UNIVERSIDAD DE COSTA RICA
ESCUELA DE CIENCIAS DE LA COMPUTACIÓN E INFORMÁTICA
CI-2414 RECUPERACIÓN DE INFORMACIÓN
PROF. KRYSCIA DAVIANA RAMÍREZ BENAVIDES

PROYECTO: SISTEMA DE RECUPERACIÓN DE INFORMACIÓN

OBJETIVOS

- Seleccionar URL's de sitios relevantes sobre diferentes necesidades de información específicas, para indexar y crear una colección de información en español.
- Crear un pequeño Sistema de Recuperación de Información en español para una colección de documentos variados.

FECHAS DE ENTREGA

- Entrega Etapa I: Viernes 27 de marzo.
- Entrega Etapa II: Viernes 24 de abril.
- Entrega Etapa III: Viernes 29 de mayo.
- Entrega Etapa IV: Del 30 de junio al 3 de julio.
- Entrega Final: Martes 7 de julio, 9am – 11am.

EVALUACIÓN

I Etapa	10%
II Etapa	10%
III Etapa	10%
IV Etapa	10%
Total	**40%**

NOTAS IMPORTANTES:

- El proyecto se puede realizar en grupo de cuatro personas como máximo.
- Se formarán los grupos el primer día de clases.
- Se realizará una evaluación del trabajo realizado a cada miembro del grupo (individualmente), para comprobar la participación de cada uno en cada etapa del proyecto. Se realizan el día de entrega de cada etapa. La nota obtenida en cada evaluación se promedia con la nota obtenida en la etapa respectiva del proyecto.
- Cuando los documentos se envían vía correo electrónico deben venir con el *subject* "**Proyecto Etapa # – Equipo #**" (por ejemplo: Proyecto Etapa I – Equipo 1), y los archivos que se adjuntan deben venir en un archivo comprimido con el nombre "**Etapa#_Equipo#.zip**" (por ejemplo: EtapaI_Equipo1.zip).
- Se realizará una evaluación del trabajo realizado a cada miembro del grupo (individualmente), para comprobar la participación de cada uno en cada etapa del proyecto. Se realizan el día de entrega de cada etapa. Estas evaluaciones deben enviarse por correo electrónico, el *subject* debe venir "**Auto-Coevaluación Etapa # – Carné**" (por ejemplo: Auto-Coevaluacion Etapa # - 993237) el archivo debe venir con el nombre "**Auto-Coevaluacion_Etapa#_Carne.ext**" (por ejemplo: Auto-Coevaluacion_Etapa1_993237.doc). La nota obtenida en cada evaluación se promedia con la nota obtenida en la etapa respectiva del proyecto.
- En cada etapa se debe entregar la documentación respectiva y la división del trabajo en la hora de la clase, esto será una prueba de la entrega del trabajo asignado.

ASPECTOS METODOLÓGICOS

Partiremos de la compresión de los estudiantes en los temas vistos en clase para asignar un proyecto, el cual está dividido en cuatro etapas, que serán desarrolladas por cada grupo de estudiantes. La realización del proyecto se hará en grupos de dos o tres personas.

Cada grupo desarrollará cada etapa, y al final del semestre culminará con un pequeño Sistema de Recuperación de Información y la presentación del mismo al profesor y al grupo.

SISTEMA DE RECUPERACIÓN DE INFORMACIÓN

Componentes básicos de un SRI (ver Figura #1):

- *Araña (Crawler):* Recorre la Web buscando las páginas a indexar.
- *Indexador:* Mantiene un índice con la información recolectada por el *crawler*.
- *Motor de Búsqueda:* Realiza las búsquedas en el índice.
- *Interfaz:* Interactúa con el usuario.

