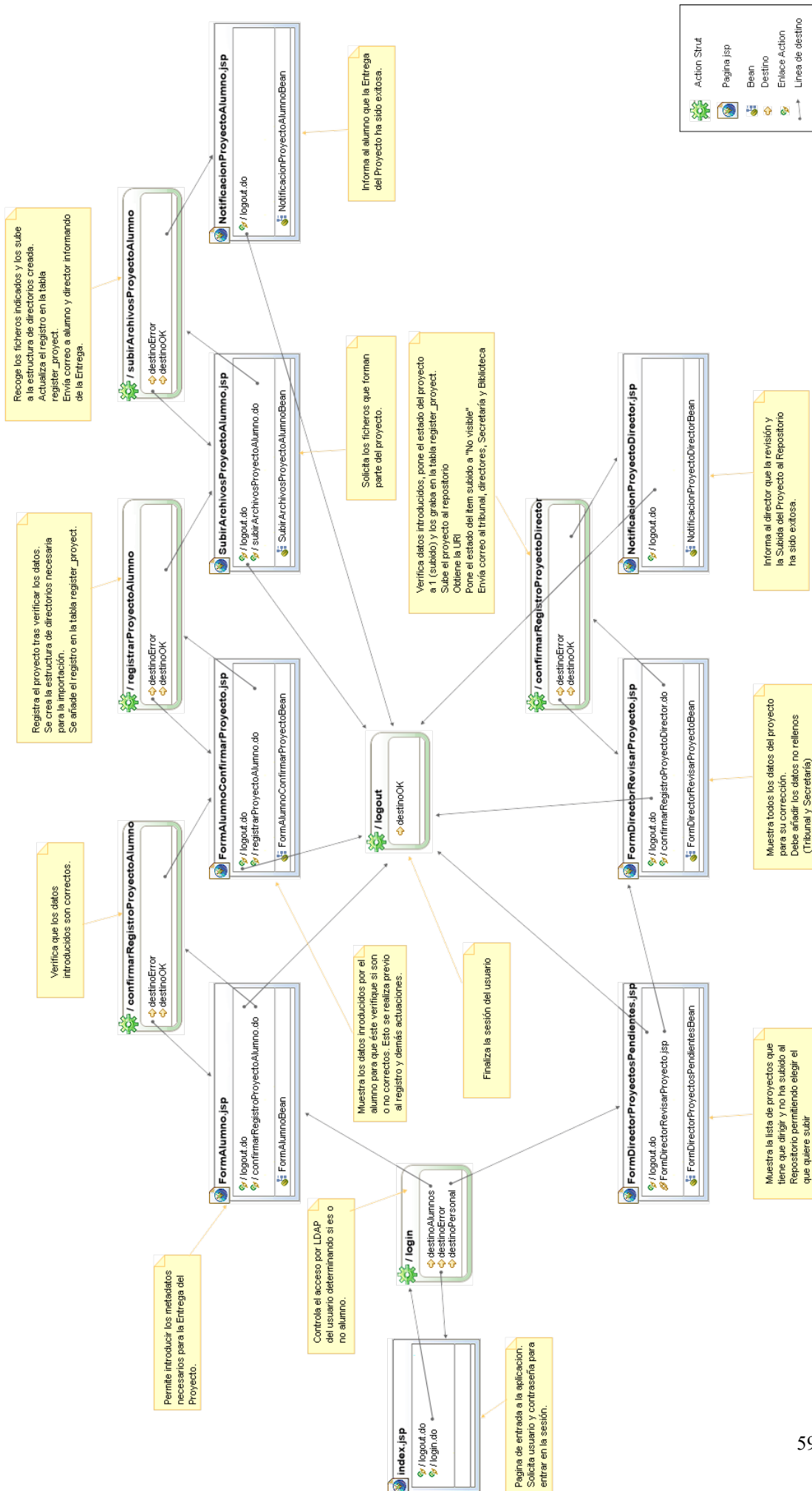
Bienvenido a la página de gestión del ítem. Desde aquí podrá retirar, readmitir o borrar el ítem. También podrá modificarlo o añadir nuevos metadatos/archivos en las otras pestañas.

ID interno del ítem:	5719
Handle:	10272/5694
Última modificación:	Sun Jun 03 23:46:53 CEST 2012
Página del ítem:	http://rabida.uhu.es/dspace/handle/10272/5694
Autorizaciones del ítem:	<input type="button" value="Editar autorizaciones"/>
Readmitir el ítem en el repositorio:	<input type="button" value="Readmitir..."/>
Move item to another collection:	<input type="button" value="Move..."/>
Borrar completamente el ítem:	<input type="button" value="Borrar permanentemente"/>

Figura 16

3.8. Modelo Entidad-Relación

MODELO INTERACION DE PAGINAS Y ACCIONES



3.9. Validación del proyecto: probando la solución propuesta

A continuación, se detalla paso a paso el flujo de trabajo que debe seguir el alumno y el director del TFG durante el proceso de Entrega del trabajo.

Con este detalle, se pretende analizar el trabajo presentado en esta memoria, mostrar su funcionamiento y demostrar su validez. Iniciemos sin más demora, las explicaciones oportunas sobre la solución propuesta.

El flujo de trabajo, para ambos, alumno y director de TFG, se inicia tras entrar en la plataforma a través de la url <http://rabida.uhu.es/dspacebuh>. En esta primera ventana (véase figura 17), el usuario se identifica ante el LDAP de la UHU, determinándose si el usuario es alumno o docente. Dependiendo de quién sea, alumno o docente, la plataforma presenta un flujo de trabajo distinto.

The screenshot shows the login interface for the DSpace repository. At the top left is the University of Huelva logo. To its right is the 'am' logo for 'Arias Montano Repositorio Institucional de la Universidad de Huelva'. In the top right corner, there is a 'Salir' button. The main content area is titled 'Acceder a Dspace' and contains a login form with two input fields: 'Nombre de usuario:' and 'Contraseña:'. Below these fields is a red 'Acceder' button. On the left side, there is a sidebar with the heading 'Enlaces de Interés' and a list of links: 'Documento de cesión de derechos', 'Manual Alumno', 'Manual Docente', and 'Normativa'. At the bottom left of the sidebar is the 'Biblioteca Universitaria Huelva' logo. The footer contains the following text: 'Copyright © 2008-2010. ARIAS MONTANO. Repositorio Institucional de la Universidad de Huelva | rabida.uhu.es/dspace | repositorio@biblio.uhu.es | Tlf.: +34 959 21 93 20

Figura 17

3.9.1. Workflow del alumno

Tras la identificación, se presenta un formulario (véase figura 18) con los datos que debe introducir el alumno. Todos ellos, necesarios para la catalogación del proyecto en Arias Montano.

Alumno: Entrega Trabajo Fin de Grado

Datos personales:

DNI:	<input type="text" value="29050730"/>	Nombre:	<input type="text" value="Montserrat"/>
<input type="text" value="montserrat.soler@alu.uhu.es"/>			
Primer Apellido:	<input type="text" value="Soler"/>	Segundo Apellido:	<input type="text" value="Bernal"/>
Tipo de proyecto a subir:			
<input type="text"/>			
Titulación:			
<input type="text"/>			

Datos del director, codirector:

Email Director:	
<input type="text"/>	
Apellidos:	Nombre:
<input type="text"/>	<input type="text"/>
Email Codirector:	
<input type="text"/>	
Apellidos:	Nombre:
<input type="text"/>	<input type="text"/>

Datos del trabajo:


Título (máx.250 caracteres):

Palabras claves (español) (máx.5 palabras separadas por coma; máx.250 caracteres):

Palabras claves (inglés) (máx.5 palabras separadas por coma; máx.250 caracteres):

Resumen (español) (máx.1000 caracteres):

Resumen (inglés) (máx.1000 caracteres):

Fecha lectura: 

Idioma:

Permite acceso abierto

Figura 18

Cabe mencionar:

- los datos del *alumno*, se muestran automáticamente. Son tomados del LDAP tras su identificación en la pantalla de login, y no pueden ser modificados. Si hay algún error, deberá indicarse al Servicio de Informática para su corrección.
- los valores que presenta el campo *Tipo de proyecto*, son tomados de la base de datos de Dspace tras una consulta a las tablas que contienen dichos datos. Para más detalle, ver apartado 3.6.3. *Extracción de datos (consultas), inserción y actualización*.

- los valores del campo *Titulación* se toman de la base de datos de Dspace. Dependiendo del *Tipo de proyecto* elegido, los valores que se presentan en esta lista serán distintos.
- los datos del *director y codirector*, también son tomados de la base de datos de Dspace. Dependiendo de la *Titulación* elegida, los valores que se presentan en esta lista serán distintos.
- si se cambia el valor del campo *Tipo de proyecto*, automáticamente se borran los datos de los campos *Titulación, director y codirector*, pues la nueva elección implicará otros valores posibles para los campos titulación, dirección, ...
- si se cambia el valor del campo *Titulación*, automáticamente se borran los datos de los campos *director y codirector* por el motivo indicado en el punto anterior.
- existe un campo oculto ***Estado*** que se rellena a 0 tras la confirmación de los datos por parte del alumno. Este campo identificará que el proyecto ha sido confirmado por el alumno pero no por el director, y por tanto, no ha sido subido al repositorio. En el momento que sea subido al repositorio, al estado del proyecto le será asignado el valor 1.
- todos los datos que se muestran en el formulario, excepto los datos del codirector, deben ser introducidos obligatoriamente antes de confirmar el envío.

Tras rellenar el formulario, antes de proceder al envío, se presenta una pantalla de confirmación (véase figura 19) con los datos introducidos:

Alumno: Entrega Proyecto Fin de Grado

Datos personales:

DNI:	<input type="text" value="29050730"/>	Nombre:	<input type="text" value="Montserrat"/>
	<input type="text" value="montserrat.soler@alu.uhu.es"/>		
Primer Apellido:	<input type="text" value="Soler"/>	Segundo Apellido:	<input type="text" value="Bernal"/>
Tipo de proyecto a subir:	<input type="text" value="Proyectos Fin de Grado"/> ▼ 84		
Titulación:	<input type="text" value="Grado en Ingeniería Informática"/> ▼ 360		

Datos del director, codirector:

Email Director:	<input type="text" value="ndirector1.adirector1@biblio.uhu.es"/> ▼		
Apellidos:	<input type="text" value="adirector1"/>	Nombre:	<input type="text" value="ndirector1"/>
Email Codirector:	<input type="text"/>		
Apellidos:	<input type="text"/>	Nombre:	<input type="text"/>

Datos del proyecto:

Título (máx.250 caracteres):
 Solución integral automatizada a los PFG de la Universidad de Huelva a través del Repositorio Institucional Arias Montano

Palabras claves (español) (máx.5 palabras separadas por coma; máx.250 caracteres):
 DSpace, Repositorio, metadatos, Proyecto Fin de Grado

Palabras claves (inglés) (máx.5 palabras separadas por coma; máx.250 caracteres):
 DSpace, repository, metadata, Final Year Project

Resumen (español) (máx.1000 caracteres):
 integral automatizada a la entrega y almacenamiento de los Proyectos Fin de Grado de la Universidad de Huelva, tomando como elemento contenedor de los proyectos, el Repositorio Institucional Universitario Arias Montano.

