

Instituto Tecnológico de Costa Rica
Administración de Tecnologías de Información
Lenguajes de Programación
Semestre II, 2015
Profesor: Ing. Isaac Alpízar Chacón, M.Sc.

Tarea Programada #1

Repositorio Institucional en C (RIC)

1. Motivación

La divulgación de la producción científica y académica generada en las universidades y otros centros de investigación es de suma importancia para compartir el conocimiento generado. Actualmente, el uso de repositorios institucionales para la gestión y el acceso a dicho contenido, es una buena forma para que se pueda acceder a todo el contenido de forma abierta. “Un repositorio institucional (RI) es un conjunto de servicios para almacenar y hacer accesibles materiales de investigación en formato digital creados por una institución y su comunidad (...). Los Repositorios Institucionales se están convirtiendo en herramientas esenciales para la comunicación académica en la era digital.”¹

El TEC cuenta con el Repositorio TEC² para reunir, conservar y difundir a través del acceso abierto a los documentos en formato digital resultantes de la actividad académica y científica del TEC.

El propósito de esta tarea programada es desarrollar en el lenguaje de programación C, un sistema simplificado para almacenar, acceder y compartir documentos, con el objetivo de mostrar las principales funcionalidades de un Repositorio Institucional.

2. Contexto y las funcionalidades esperadas

Se deberá crear un RI en C que permita agregar, consultar, buscar y exportar documentos. El programa se utilizará a través de la consola, no se implementará interfaz gráfica.

El programa deberá desplegar un menú al usuario con las siguientes funciones:

1 Agregar documento

Esta función es la encargada de agregar un nuevo documento al RIC. El programa debe pedirle al usuario la dirección en el sistema donde se encuentra el archivo (por ejemplo: `/home/user/documents/tesis1.pdf`); este se debe copiar en el directorio interno del repositorio; y luego se solicita al usuario una serie de metadatos que describen el documento.

¹ http://wikiaa.unr.edu.ar/index.php/Repositorios_Institucionales

² <http://repositoriotec.tec.ac.cr/>

Los metadatos que se deben pedir al usuario son los definidos en el *Dublin Core* versión 1.1³. Se debe pedir al usuario que ingrese el valor para cada uno de los metadatos, tomando en cuenta:

- a. Los siguientes campos son obligatorios (siempre se debe introducir un valor): *title*, *creator* (autor), *subject* (palabra clave), *description*, *date*, *type*, *format*, *identifier*, *language*.
- b. Los siguientes campos son opcionales (el usuario puede dejarlos en blanco): *publisher*, *contributor*, *source*, *relation*, *coverage*, *rights*.
- c. Los siguientes campos deben generarse automáticamente, y el valor se presenta al usuario, el cual puede aceptarlo o cambiarlo: *format* (tipo de documento, seguido del tamaño del archivo en bytes, ejemplo: "pdf/800bytes"), *language* (por defecto es *SPA*), *identifier* (debe ser un identificador único generado por el sistema).
- d. Los siguientes campos pueden tener múltiples instancias (se puede repetir el campo, pero con diferente valor): *title*, *creator*, *subject*.
- e. El campo *type* solo puede ser uno de los siguientes valores (debe mostrarse al usuario una forma fácil de seleccionar el deseado): info:eu-repo/semantics/bachelorThesis, info:eu-repo/semantics/masterThesis, info:eu-repo/semantics/doctoralThesis, info:eu-repo/semantics/article, info:eu-repo/semantics/report, info:eu-repo/semantics/book, info:eu-repo/semantics/bookPart, info:eu-repo/semantics/review, info:eu-repo/semantics/conferenceObject, info:eu-repo/semantics/lectura, info:eu-repo/semantics/workingPaper, info:eu-repo/semantics/preprint, info:eu-repo/semantics/contributionToPeriodical, info:eu-repo/semantics/patent, info:eu-repo/semantics/other.
- f. El campo *language* debe seguir el ISO 639-3⁴.
- g. El campo *date* debe seguir el formato YYYY, YYYY-MM, YYYY-MM-DD.