Resumen (inglés) (máx.1000 caracteres):
 This project aims to provide an automated solution to the delivery and storage of Final Year Projects at the University of Huelva, on the container element of the project, the University Institutional Repository Arias Montano.

Fecha lectura: 02/07/2012

Idioma: español

Permite acceso abierto Sí [Información Acceso Abierto](#)

Presentar Proyecto

Figura 19

El alumno revisará los datos mostrados. Si observa algún error, pulsará sobre el botón “NO confirmar”, regresando a la pantalla anterior que le permitirá modificar los datos. Si el alumno “Confirma los datos”, se muestra una pantalla (véase figura 20) informando que los datos han sido grabados y solicitando los ficheros que contienen la memoria y/o ejecutables.

Alumno: Entrega Proyecto Fin de Grado

Subir archivos

D.Ñña.:

Para terminar la entrega de su proyecto:

Solución integral automatizada a los PFG de la Universidad de Huelva a través del Repositorio Institucional Arias Montano

debe subir los ficheros que contienen su memoria.

Fichero pdf (obligatorio): contendrá la memoria de su proyecto

Fichero zip ó rar (opcional): contendrá los ejecutables comprimidos

Figura 20

Es obligatorio introducir el fichero que contiene la memoria con formato pdf. El archivo con formato comprimido zip o rar es opcional. Contendrá todos los ejecutables desarrollados en el proyecto.

Tras pulsar sobre el botón *Subir archivos*, se inicia la carga del archivo. Esta puede tardar varios minutos dependiendo del peso del archivo. Mientras está subiéndose, se muestra la pantalla de la figura 21:

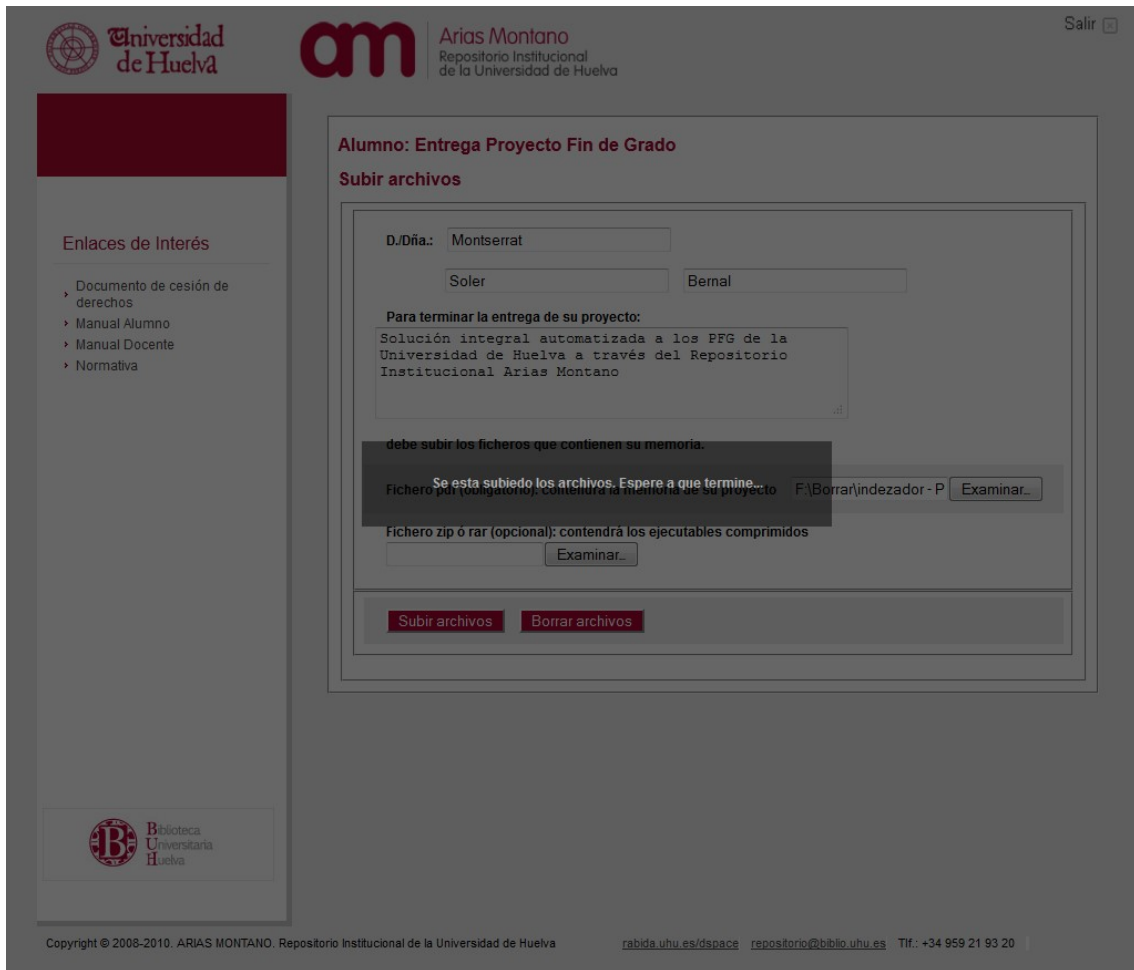


Figura 21

En el momento que finaliza el almacenamiento del archivo, se muestra la pantalla de la figura 22, notificando que la entrega ha sido realizada:

Alumno: Entrega Proyecto Fin de Grado
Sus datos han sido confirmados

D./Dña.:

Su proyecto:
 Solución integral automatizada a los PFG de la Universidad de Huelva a través del Repositorio Institucional Arias Montano

y sus ficheros anexos al proyecto han sido grabados.

En este instante, su director de proyecto, está siendo informado de la entrega a través del correo:

Usted recibirá un correo con los datos de su proyecto. Guárdelo como comprobante de la Entrega.

Figura 22

En el momento que se procede al envío de los ficheros:

- se crea un directorio dentro de la estructura /dspace/FichImportar con el nombre del identificador del proyecto seguido del directorio ítem (véase figura 23), donde serán almacenados ambos ficheros. La estructura resultante será:

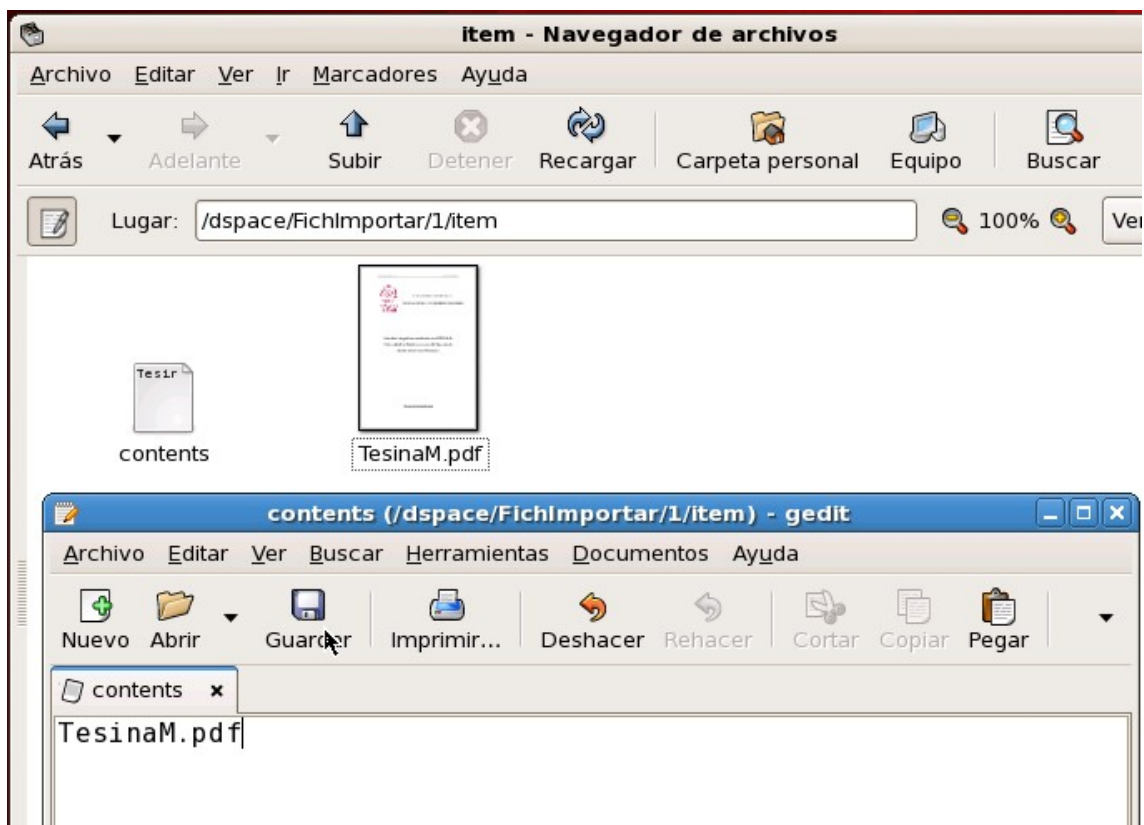


Figura 23

Al mismo tiempo se genera un fichero denominado **contents** que poseerá una línea por cada fichero subido, tal como se ha visto en la ventana anterior.

Toda esta estructura (/dspace/FichImportar/29), será importada utilizando el comando **import** al Repositorio junto al fichero dublin_core.xml, generado con los metadatos introducidos por el alumno y confirmados por el director.

- Se graban todos los datos introducidos por el alumno en una tabla creada en Dspace para dicho fin (*registerproject*)
- Se envía un correo (véase figura 24) al alumno para que quede constancia de su envío y al director para que sepa que tiene un proyecto pendiente de subir al repositorio.

Asunto: Comprobante de la Entrega de su Proyecto.
De: repositorio@biblio.uhu.es
Fecha: Sab, 9 de Junio de 2012, 5:53 pm
Para: ndirector1.adirector1@biblio.uhu.es ([menos](#))
montserrat.soler@alu.uhu.es
Prioridad: Normal
Opciones: [Ver encabezado completo](#) | [Vista preliminar](#) | [Spam](#) | [No Spam](#)

El proyecto de D./D^a. Montserrat Soler Bernal, con D.N.I. 29050730 ha sido entregado.

Los datos del proyecto son:

* TIPO: Proyectos Fin de Grado

* TITULACIÓN: Grado en Ingeniería Informática

* DIRECTOR: adirector1, ndirector1

* CODIRECTOR: ,

* TÍTULO: Solución integral automatizada a los PFG de la Universidad de Huelva a través del Repositorio Institucional Arias Montano

* PALABRAS CLAVES (spi): Dspace, Repositorio, metadatos, Proyecto Fin de Grado

* PALABRAS CLAVES (eng): DSpace, repository, metadata, Final Year Project

* RESUMEN (spi): Este proyecto intenta ofrecer una solución integral automatizada a la entrega y almacenamiento de los Proyectos Fin de Grado de la Universidad de Huelva, tomando como elemento contenedor de los proyectos, el Repositorio Institucional Universitario Arias Montano.

* RESUMEN (eng): This project aims to provide an automated solution to the delivery and storage of Final Year Projects at the University of Huelva, on the container element of the project, the University Institutional Repository Arias Montano.

* FECHA LECTURA: 02/07/2012

* IDIOMA: spa

* PERMITE ACCESO: si

* FICHERO 1 (memoria pdf): datos proyecto.pdf

* FICHERO 2 (ejecutables zip o rar):

Guarde este correo como comprobante de la Entrega de su Proyecto

[Bajar este mensaje como un archivo](#)

Figura 24

En este momento, la función del alumno en la entrega del proyecto ha finalizado. Ya solo debe prepararse para su defensa ante el tribunal.

3.9.2. Workflow del director

El docente se introduce dentro del flujo de trabajo en el momento que recibe un correo notificándole que un TFG se ha presentado y que él es el director. El modelo de correo recibido es el mostrado en el punto anterior.

Al entrar en la plataforma e identificarse ante el LDAP, se muestra un formulario con todos los trabajos que él dirige y cuyo estado está a 0, es decir, trabajos que aún no han sido subidos al repositorio (véase figura 25). En el momento que se suban se les asigna automáticamente un 1.



Director: Entrega Trabajo Fin de Grado

Trabajos pendientes de carga

Trabajos pendientes de subir al Repositorio:

Identificador del trabajo: 19
Título : Solución integral automatizada a los PFG de la Universidad de Huelva a través del Repositorio Institucional Arias Montano

DNI alumno: 29050730 **Nombre :** Montserrat
Primer apellido : Soler **Segundo apellido :** Bernal

[Ver trabajo](#)

Figura 25

Tras seleccionar el proyecto indicado en el correo, se abre un formulario (véase figura 26) con todos los datos introducidos por el alumno.

Director: Entrega Proyecto Fin de Grado

Confirmación de datos

Datos personales:

DNI: **Nombre:**

Primer Apellido: **Segundo Apellido:**

Tipo de proyecto a subir:

Titulación:

Datos del director, codirector:

Email Director:

Apellidos: **Nombre:**

Email Codirector:

Apellidos: **Nombre:**

Datos del Tribunal:

Email miembro1 tribunal:

Apellidos: **Nombre:**

Email miembro2 tribunal:

- teresa.contreras@biblio.uhu.es
- susana.moreno@biblio.uhu.es
- ntribunal1.tribunal1@biblio.uhu.es
- ntribunal2.tribunal2@biblio.uhu.es**
- ntribunal3.tribunal3@biblio.uhu.es
- ntribunal4.tribunal4@biblio.uhu.es
- ntribunal5.tribunal5@biblio.uhu.es
- ndirector1.adirector1@biblio.uhu.es
- ndirector2.adirector2@biblio.uhu.es

Nombre:

Apellidos: **Nombre:**

Datos de su centro:

Elija su centro:

Datos del proyecto:


Título (máx.250 caracteres):
 Solución integral automatizada a los PFG de la Universidad de Huelva a través del Repositorio Institucional Arias Montano

Palabras claves (spi)(máx.5 palabras separadas por coma; máx.250 caracteres):
 Dspace, Repositorio, metadatos, Proyecto Fin de Grado

Palabras claves (eng)(máx.5 palabras separadas por coma; máx.250 caracteres):
 DSpace repository, metadata, Final Year Project

Resumen (spi)(máx.1000 caracteres):
 Este proyecto intenta ofrecer una solución integral automatizada a la entrega y almacenamiento de los Proyectos Fin de Grado de la Universidad de Huelva, tomando como elemento contenedor de los proyectos, el Repositorio

Resumen (eng)(máx.1000 caracteres):
 This project aims to provide an automated solution to the delivery and storage of Final Year Projects at the University of Huelva, on the container element of the project, the University Institutional Repository Arias Montano.

Fecha lectura: 

Idioma:

Fichero 1: (Memoria pdf)

Fichero 2: (Aplicacion zip)

Permite acceso abierto

[Subir datos Repositorio](#)

Figura 26

Cabe mencionar:

- existen datos que el director no puede modificar:
 - los datos del alumno se cogen del LDAP,

- el tipo de proyecto a subir, la titulación, los datos del director y codirector se cogen de Dspace y están relacionados entre sí (dependiendo del tipo de trabajo elegido aparecen unas titulaciones u otras,...),
- los ficheros que contienen la memoria y la aplicación.

Se supone que todos estos datos son básicos y no hay posible confusión en el momento de ser introducidos por el alumno.

- otros datos deben ser introducidos obligatoriamente por el director:
 - los miembros del tribunal
 - la escuela o facultad a la que pertenece el trabajo presentado.
- existen datos que debe revisar y corregir sin son incorrectos: titulo, palabras claves (spi, eng), resumen (spi, eng), fecha lectura, idioma, permite acceso abierto.

Tras la verificación de los datos, se procede a “Subir datos al Repositorio”. Esta acción implica:

- actualizar la tabla (*registerproject*) y poner el estado del proyecto a 1 para que no sea mostrado al director como tarea pendiente de subir.
- generar el fichero dublin_core.xml (véase figura 27)

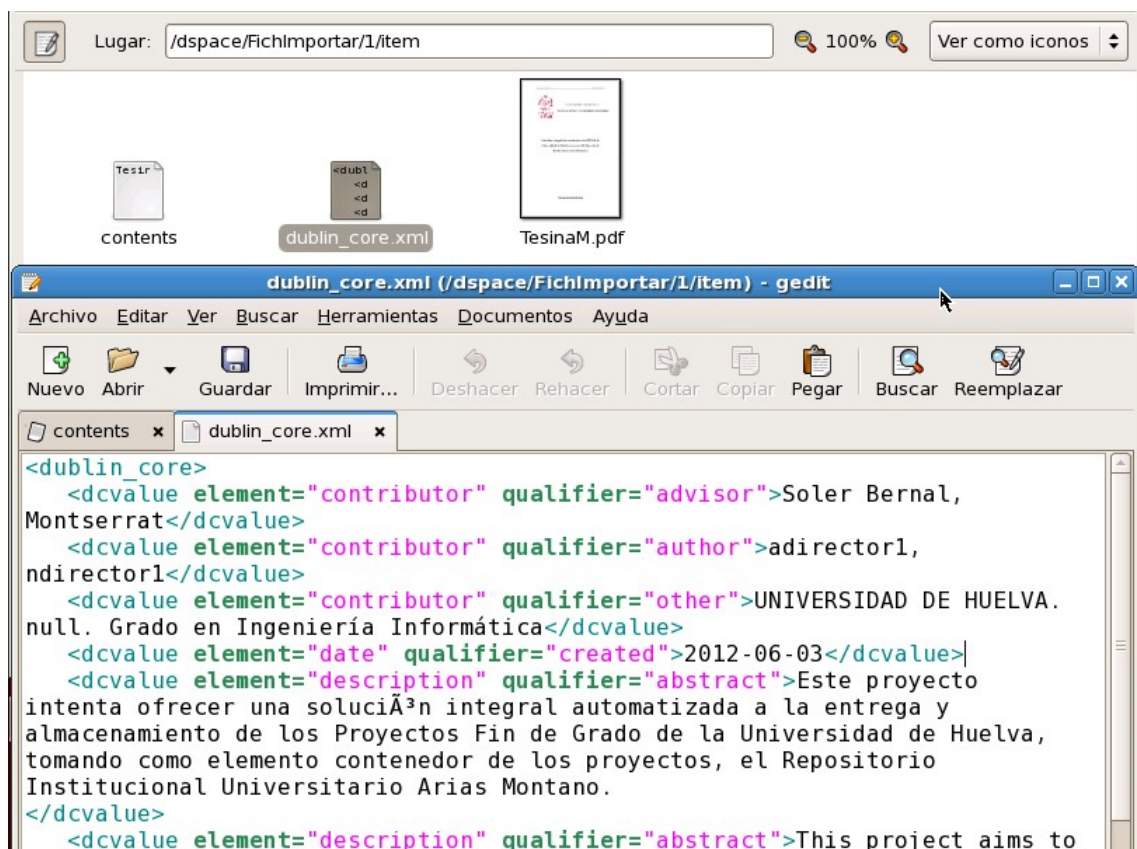


Figura 27

- subir el repositorio, mediante el comando import, toda la estructura ubicada en /dspace/FichImportar/idProyecto/item

- extraer la url con el handle que identifica al trabajo subido (URI): esa url se haya en el fichero mapfile generado tras la importación
- modificar a ítem oculto el trabajo subido, para que solo pueda ser leído por el grupo administrador de la colección (director, tribunal, ...)
- enviar correos informando del acceso al proyecto al tribunal, al director, al codirector si lo hubiera, a la Secretaría y a la Biblioteca. De esta forma:
 - el director, codirector y tribunal podrá leer el TFG.
 - La Secretaría tendrá notificación de la entrega del trabajo para realizar el acta.
 - La Biblioteca será informada de la subida de un ítem.