2 Consultar documento

Se solicita al usuario el identificador del documento (campo *identifier*) y se despliega en la pantalla toda la información relacionada al documento: todos sus metadatos, y el nombre del archivo.

3 Buscar documentos

El usuario debe seleccionar un criterio de búsqueda: título, palabra clave o autor. Luego ingresa el texto de búsqueda, y se debe desplegar un listado con todos los documentos que cumplen el criterio de búsqueda. La lista resultante debe incluir un consecutivo (empezando en uno) y el título del documento. Es decir, si hay 5 resultados, se mostrará algo como:

1. Título 1
2. Título 2
3. Título 3

³ <http://dublincore.org/documents/dces/>, https://en.wikipedia.org/wiki/Dublin_Core

⁴ http://www.iso.org/iso/home/standards/language_codes.htm , <http://www-01.sil.org/iso639-3/codes.asp>

4. Título 4
5. Título 5

Y luego el usuario podrá seleccionar el número de resultado, y se desplegará toda la información asociada al documento (como en la funcionalidad 2).

Nota: Si se selecciona el criterio de búsqueda título, y el usuario ingresa la palabra “biología”, se deben recuperar todos los documentos en donde la palabra “biología” forma parte del título, no tiene que ser una coincidencia exacta, de igual manera para el autor o para una palabra clave.

4 Exportar metadatos

Se solicita al usuario el identificador del documento (campo *identifier*) y se debe generar un archivo XML con todos los metadatos asociados. El archivo XML debe seguir el formato oai_dc⁵. Solo se deben incluir los metadatos que tengan un valor, por ejemplo, si uno de los campos opcionales no se introdujo a la hora de ingresar el documento, entonces dicho campo no debe aparecer en el documento XML.

Ejemplo de documento XML⁶.

```
<oai_dc:dc
xsi:schemaLocation
="
http://www.openarchives.org/OAI/2.0/oai_dc/ http://www.openarchives.org/OAI/2.0/oai_dc.xsd
">
<dc:title>
    Diagnóstico de los sistemas de producción del cultivo de palmito (Bactris gasipaes H.B.K.) en
    el cantón de Sarapiquí, Costa Rica 2005 - 2006
</dc:title>
<dc:creator>
    Montoya Campos, Jorge Alberto
</dc:creator>
<dc:subject>
    Costos
</dc:subject>
```

⁵ https://www.openarchives.org/OAI/2.0/oai_dc.xsd

⁶

http://repositoriotec.tec.ac.cr/oai/request?verb=GetRecord&metadataPrefix=oai_dc&identifier=oai:repositoriotec.tec.ac.cr:2238/5908,
http://repositoriotec.tec.ac.cr/oai/request?verb=ListRecords&metadataPrefix=oai_dc

<dc:subject>

TFSC

</dc:subject>

<dc:subject>

Manejo postcosecha

</dc:subject>

<dc:subject>

Cosecha

</dc:subject>

<dc:subject>

Enfermedades de la planta

</dc:subject>

<dc:subject>

Control de malezas

</dc:subject>

<dc:subject>

Fertilizacion

</dc:subject>

<dc:subject>

Suelo

</dc:subject>

<dc:subject>

Drenaje

</dc:subject>

<dc:subject>

Topografia

</dc:subject>

<dc:subject>

Produccion

</dc:subject>

<dc:subject>

Cultivo

</dc:subject>

<dc:subject>

Palmito (Bactris gasipaes)

</dc:subject>

<dc:subject>

Sistemas de Posicionamiento Global

</dc:subject>

<dc:subject>

Diagnóstico de sistema de producción

</dc:subject>

<dc:description>

Tesis (Licenciatura en Ingeniería en Agronomía) Instituto Tecnológico de Costa Rica, Escuela de Ingeniería en Agronomía, 2007

</dc:description>

<dc:description>

Los procesos de globalización a nivel mundial obligan a los diversos países a ser cada día más eficientes y eficaces en su accionar con el fin de elevar los niveles de competitividad y sostenibilidad de sus diversas actividades y en especial en el campo agropecuario (...).