La ventana notificando al director del TFG que ha finalizado su función en la entrega, se muestra en la figura 28:

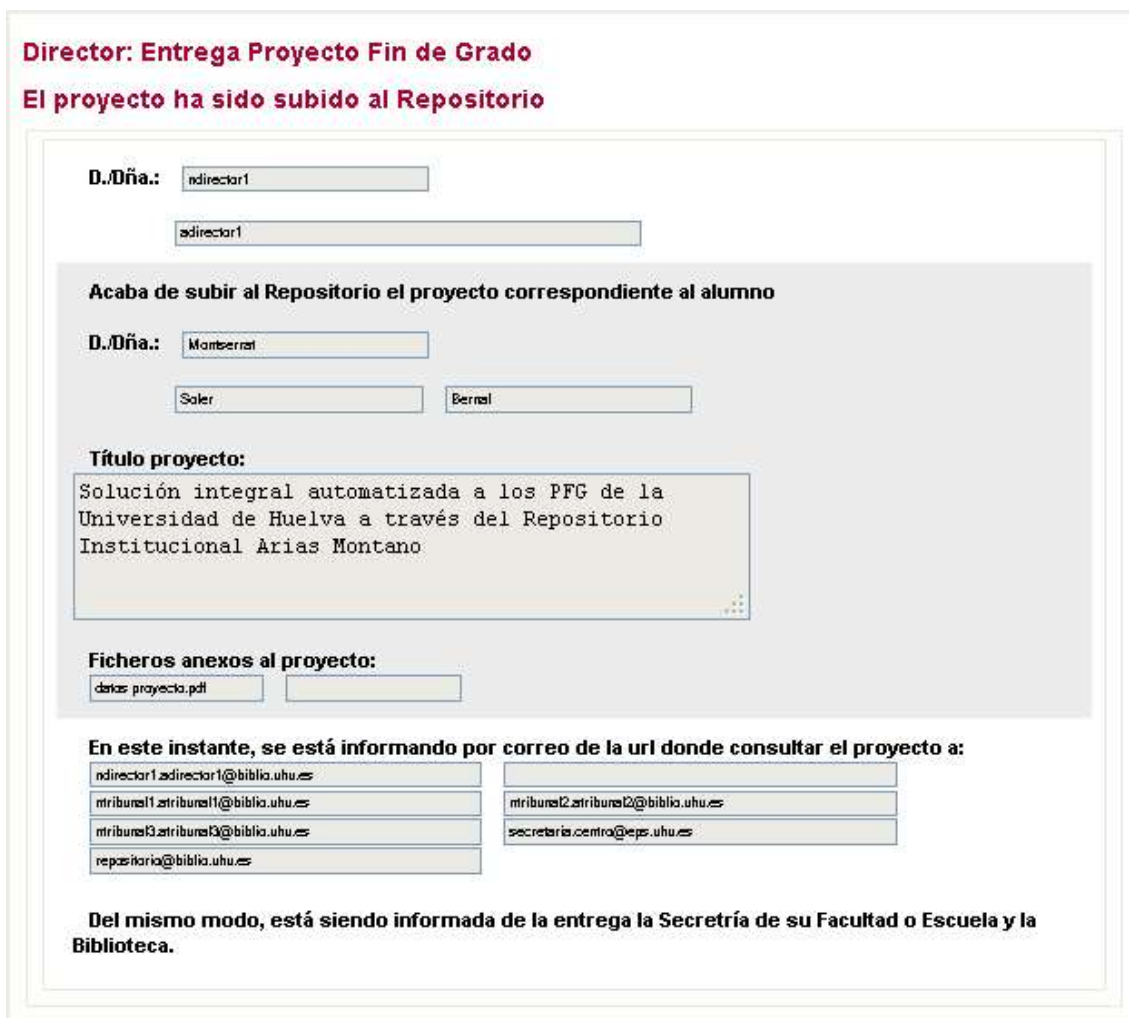


Figura 28

El correo que se recibe tras la importación del ítem al Repositorio se muestra en la figura 29:

Asunto: Proyecto subido al Repositorio Arias Montano.
De: repositorio@biblio.uhu.es
Fecha: Dom, 10 de Junio de 2012, 1:57 pm
Para: ndirector1.adirector1@biblio.uhu.es ([menos](#))
ntribunal1.atribunal1@biblio.uhu.es
ntribunal2.atribunal2@biblio.uhu.es
ntribunal3.atribunal3@biblio.uhu.es
secretaria.centro@eps.uhu.es
repositorio@biblio.uhu.es

Prioridad: Normal
Opciones: [Ver encabezado completo](#) | [Vista preliminar](#) | [Spam](#) | [No Spam](#)

PROPUESTA DE DEFENSA DE Proyectos Fin de Grado

* Alumno/a: Montserrat Soler Bernal
* DNI: 29050730

* Título: Solución integral automatizada a los PFG de la Universidad de Huelva a través del Repositorio Institucional Arias Montano

* Tipo de proyecto: Proyectos Fin de Grado

* Titulación: Grado en Ingeniería Informática

TRIBUNAL

* Director: ndirector1 adirector1
* Codirector:
* Tribunal: ntribunal1 atribunal1
* Tribunal: ntribunal2 atribunal2
* Tribunal: ntribunal3 atribunal3

Figura 29

Para verificar que todo el proceso ha concluido satisfactoriamente, sólo acceder a la URI que indica el correo (véase figura 30), y comprobar que el ítem ha sido subido y que los datos son correctos:

The screenshot shows the Arias Montano Institutional Repository interface. At the top, there is the logo of the University of Huelva and the 'am' logo for Arias Montano. A search bar is visible with the text 'Buscar en Arias Montano:'. Below the search bar, there are navigation links: 'Inicio', 'Mi DSpace', 'Registrar', 'Contacto', 'Preguntas frecuentes', and 'Ayuda'. A breadcrumb trail indicates the location: 'DSpace Principal > Aprendizaje - Docencia > Proyectos Fin de Grado > Grado en Ingeniería Informática > Ver ítem >'. The main content area displays the title 'Solución integral automatizada a los PFG de la Universidad de Huelva a través del Repositorio Institucional Arias Montano' and a list of metadata fields:

Título:	Solución integral automatizada a los PFG de la Universidad de Huelva a través del Repositorio Institucional Arias Montano
Autoría:	Soler Bernal, Montserrat
Dirección:	adirector1, ndirector1
Departamentos:	UNIVERSIDAD DE HUELVA. Escuela Técnica Superior de Ingeniería. Grado en Ingeniería Informática
Autoría:	adirector1, ndirector1; Soler Bernal, Montserrat; UNIVERSIDAD DE HUELVA. Escuela Técnica Superior de Ingeniería. Grado en Ingeniería Informática
Fecha de publicación:	2012-06-09
Fecha de entrada:	2012-06-09
Fecha disponible:	2012-06-09
Fecha de creación:	2012-06-09
Tipo de documento:	Proyectos Fin de Grado
Idioma:	spa

On the left side, there are navigation menus for 'Listar por' (Communities and collections, Titles, Author, Matter, By publication date) and 'De interés' (Collection of doctoral theses, Latest Deposits, Search engines and harvesters that index Arias Montano). There are also 'Otros enlaces' (Autoarchive, SHERPA/RoMEO, Dulcinea, Other resources, AccesoAbierto.net).

Figura 30

Por último, y para concluir esta demostración, mencionar que todo error producido durante la intervención del alumno o director, será indicado en la parte superior de la ventana, tras el encabezado, tal como se muestra en la figura 31.

The screenshot shows an error message at the top: 'ERROR: No esta autorizado para entrar....'. Below the error message is a login form titled 'Acceder a Dspace'. The form contains two input fields: 'Nombre de usuario:' and 'Contraseña:'. At the bottom of the form is a red button labeled 'Acceder'.

Figura 31

El trabajo ya ha sido subido al repositorio y está retirado (oculto). Sólo podrán acceder a él, el tribunal que ha recibido un correo informando de la url.

No olvidar que tras la defensa del proyecto, la Biblioteca, si así ha sido solicitado por el alumno, hará público el proyecto en el Repositorio.

3.10. Informe final con cuestiones que deben mencionarse

Algunas cuestiones que deben indicarse respecto a este proyecto son:

3.10.1.- Visualización de las tildes y la ñ

A la hora de programar JSP y estando configurada la base de datos, las tablas y el archivo JSP en UTF-8, no se consigue mostrar la ñ y acentos de manera adecuada. Aparentemente, el navegador o el servidor o ambos, en el momento de transmitir los datos ignoran la especificación del charset. Para solucionar este problema, se añaden, las siguientes directivas en todas las páginas jsp:

- Al principio de la página jsp, se añade la directiva:

```
<%@page pageEncoding="iso-8859-1" contentType="text/html; charset=iso-8859-1" %>
```

- Dentro de la sección <head>, se añade la directiva:

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

Tras la inserción de este código, la visualización es correcta.

3.10.2.- Añadir una fila de mensajes (al estilo de linkedin)

Se crea una estructura para que muestre los mensajes de error al principio de la página, al igual que lo hace **linkedin**. Para esta finalidad, se añade el siguiente código en todas las páginas jsp:

```
<html:messages id="msj" message="true" property="aviso"><div class="msg"><bean:write name="msj" /></div></html:messages>
```

3.10.3.- Creación de un log para el control de errores

Para controlar los errores, se crea un fichero de tipo log (véase figura 32) que permite recoger los errores producidos en el código para determinar con mayor claridad la causa del problema y poder localizarlo con mayor facilidad.



Figura 32

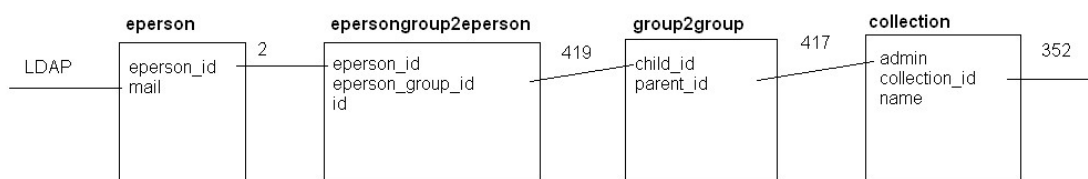
3.10.4.- Conexión con el Cliente de correo

Se implementa un cliente de correo para poder enviar los correos tras la entrega del TFG y la subida de éste al repositorio institucional.

3.10.5.- Extraer el nombre de la colección a la que pertenece un docente conociendo su email extraído de LDAP al identificarse

En un determinado momento, es necesario conocer el nombre de la titulación que un docente puede administrar, y sólo se conoce su correo extraído al identificarse ante el LDAP.

Queda claro que partiendo de la tabla **eperson** que contiene el email del docente debemos enlazar con la tabla **collection** donde se halla el nombre de la titulación. El recorrido se muestra en la siguiente figura:



La estructura de las tablas puede revisarse en apartados anteriores.

Se expone a continuación un ejemplo, para demostrar el funcionamiento interno de Dspace:

- Al crearse una Colección en Dspace, automáticamente se crea un Grupo de Administradores para dicha colección:

En el ejemplo de la figura 33, se crea la colección **Comunicar – V. 37., (2011)** y Dspace le asigna el identificador **352**



Figura 33

Al mismo tiempo, Dspace crea el Grupo de Administradores **COLLECTION_352_ADMIN**, para esa colección, tal como se indica en al figura 34.



Figura 34

- A ese Grupo de Administradores, puede asignársele distintos grupos para que administren dicha colección. Por tal motivo, se crea **Grupo_tit_informatica** al que Dspace asigna automáticamente el identificador **419** (véase figura 35).



Figura 35

- A ese grupo, se le asignan dos docentes (que pueden pertenecer al tribunal de la colección que acabamos de crear).
- Tras ser creado el grupo, es asignado al grupo de Administradores de la Colección (véase figura 36).

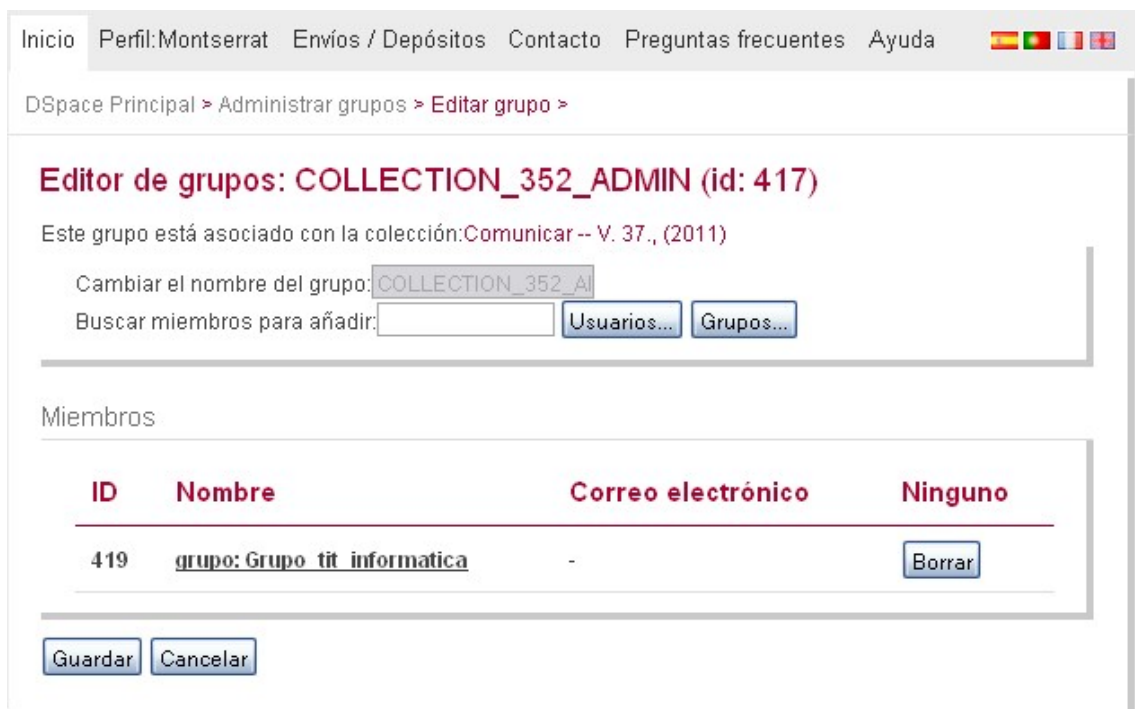


Figura 36

- Las consultas necesarias para extraer el id_colección a partir del email son:

```
SELECT eperson_id, mail
FROM eperson
WHERE email='teresa.contreras@biblio.uhu.es';
      2   teresa.contreras@biblio.uhu.es
```

```
SELECT eperson_id, eperson_group_id, id
FROM epersongroup2eperson
WHERE eperson_id=2;
      2419   481
```

```
SELECT child_id, parent_id,
FROM group2group
WHERE child_id =419;
      419   417
```

```
SELECT admin, collection_id, name
FROM collection
WHERE admin =417;
      417   352   Comunicar -- V. 37., (2011)
```

4. Conclusiones y líneas de trabajos futuras

4.1. Conclusiones

Con este trabajo se pretende informatizar la “Entrega de los Trabajos Fin de Grado”. La informatización se realiza creando una aplicación JEE que se integra al cien por cien dentro del Repositorio Institucional de la Universidad de Huelva Arias Montano, desarrollado bajo Dspace, software open access.

Este nuevo desarrollo, presenta un aspecto completamente integrado con el repositorio y permite una total interrelación con la base de datos de Dspace, permitiendo de este modo la integración de información entre aplicaciones.

La *finalidad* del proyecto que en esta memoria se describe, es catalogar en Arias Montano el trabajo que el alumno introduce en este nuevo programa tras la intervención del director del TFG y sin recaer el trabajo sobre el personal bibliotecario. El trabajo no se duplica en la Biblioteca, se aprovechan los datos proporcionados por alumno y director para realizar la catalogación.

Por otra parte, proporciona una forma fácil y cómoda de poner a disposición del tribunal el trabajo del alumno para su lectura, respetando uno de los compromisos medioambientales de esta Universidad “La preservación de la naturaleza supone la eliminación del papel”. Cabe mencionar, el ahorro económico y de tiempo que implica esta nueva forma de entrega para el alumno.

A continuación se exponen las *características y funcionalidades* que implica hoy día la entrega de un TFG y las que tendría si se informatizara este proceso.

Actualmente, la “entrega” implica:

- presentar tres manuales impresos de la memoria del TFG al director de proyecto. Éste los repartirá entre los miembros que formarán parte del tribunal de la defensa del trabajo.
- presentar un CD con la memoria y ejecutables en la Secretaría.
- el director asignado al TFC solicita a su Secretaría el Acta mediante un impreso denominado "*Propuesta de defensa de trabajo fin de carrera*" (<http://www.uhu.es/etsi/impresos/propuestaDef.pdf>). Desde la recepción de ese documento, la Secretaría podrá generar el Acta de Evaluación que se rellenará el día de la defensa.
- el día de la defensa del TFC ante el tribunal, el Acta será recogida por el director. Tras la defensa, evaluación y calificación del trabajo, el director entregará el Acta nuevamente en la Secretaría para que la calificación figure en el expediente del alumno.
- la Secretaría remitirá a la Biblioteca el CD para que pueda ser catalogado y almacenado para su posterior consulta tras la defensa si el alumno lo permite.

- si el alumno permite la consulta del proyecto, deberá entregar en la Biblioteca un impreso firmado, por el que queda constancia su aprobación.

El trabajo que se desarrolla en esta memoria pretende que la “entrega” se realice a través de una aplicación que crea un flujo de trabajo entre alumno, director, tribunal, la Secretaría y la Biblioteca, de tal forma que cada uno tiene un rol asignado. A continuación se detalla el rol de cada uno:

- Alumno: previa identificación ante el LDAP universitario, que lo identifica como alumno, rellena un formulario con los datos necesarios para la catalogación del trabajo y sube tanto los archivos de memoria como los ejecutables, que constituyen su TFG. Recibirá un correo como comprobante de la entrega.
- Director: tras recibir un correo cuando un alumno suyo realiza una entrega, entra en la aplicación creada y previa identificación ante el LDAP, le reconocerá como docente y le presentará todos los trabajos entregados por sus alumnos que aún no han sido subidos al repositorio. Tras seleccionar uno, revisará los datos introducidos por el alumno y si es necesario los corregirá, también introducirá los datos del tribunal y la secretaria de su centro. Posteriormente subirá el trabajo al Repositorio y automáticamente quedará retirado para ser consultado únicamente por los miembros del tribunal y por la Biblioteca.
- Secretaría: recibirá un correo en el momento que el trabajo ha sido subido al repositorio. La información proporcionada en ese correo le permitirá generar el Acta necesaria para el día de la defensa del proyecto.
- Miembros del tribunal y director: en el correo recibido se les indica la url de acceso al trabajo. De esta forma podrán leerlo desde sus ordenadores o cualquier lector si se lo bajan a un pen o cualquier otro dispositivo de almacenamiento.
- Biblioteca: si tras la defensa recibe firmado por el alumno el escrito de cesión de derechos, podrá publicarlo en abierto en el Repositorio Institucional Arias Montano

Como aportaciones finales a esta conclusión, mencionar las ventajas que implicaría la puesta en marcha de este proyecto:

- Ahorro de tiempo y dinero para el alumno.
- Sostenibilidad con el medio ambiente.
- Ahorro de tiempo y espacio para la Biblioteca.
- El trabajo no se duplica sino que se aprovecha. La información proporcionada por alumno y director sirven a la Secretaría y a la Biblioteca para desarrollar su trabajo.
- Da un carácter más formal a la entrega del TFG.
- Posibilidades de ampliar este desarrollo a la entrega de otros trabajos como trabajos fin de master, tesis,...

4.2. Líneas de Trabajo Futuras

Son muchas las ideas que pueden surgir y desarrollarse tras la implantación de este proyecto.

La primera de ellas sería ofrecer un “*kit de implementación de este proyecto dentro del contexto de la web de la UHU*”, ofreciendo de esta forma, un modo de acceso para profesor y alumno claro y rápido.

Una posibilidad sería integrarlo dentro de la plataforma de Administración Electrónica como se indica en la figura 37 (<http://ae.uhu.es/>).



Figura 37

En ella existen procedimientos destinados al Estudiante (véase figura 38) y al PDI, bastaría con introducir una entrada en ambos grupos titulada “Entrega de TFG”, y habilitar para su acceso únicamente el icono *Acceso sin certificado*.



Figura 38

De esta forma, la entrega tendría un fácil acceso para alumnos y profesores y tendría un carácter oficial.

La segunda idea sería ***incluir este desarrollo dentro de un procedimiento de Administración Electrónica*** mediante el “*Acceso con certificado*” De esta forma la entrega adquiere carácter institucional quedando registrado telemáticamente todo el proceso de entrega en la que interviene alumno, director, miembros del tribunal, la Secretaría y la Biblioteca.

La tercera idea, implicaría ***la ampliación de este proyecto a la entrega de todo tipo de trabajos***, no solo TFG. Podría abarcar todo tipo de trabajos que se realice en la Universidad de Huelva con carácter institucional: Trabajos Fin de Máster, Tesis,....

Para este fin, solo debería ampliarse las subcomunidades pertenecientes a *Aprendizaje y Docencia*, creando una por cada tipo de trabajos a englobar (Trabajo Fin de Máster,...). Dentro de cada subcomunidad creada, deberá crearse las colecciones correspondientes a la titulación del trabajo a entregar y crear los grupos de profesores que pueden administrar dicha colección.

4.3. Normativa de entrega de Trabajos Fin de Grado

Hoy día, la normativa de entrega de TFG no existe, pues aún no se ofrece en la oferta académica el último curso de los grados que tiene la Universidad de Huelva. Cabe mencionar, que ante la creación de esta Normativa, si este proyecto se llevara a cabo, debería tenerse en cuenta los siguientes apartados:

- Todos los docentes que formen parte del tribunal de un trabajo, deben darse de alta en el Repositorio de Arias Montano. (<http://rabida.uhu.es/dspace/login>)
- El director deberá proporcionar a la Biblioteca, los datos de los docentes que formarán parte del tribunal para incluirlos en el grupo de administradores de la colección, en el caso que no estuvieran incluidos.
- La URL de acceso a la entrega de trabajos, estará incluida como uno de los procedimientos de Administración Electrónica (<http://ae.uhu.es/>)
- El alumno deberá entregar firmado en la Biblioteca, los derechos de acceso a su trabajo si permite su catalogación en abierto.
- Todos los documentos subidos a Arias Montano, están protegidos por derechos de autor y sujetos a la licencia Creative Commons
- El alumno deberá subir su memoria en formato pdf protegido contra copia.
- Si el alumno necesitara subir la programación de su trabajo, la entregará en un único archivo con formato comprimido zip o rar.
- A todos los efectos, en el momento que la Secretaría reciba el correo notificando que un trabajo ha sido entregado y subido al repositorio, elaborará el Acta de entrega del trabajo al igual que actualmente lo realiza cuando el director solicita a su Secretaría el Acta mediante un impreso denominado "Propuesta de defensa de trabajo fin de carrera" (<http://www.uhu.es/etsi/impresos/propuestaDef.pdf>).