</dc:description>

<dc:description>

Centro de Cooperación Internacional de Investigación Agronómica para el Desarrollo (CIRAD) Instituto Tecnológico de Costa Rica Sede San Carlos. Escuela de Ingeniería en Agronomía.

</dc:description>

<dc:date>

2015-05-21T22:28:17Z

</dc:date>

<dc:date>

2015-05-21T22:28:17Z

</dc:date>

<dc:date>

2007

</dc:date>

<dc:type>

Thesis

</dc:type>

```
<dc:identifier>
    http://hdl.handle.net/2238/5908
</dc:identifier>
<dc:language>
    es
</dc:language>
<dc:publisher>
    Instituto Tecnológico de Costa Rica
</dc:publisher>
</oai_dc:dc>
```

Nota: El documento generado debe quedar en la carpeta */tmp* y se debe usar el identificador del documento como nombre del archivo XML.

5 Recuperar archivos

Esta funcionalidad permite recuperar el archivo asociado a uno o más documentos. Aquí hay dos opciones:

- El usuario ingresa el identificador del documento (campo *identifier*) del cual quiere recuperar el archivo asociado, y este se copia en la carpeta */tmp*.
- El usuario ingresa varios identificadores, y se debe generar un único nuevo archivo *pdf* que contiene todos los archivos “pagados”, y se debe copiar en la carpeta */tmp*.

Nota: Para la generación de un *pdf* que contiene varios archivos, se puede utilizar alguna librería externa de C, o de Linux. En cualquier caso, se debe documentar los pasos necesarios para la instalación de dicha la librería.

6 Salir

Finaliza el programa. Al seleccionar esta opción, se debe asegurar que toda la información del programa se almacena de forma persistente, es decir, que al volver a correr el programa, todos los datos y archivos se encuentran disponibles.

Otros aspectos

- a. El programa deberá crear su propio espacio físico (i.e. una carpeta) donde se deberá guardar una copia de todos los documentos que se ingresen en el RIC.
- b. Cada grupo deberá definir como guardar de forma persistente la información (por ejemplo, en un solo archivo o en varios).
- c. El sistema debe aceptar archivos con extensión *.txt* y *.pdf*.
- d. El programa debe almacenar toda la información de un documento como un **struct**.

- e. Para los metadatos que se pueden repetir, por ejemplo *subject*, puede haber una ocurrencia máxima de 5 instancias.

Puntos Extra

Agregar el concepto de “Comunidades” y “Colecciones” representa 10 puntos extra. Se pueden crear comunidades, y dentro de las comunidades una o más colecciones. Cuando se ingresa un documento, se indica a cual colección pertenece, y cuando se muestra la información de un documento también se indica la colección. Debe existir una funcionalidad de desplegar todos los documentos para una colección específica. Los documentos solo se pueden agregar a colecciones, no a comunidades.

3. Aspectos técnicos

El proyecto deberá estar escrito en el lenguaje de programación C (no C++), y deberá de funcionar en el sistema operativo Linux. En caso de requerir librerías adicionales para compilar y ejecutar el programa, deberán especificarlo en la documentación, ya que de lo contrario se descontarán puntos en la evaluación. **Deben crear un MAKEFILE** que contenga todos los pasos necesarios para compilar el código y generar el ejecutable (investigar).

4. Foro de dudas

Se habilitará un foro de consultas sobre el proyecto en el TEC Digital. Por favor, plantee sus dudas o consultas por esa vía, no se resolverán dudas por correo electrónico.

5. Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de los mismos.

Para la documentación interna, deberán incluir comentarios descriptivos para cada función, con sus entradas, salidas, restricciones y objetivo.

La documentación externa deberá incluir:

1. Portada.
2. Manual de usuario: instrucciones de compilación, ejecución y uso.
3. Pruebas de funcionalidad: incluir *screenshots*.
4. Descripción del problema.
5. Diseño del programa: decisiones de diseño, algoritmos usados.
6. Librerías usadas: creación de pdf, etc.
7. Análisis de resultados: objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).