5. Bibliografía

DuraSpace - Dspace Manual: Release 1.6.2. - 2010

http://www.dspace.org/1_6_2Documentation/DSpace-Manual.pdf

Fundación Wikipedia – Modelo Vista Controlador - 2012

http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

Oracle - Java SE Downloads

<http://java.sun.com/javase/downloads/index.jsp>

The Apache Software Foundation - Apache Maven Project – 2002-2012

<http://maven.apache.org/download.html>

The Apache Software Foundation - The Apache Ant Project - 2011

<http://ant.apache.org>

The Apache Software Foundation - Apache Tomcat – 1999-2012

<http://tomcat.apache.org/whichversion.html>

The PostgreSQL Global Development Group – PostgreSQL - 2012

<http://www.postgresql.org/>

6. Anexos

Anexo 1: Oferta académica en Grados:

<http://www.uhu.es/estudios/ofertaacademica/grados.html>

<p>Facultad de Ciencias Experimentales Grado en Ciencias Ambientales Grado en Geología Grado en Química Doble Grado en Ciencias Ambientales y Geología</p> <p>Facultad de Humanidades Grado en Filología Hispánica Grado en Historia Grado en Estudios Ingleses</p> <p>Facultad de Ciencias de la Educación Grado en Educación Social Grado Maestro de Educación Infantil Grado Maestro de Educación Primaria Grado en Psicología Grado en Ciencias de la Actividad Física y del Deporte</p> <p>Facultad de Ciencias Empresariales Grado en Administración y Dirección de Empresas Doble Grado Administración y Dir. de Empresas y Turismo Grado en Finanzas y Contabilidad Doble Grado en ADE, Finanzas y Contabilidad Grado en Turismo</p> <p>Facultad de Derecho Grado en Derecho</p> <p>Escuela Técnica Superior de Ingeniería Superior Grado en Ingeniería Agrícola Grado en Ingeniería Eléctrica Grado en Ingeniería Electrónica Industrial Grado en Ingeniería Energética Grado en Ingeniería en Explotación de Minas y Recursos Energéticos Grado en Ingeniería Forestal y del Medio Natural Grado en Ingeniería Informática Grado en Ingeniería Mecánica Grado en Ingeniería Química Industrial</p> <p>Facultad de Enfermería Grado en Enfermería</p> <p>Facultad de Ciencias del Trabajo Grado en Relaciones Laborales y Recursos Humanos</p> <p>Facultad de Trabajo Social Grado en Trabajo Social</p>
--

Publicación de estas titulaciones correspondientes a los Grados, como colecciones dentro del Repositorio Institucional Arias Montano (ver figura 39):

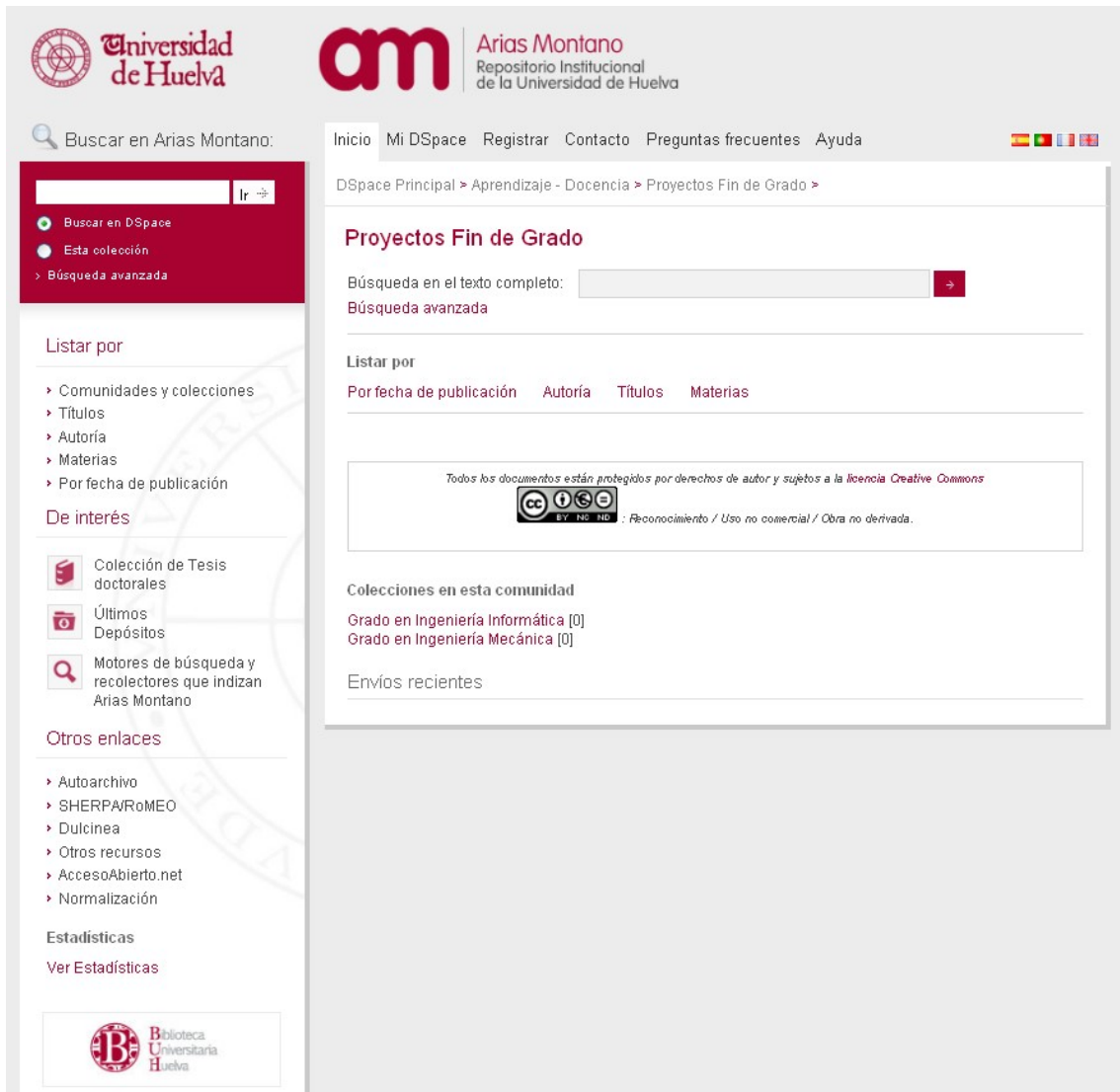


Figura 39

Anexo2: Formulario para los TFG:

/dspace/enovation/huelva-1173/dspace/config/input-forms.xml

```

<form name="tesis">
  <page number="1">
    <field>
      <dc-schema>dc</dc-schema>
      <dc-element>contributor</dc-element>
      <dc-qualifier>author</dc-qualifier>
      <repeatable>>true</repeatable>
      <label>Authors</label>
      <input-type>name</input-type>
      <hint>Enter the names of the authors.</hint>
      <required>You must enter at least one author</required>
    </field>
    <field>
      <dc-schema>dc</dc-schema>
      <dc-element>contributor</dc-element>
      <dc-qualifier>advisor</dc-qualifier>
      <repeatable>>true</repeatable>
  
```

```

<label>Advisor</label>
<input-type>name</input-type>
<hint>Enter the names of the advisors.</hint>
<required>You must enter at least one advisor</required>
</ field >
< field >
  <dc-schema>dc</dc-schema>
  <dc-element>contributor</dc-element>
  <dc-qualifier>other</dc-qualifier>
  <repeatable>>false</repeatable>
  <label>Department</label>
  <input-type>onebox</input-type>
  <hint>Enter: Universidad de Huelva. Department.</hint>
  <required>You must enter: UHU. Department.</required>
</ field >
< field >
  <dc-schema>dc</dc-schema>
  <dc-element>title</dc-element>
  <dc-qualifier></dc-qualifier>
  <repeatable>>false</repeatable>
  <label>Title</label>
  <input-type>onebox</input-type>
  <hint>Enter the main title of the item.</hint>
  <required>You must enter a main title ...</required>
</ field >
< field >
  <dc-schema>dc</dc-schema>
  <dc-element>title</dc-element>
  <dc-qualifier>alternative</dc-qualifier>
  <repeatable>>true</repeatable>
  <label>Other Titles</label>
  <input-type>onebox</input-type>
  <hint>If the item has any alternative titles, ...</hint>
  <required></required>
</ field >
< field >
  <dc-schema>dc</dc-schema>
  <dc-element>date</dc-element>
  <dc-qualifier>issued</dc-qualifier>
  <repeatable>>false</repeatable>
  <label>Date of Issue</label>
  <input-type>date</input-type>
  <hint>Please give the date of previous publication ... </hint>
  <required>You must enter at least the year.</required>
</ field >
< field >
  <dc-schema>dc</dc-schema>
  <dc-element>date</dc-element>
  <dc-qualifier>created</dc-qualifier>
  <repeatable>>false</repeatable>
  <label>Date of creation</label>
  <input-type>date</input-type>
  <hint>Enter the date of creation</hint>
  <required>You must enter at least the year.</required>
</ field >
< field >
  <dc-schema>dc</dc-schema>
  <dc-element>publisher</dc-element>
  <dc-qualifier></dc-qualifier>
  <repeatable>>false</repeatable>

```

```

    <label>Publisher</label>
    <input-type>onebox</input-type>
    <hint>Enter the name of the publisher of the ...</hint>
    <required></required>
  </ field >
  < field >
    <dc-schema>dc</dc-schema>
    <dc-element>identifier</dc-element>
    <dc-qualifier>citation</dc-qualifier>
    <repeatable>>false</repeatable>
    <label>Citation</label>
    <input-type>onebox</input-type>
    <hint>Enter the standard citation for the ...</hint>
    <required></required>
  </field>
  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>relation</dc-element>
    <dc-qualifier>ispartofseries</dc-qualifier>
    <repeatable>>true</repeatable>
    <label>Series/Report No.</label>
    <input-type>series</input-type>
    <hint>Enter the series and number assigned to ..</hint>
    <required></required>
  </field>
  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>identifier</dc-element>
    <dc-qualifier></dc-qualifier>
    <!-- An input-type of qualdrop_value MUST be marked ... -->
    <repeatable>>true</repeatable>
    <label>Identifiers</label>
    <input-type value-pairs-name="common_identifiers">
      qualdrop_value</input-type>
    <hint>If the item has any identification numbers ..</hint>
    <required></required>
  </field>
  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>type</dc-element>
    <dc-qualifier></dc-qualifier>
    <repeatable>>true</repeatable>
    <label>Type</label>
    <input-type value-pairs-name="common_types">dropdown
    </input-type>
    <hint> Select the type(s) of content of the item...</hint>
    <required></required>
  </field>
  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>language</dc-element>
    <dc-qualifier>iso</dc-qualifier>
    <repeatable>>false</repeatable>
    <label>Language</label>
    <input-type value-pairs-name="common_iso_languages">
      dropdown</input-type>
    <hint>Select the language of the main content of ..!</hint>
    <required></required>
  </field>
  <field>

```

```

    <dc-schema>dc</dc-schema>
    <dc-element>description</dc-element>
    <dc-qualifier>abstract</dc-qualifier>
    <repeatable>>false</repeatable>
    <label>Abstract</label>
    <input-type>textarea</input-type>
    <hint> Enter the abstract of the item below. </hint>
    <required></required>
  </field>
  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>rights</dc-element>
    <dc-qualifier></dc-qualifier>
    <repeatable>>false</repeatable>
    <label>Rights</label>
    <input-type>textarea</input-type>
    <hint>Enter right grants on the item</hint>
    <required></required>
  </field>
  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>subject</dc-element>
    <dc-qualifier></dc-qualifier>
    <!-- An input-type of twobox MUST be marked as repeatable -->
    <repeatable>>true</repeatable>
    <label>Subject Keywords</label>
    <input-type>twobox</input-type>
    <hint> Enter appropriate subject keywords or ... </hint>
    <required></required>
    <vocabulary>srsc</vocabulary>
  </field>
  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>description</dc-element>
    <dc-qualifier>sponsorship</dc-qualifier>
    <repeatable>>false</repeatable>
    <label>Sponsors</label>
    <input-type>textarea</input-type>
    <hint> Enter the names of any sponsors and/or ... </hint>
    <required></required>
  </field>
  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>description</dc-element>
    <dc-qualifier></dc-qualifier>
    <repeatable>>false</repeatable>
    <label>Description</label>
    <input-type>textarea</input-type>
    <hint> Enter any other description or comments ... </hint>
    <required></required>
  </field>
</page>
</form>

```

Anexo 3: Conexión del Formulario con las Colecciones:

/dspace/enovation/huelva-1173/dspace/config/input-forms.xml

```
<form-map>
```

```
<name-map collection-handle="default" form-name="traditional" />

<!-- collection-handle poner n handle de la colección tesis -->
<name-map collection-handle="10272/3" form-name="tesis" />

<!-- collection-handle poner n handle de la colección OpenAire -->
<name-map collection-handle="10272/5372" form-name="OpenAIRE" />

<!-- collection-handle poner n handle de la colección TFG -->
<name-map collection-handle="10272/5615" form-name="TFG " />
</form-map>
```

Anexo 4: Definir los datos del formulario TFG que deben visualizarse:
/dspace/enovation/huelva-1173/dspace/config/dspace.cfg

```
webui.itemdisplay.TFG = dc.title, dc.title.alternative, dc.contributor.author, \
dc.contributor.advisor, dc.contributor.other, \
dc.subject.other, dc.date.submitted(date), \
dc.type, \
dc.description.abstract, \
dc.identifier.uri(link), \
dc.rights
webui.itemdisplay.TFG.collections = 10272/5615
```

Anexo 5: Crear un grupo de usuarios por cada colección creada (ver figura 40):

Universidad de Huelva | am | Arias Montano | Repositorio Institucional de la Universidad de Huelva | Salir

Inicio | Perfil:Montserrat | Envíos / Depósitos | Contacto | Preguntas frecuentes | Ayuda

DSpace Principal > Administrar grupos > Editar grupo >

Editor de grupos: Grupo_Grado_Ing_Informatica (id: 419)

Cambiar el nombre del grupo:

Buscar miembros para añadir:

Miembros

ID	Nombre	Correo electrónico	Ninguno
2	Teresa Contreras Barranco	teresa.contreras@biblio.uhu.es	<input type="button" value="Borrar"/>
179	Susana Moreno Garrido	susana.moreno@biblio.uhu.es	<input type="button" value="Borrar"/>
231	ntribunal1_tribunal1	ntribunal1.tribunal1@biblio.uhu.es	<input type="button" value="Borrar"/>
232	ntribunal2_tribunal2	ntribunal2.tribunal2@biblio.uhu.es	<input type="button" value="Borrar"/>
233	ntribunal3_tribunal3	ntribunal3.tribunal3@biblio.uhu.es	<input type="button" value="Borrar"/>
234	ntribunal4_tribunal4	ntribunal4.tribunal4@biblio.uhu.es	<input type="button" value="Borrar"/>
235	ntribunal5_tribunal5	ntribunal5.tribunal5@biblio.uhu.es	<input type="button" value="Borrar"/>
236	ndirector1_adirector1	ndirector1.adirector1@biblio.uhu.es	<input type="button" value="Borrar"/>
237	ndirector2_adirector2	ndirector2.adirector2@biblio.uhu.es	<input type="button" value="Borrar"/>

Figura 40

Dar permisos a ese grupo como administrador de la colección (ver figura 41):



Figura 41

Recordemos que al crear la colección, Dspace crea automáticamente un grupo de administradores para ella. Nosotros solo debemos asociar el grupo creado en el paso anterior, al grupo de administradores creado para esa colección

Anexo 6: Conexión al LDAP de la Universidad de Huelva

LDAP.java

```

package seguridad;

import java.util.Hashtable;
import java.util.StringTokenizer;

import javax.naming.NamingEnumeration;
import javax.naming.directory.Attribute;
import javax.naming.directory.Attributes;
import javax.naming.directory.BasicAttribute;
import javax.naming.directory.BasicAttributes;
import javax.naming.directory.DirContext;
import javax.naming.directory.InitialDirContext;
import javax.naming.directory.SearchControls;
import javax.naming.directory.SearchResult;

import uhu.buh.dspace.TFG.UsuarioWeb;

public class LDAP {
    //Estos son los datos que me interesa extraer del LDAP de la UHU, del usuario que se identifica
    String _ldap_email_field = null; //email
    String _ldap_nombre_field = null; //nombre
    String _ldap_apellido1_field = null; //apellido1
    String _ldap_apellido2_field = null; //apellido2
    String _ldap_uhuUserDni_field = null; //dni
    String _ldap_uid_field = null; //usuario de correo
    String _ldap_uhuOraDn_field = null;
    
```

```

//uid=montserrat.soler.sic,ou=sic,ou=personal,dc=uhu,dc=es

String _ldap_object_context    = null;
String _ldap_search_context    = null;
String _ldap_provider_url      = null;

    public LDAP() {
        ;
    }

    public LDAP(String ldap_email_field, String ldap_nombre_field, String ldap_apellido1_field,
String ldap_apellido2_field, String ldap_uhuUserDni_field, String ldap_uid_field, String
ldap_uhuOraDn_field, String ldap_object_context, String ldap_search_context, String
ldap_provider_url) {

        setLdap_email_field(ldap_email_field);
        setLdap_nombre_field(ldap_nombre_field);
        setLdap_apellido1_field(ldap_apellido1_field);
        setLdap_apellido2_field(ldap_apellido2_field);
        setLdap_uhuUserDni_field(ldap_uhuUserDni_field);
        setLdap_uid_field(ldap_uid_field);
        setLdap_uhuOraDn_field(ldap_uhuOraDn_field);
        setLdap_object_context(ldap_object_context);
        setLdap_search_context(ldap_search_context);
        setLdap_provider_url(ldap_provider_url);
    }

    public UsuarioWeb ldapAuthenticate(String uid, String password) {
        UsuarioWeb usuarioWeb    = null;
        Hashtable env            = null;
        DirContext ctx           = null;
        Attributes matchAttrs    = null;
        Attributes atts          = null;
        Attribute att            = null;
        SearchControls ctls      = null;
        SearchResult sr          = null;
        NamingEnumeration answer = null;
        String uhuOraDn          = null;

        try {
            // Set up environment for creating initial context
            env = new Hashtable(11);
            env.put(javax.naming.Context.INITIAL_CONTEXT_FACTORY,
"com.sun.jndi.ldap.LdapCtxFactory");
            env.put(javax.naming.Context.PROVIDER_URL, getLdap_provider_url());

            // Create initial context
            ctx = new InitialDirContext(env);

            matchAttrs = new BasicAttributes(true);
            matchAttrs.put(new BasicAttribute(getLdap_uid_field(), uid));

            String attlist[] = {getLdap_email_field(), getLdap_nombre_field(),
getLdap_apellido1_field(), getLdap_apellido2_field(), getLdap_uhuUserDni_field(),
getLdap_uid_field(),getLdap_uhuOraDn_field()};
            ctls = new SearchControls();
            ctls.setSearchScope(SearchControls.SUBTREE_SCOPE);

            // look up attributes
            answer = ctx.search(getLdap_search_context(),("&(" + getLdap_uid_field() + "=" + uid + ")"),

```

```

ctls);

while(answer.hasMore()) {
    sr = (SearchResult)answer.next();
    atts = sr.getAttributes();
    ctx.close();

    // Authenticate
    env.put(javax.naming.Context.SECURITY_AUTHENTICATION, "simple");
    env.put(javax.naming.Context.SECURITY_PRINCIPAL, atts.get("uhuOraDn").get());
    env.put(javax.naming.Context.SECURITY_CREDENTIALS, password);
    ctx = new InitialDirContext(env);
    //si llegamos a este punto nos hemos autenticado correctamente vs el LDAP, sino habra una
    excepcion
    usuarioWeb = new UsuarioWeb();

    if (attlist[0] != null) {
        att = atts.get(attlist[0]);
        if (att != null) usuarioWeb.setEmail((String)att.get());
    }
    if (attlist[1] != null) {
        att = atts.get(attlist[1]);
        if (att != null) usuarioWeb.setNombre((String)att.get());
    }
    if (attlist[2] != null) {
        att = atts.get(attlist[2]);
        if (att != null) usuarioWeb.setApellido1((String)att.get());
    }
    if (attlist[3] != null) {
        att = atts.get(attlist[3]);
        if (att != null) usuarioWeb.setApellido2((String)att.get());
    }
    if (attlist[4] != null) {
        att = atts.get(attlist[4]);
        if (att != null) usuarioWeb.setUhuUserDni((String)att.get());
    }
    if (attlist[5] != null) {
        att = atts.get(attlist[5]);
        if (att != null) usuarioWeb.setUid((String)att.get());
    }
    }

    // uid=santiago.robles,ou=alumnos,dc=uhu,dc=es
    if (attlist[6] != null) {
        att = atts.get(attlist[6]);
        if (att != null){
            uhuOraDn = (String)att.get();
            StringTokenizer st = new StringTokenizer(uhuOraDn, ",");
            while (st.hasMoreTokens()){
                String token=st.nextToken();
                if(token.substring(0, token.indexOf("=")).equals("ou")){
                    if
                    (token.substring(token.indexOf("=")+1).equals(usuarioWeb.TIPO_ALUMNOS)){
                        usuarioWeb.setTipo(usuarioWeb.TIPO_ALUMNOS);
                    }
                    else if (token.substring(token.indexOf("=")+1).equals(usuarioWeb.TIPO_PERSONAL)){
                        usuarioWeb.setTipo(usuarioWeb.TIPO_PERSONAL);
                    }
                }
            }
        }
    }
}