6. Evaluación

1. Documentación interna 3%.
2. Documentación externa 12%.
3. Ingresar documento 15%.
4. Consultar documento 10%.
5. Buscar documentos 15%.
6. Exportar metadatos 10%.
7. Recuperar archivos 10%.
8. Persistencia de datos 5%.
9. Revisión de tarea 10%.
10. Hora de Entrega 10%.

7. Aspectos administrativos

1. Debe crear un archivo .zip ("TP1.zip") que contenga únicamente un archivo **info.txt** y 2 carpetas llamadas **documentacion** y **solucion_computacional**, en la primera deberá incluir el documento de *word* o *open office* (no pdf) solicitado y en la segunda los archivos y/o carpetas necesarias para la implementación de esta tarea. El archivo **info.txt** debe contener la siguiente información (cualidades):
 - a. Nombre del curso
 - b. Número de semestre y año lectivo
 - c. Nombre del Estudiante x2 o x3
 - d. Número de carnet x2 o x3
 - e. Número de tarea programada
 - f. Fecha de entrega
 - g. Estatus de la entrega (definido por el responsable de la implementación de la tarea, debe ser **CONGRUENTE** con la solución entregada):
[Deplorable|Regular|Buena|MuyBuena|Excelente|Superior]
2. Deberá subir el archivo antes mencionado al TEC Digital en el curso de LENGUAJES DE PROGRAMACIÓN GR 01, en la asignación llamada "Tarea Programada I" debajo del rubro de "Tareas programadas". En la evaluación del Proyecto el rubro de "Hora de Entrega" valdrá por 10 puntos de la nota total del proyecto, según la siguiente escala:
 - a. Si se entrega antes de las 11:50 **PM** del viernes 04 de setiembre del 2015, 10 puntos.
 - b. Si se entrega antes de las 11:50 **AM** del sábado 05 de setiembre del 2015, 6.67 puntos.
 - c. Si se entrega antes de las 11:50 **AM** del domingo 06 de setiembre del 2015, 3.33 puntos.
 - d. Si se entrega antes de las 11:50 **PM** del domingo 06 de setiembre del 2015, 0 puntos. Después de este punto, **NO SE ACEPTARÁN** más trabajos.

3. Dentro de la carpeta de **solucion_computacional**, deberá incluir un archivo .txt (**PrimerNombreMiembro1.PrimerNombreMiembro2. PrimerNombreMiembro3.txt**) que contenga todo el texto de la solución del o de los archivos .c (**archivos de C**) presentados. Este archivo puede ser revisado en el sistema de Control de Plagio del TEC Digital. **Todo el código de cada proyecto debe ser 100% original, y no se va a tolerar el plagio.** Este archivo también deberá ser subido al TEC Digital en el curso de LENGUAJES DE PROGRAMACIÓN GR 01, en la asignación llamada “Tarea Programada I (archivo TXT)” debajo del rubro de “Tareas programadas”. Se deberá subir el archivo siguiendo los mismos rangos de tiempo que los descritos en el punto anterior.
4. Las tareas deberán ser revisadas con el profesor o el asistente. Todos los miembros del grupo deberán participar de la revisión, ya que de lo contrario no se les asignará el puntaje correspondiente. La nota de la revisión es individual, el resto de la nota es grupal.
5. La tarea se hará en grupos de 2 o 3 personas.
6. La tarea vale un 15% de la nota del curso.
7. Antes de la entrega del proyecto se pasará en clase una lista donde deben anotar los miembros de su equipo de trabajo. Opcionalmente, pueden mandar un correo electrónico con el asunto “[Lenguajes 215] TP1:GRUPO” e indicar los nombres de las personas que conforman el grupo de trabajo para esta tarea programada. Esta información es necesaria para crear los grupos en el TEC Digital. Si no se anotan en la lista, o no envían el correo, se calificará la tarea de forma individual al miembro del equipo que la suba.