```

```

    }
}
    // Close the context when we're done
ctx.close();
    } catch (Exception e) {
        // si se produce un error, se indica devolviendo null
        usuarioWeb = null;
    }

    return(usuarioWeb);
}

public String getLdap_email_field() {return( _ldap_email_field);}
public String getLdap_nombre_field() {return( _ldap_nombre_field);}
public String getLdap_apellido1_field() {return( _ldap_apellido1_field);}
public String getLdap_apellido2_field() {return( _ldap_apellido2_field);}
public String getLdap_uhuUserDni_field() {return( _ldap_uhuUserDni_field);}
public String getLdap_uid_field() {return( _ldap_uid_field);}
public String getLdap_uhuOraDn_field() {return( _ldap_uhuOraDn_field);}
public String getLdap_object_context() {return( _ldap_object_context);}
public String getLdap_search_context() {return( _ldap_search_context);}
public String getLdap_provider_url() {return( _ldap_provider_url);}

public void setLdap_email_field(String ldap_email_field) { _ldap_email_field =
ldap_email_field;}
public void setLdap_nombre_field(String ldap_nombre_field) { _ldap_nombre_field =
ldap_nombre_field;}
public void setLdap_apellido1_field(String ldap_apellido1_field) { _ldap_apellido1_field =
ldap_apellido1_field;}
public void setLdap_apellido2_field(String ldap_apellido2_field) { _ldap_apellido2_field =
ldap_apellido2_field;}
public void setLdap_uhuUserDni_field(String ldap_uhuUserDni_field) {
_ldap_uhuUserDni_field = ldap_uhuUserDni_field;}
public void setLdap_uid_field(String ldap_uid_field) { _ldap_uid_field = ldap_uid_field;}
public void setLdap_uhuOraDn_field(String ldap_uhuOraDn_field) { _ldap_uhuOraDn_field =
ldap_uhuOraDn_field;}
public void setLdap_object_context(String ldap_object_context) { _ldap_object_context =
ldap_object_context;}
public void setLdap_search_context(String ldap_search_context) { _ldap_search_context =
ldap_search_context;}
public void setLdap_provider_url(String ldap_provider_url) { _ldap_provider_url =
ldap_provider_url;}
}
}

```

Anexo 7: Crear conexión a BD

DspaceBuh.java

```

private Connection crearConexionDB(String host, String port, String login, String passwd)
{
    String url          = null;
    String driver       = null;
    Properties props    = null;

    Connection conexionDB = null;
}

```

```

url = "jdbc:postgresql://" + host + (port != null && port.trim().length() > 0?":" + port:"") +
"/dspace";

driver = "org.postgresql.Driver";

try {
    props = new Properties();
    Class.forName(driver);

    props.setProperty("user", login);
    props.setProperty("password", passwd);

    conexionDB = DriverManager.getConnection(url, props);
} catch ( ClassNotFoundException e) {
    toLog(e.getMessage());
    conexionDB = null;
    System.out.println(e.getMessage());
} catch (SQLException e) {
    toLog(e.getMessage());
    conexionDB = null;
    System.out.println(e.getMessage());
} catch (Exception e) {
    toLog(e.getMessage());
    conexionDB = null;
    System.out.println(e.getMessage());
}

return(conexionDB);
}

```

Anexo 8: Construir `dublin_core.xml` y ubicarlo en `/dspace/FichImportar/idproyecto/item`

DspaceBuh.java

```

// Este método o función se utiliza para generar el fichero xml,
// una vez confirmados los datos por el director.
// El fichero será subido a /dspace/FichImportar/idproyecto

public String generarXML(int idProyecto) {
    String devolver = "";
    PreparedStatement pStmt = null;
    ResultSet resultado = null;
    String sql = null;

    StringTokenizer st = null;

    String palabras_claves_spi = null;
    String palabras_claves_eng = null;
    String separator = null;

    File rutaProyectoAlumno = null;
    File dublin_core = null;
    FileWriter fw = null;
    PrintWriter pw = null;

    try {
        separator = System.getProperty("file.separator");
    }
}

```

```

        if(getConexionDB() != null) {
            sql = "SELECT nombre_alumno, apellido1_alumno, apellido2_alumno, " +
"firstname AS nombreDirector, lastname AS apellidosDirector, director_id, " +
"registerproyect.collection_id, collection.name AS nombreColeccion, " +
"registerproyect.community_id, community.name AS nombreComunidad, " +
"fechaLectura, resumenSpi, resumenEng, palabras_claves_spi, palabras_claves_eng, " +
"titulo, idioma, fichero1, fichero2 " +
"FROM ((registerproyect " +
"INNER JOIN eperson ON registerproyect.director_id = eperson.eperson_id) " +
"INNER JOIN collection ON registerproyect.collection_id = collection.collection_id) " +
"INNER JOIN community ON registerproyect.community_id = community.community_id " +
"WHERE register_proyect_id = ?;";

            pStmt = getConexionDB().prepareStatement(sql);
            pStmt.setInt(1, idProyecto);
            resultado = pStmt.executeQuery();

            if(resultado.next()){
                // Ahora se guarda el fichero xml generado en /dspace/FichImportar/proyecto_id
                rutaProyectoAlumno = new File(this.directorioProyectosAlumnos +
String.valueOf(idProyecto)); //esta instrucción abre la ruta

                if (!rutaProyectoAlumno.exists()) {
                    rutaProyectoAlumno.mkdir();
                }

                dublin_core = new File(rutaProyectoAlumno + separator + "dublin_core.xml");
                fw = new FileWriter(dublin_core);
                pw = new PrintWriter(fw);

                pw.println("<dublin_core>");
                pw.println("  <dcvalue element=\"contributor\" qualifier=\"advisor\">" +
resultado.getString("apellido1_alumno").trim() + " " + resultado.getString("apellido2_alumno").trim() +
", " + resultado.getString("nombre_alumno").trim() + "</dcvalue>");
                pw.println("  <dcvalue element=\"contributor\" qualifier=\"author\">" +
resultado.getString("apellidosDirector").trim() + ", " + resultado.getString("nombreDirector").trim() +
"</dcvalue>");
                pw.println("  <dcvalue element=\"contributor\" qualifier=\"other\">UNIVERSIDAD
DE HUELVA." + resultado.getString("nombreColeccion") + "</dcvalue>");
                pw.println("  <dcvalue element=\"date\" qualifier=\"created\">" +
resultado.getString("fechaLectura") + "</dcvalue>");
                pw.println("  <dcvalue element=\"description\" qualifier=\"abstract\">" +
resultado.getString("resumenSpi") + "</dcvalue>");
                pw.println("  <dcvalue element=\"description\" qualifier=\"abstract\">" +
resultado.getString("resumenEng") + "</dcvalue>");
                pw.println("  <dcvalue element=\"language\" qualifier=\"iso\">" +
resultado.getString("idioma") + "</dcvalue>");

                if(palabras_claves_spi != null) {
                    st = new StringTokenizer(palabras_claves_spi, ",");
                    while(st.hasMoreTokens()) {
                        pw.println("  <dcvalue element=\"subject\" qualifier=\"other\">" +
st.nextToken() + "</dcvalue>");
                    }
                }

                if(palabras_claves_eng != null) {
                    st = new StringTokenizer(palabras_claves_eng, ",");
                    while(st.hasMoreTokens()) {

```

```

        pw.println("    <dcvalue element=\"subject\" qualifier=\"other\">" +
st.nextToken() + "</dcvalue>");
    }
    }
    pw.println("    <dcvalue element=\"title\" qualifier=\"none\">" +
resultado.getString("titulo") + "</dcvalue>");
    pw.println("    <dcvalue element=\"type\" qualifier=\"other\">" +
resultado.getString("nombreComunidad") + "</dcvalue>");
    pw.println("</dublin_core>");

    pw.close();
    fw.close();
    }
    }
} catch (Exception e) {
    devolver = e.getMessage();
}

return(devolver);
}

```

Anexo 9: Subir ítem al repositorio.

DspaceBuh.java

```

//Este método o función permite subir el proyecto al repositorio
//y devolver la url Handle
public String subirRepositorioItem(int idProyecto, int idColeccion, String importarSiNo) {
    String devolver = "";
    String urlHandle = "";

    File uriMapfile = null;

    String s = null;
    Process p = null;
    BufferedReader stdInput = null;
    BufferedReader stdError = null;
    String comando = null;
    String linea = null;

    try {
        if (importarSiNo.equals("Si")){

            comando = "sudo /dspace/bin/import -a -e montserrat.soler@sic.uhu.es -c 10272/" +
String.valueOf(idColeccion) + " -s /dspace/FichImportar/" +
String.valueOf(idProyecto) + " -m /dspace/FichImportar/" +
String.valueOf(idProyecto) + "/mapfile";

            p = Runtime.getRuntime().exec(comando);

            stdInput = new BufferedReader(new InputStreamReader(p.getInputStream()));

            stdError = new BufferedReader(new InputStreamReader(p.getErrorStream()));

            while ((s = stdInput.readLine()) != null) {
                ;
            }

            // read any errors from the attempted command

```

```

        while ((s = stdError.readLine()) != null) {
            ;
        }

        //recojo la URI del fichero mapfile
        uriMapfile = new File("/dspace/FichImportar/" + String.valueOf(idProyecto) +
            "/mapfile");

        stdInput = new BufferedReader(new FileReader(uriMapfile));
        linea = stdInput.readLine();
        linea = linea.split("/")[1];

        //devolver = "http://hdl.handle.net/10272/5620";
        devolver = "http://hdl.handle.net/10272/" + linea;

        //Oculta el item subido

    }
    else {
        devolver = "http://hdl.handle.net/10272/5620 prueba";
    }
} catch (Exception e) {
    toLog(e.getMessage());
    devolver = e.getMessage();
}

return(devolver);
} // fin del script

<body>
<h2>Lanzamiento sentencia IMPORT en linux desde Java</h2>
<%
try {
    String s = null;
    Process p = null;
    BufferedReader stdInput = null;
    BufferedReader stdError = null;

    p = Runtime.getRuntime().exec(
        "sudo /dspace/bin/import -r -e montserrat.soler@sic.uhu.es
        -c 10272/5616 -s /dspace/FichImportar/item1
        -m /dspace/FichImportar/item1/mapfile");

    stdInput = new BufferedReader(new InputStreamReader(p.getInputStream()));
    stdError = new BufferedReader(new InputStreamReader(p.getErrorStream()));

    out.println("<p>Salida devuelta por el comando o sentencia:</p>");

    while ((s = stdInput.readLine()) != null) {
        out.println(s + "<br>");
    }

    // read any errors from the attempted command
    System.out.println("<p>Salida devuelta por el error del comando o sentencia (si existiera):</p>");

    while ((s = stdError.readLine()) != null) {
        out.println(s + "<br>");
    }
} catch (Exception e) {
    out.println("<h3>Se ha producido el error:<br>" + e.getMessage() + "</h3>");
}

%>
</body>

```