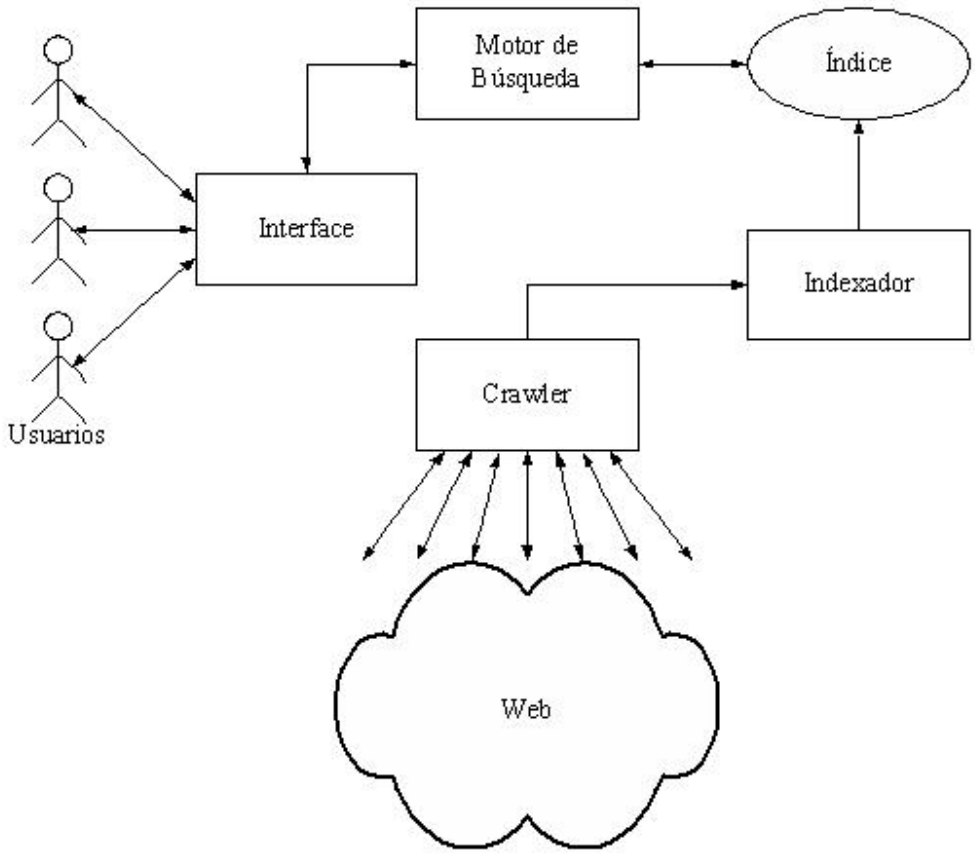

Figura #1: Componentes de un SRI

I ETAPA: CRAWLER

Fecha de Entrega: **27 / 03 / 2020**

Valor: **10%**

Enunciado:

Indique cinco tipos de necesidad de información que usted considere importante incluir en una colección en español para ser buscada, dé un título y una descripción de las necesidades de información seleccionadas.

Utilizando el SRI *Google*, busque y presente para cada tipo de necesidad de información, los 10 URL's de sitios devueltos por el SRI que usted considere relevantes.

Producir un archivo llamado "URLS" con los 50 URL's considerados relevantes, que contenga para cada URL el alias, el URL y la ubicación local del archivo (por ejemplo: 1.html www.nacion.com_amanda_amanda20.html .../Coleccion/1.html). El formato de salida propuesto es:

- En el archivo "URLS" cada línea define un alias con su respectivo URL y la ubicación local del archivo URL. Se recomienda que el ancho de la columna sea fijo y corresponda a (puede hacer las modificaciones justificadas que considere conveniente):
  - 15 caracteres para el alias.
  - 1 espacio en blanco.
  - 256 espacios para el URL.
  - 1 espacio en blanco.
  - 256 espacios para la ubicación local del archivo URL.
  - Cambio de línea.

Instrucciones:

- ✓ Presente un documento con los URL's separados por necesidad de información; el título, la descripción y la justificación de cada necesidad de información seleccionada, la división de trabajo, incluya portada (ver ejemplo).
- ✓ En un documento de EXCEL anexe los URL's, coloque en una hoja independiente cada grupo de URL's dependiendo de la necesidad de información que proporciona cada uno (ver ejemplo), es como el archivo URLS.txt.
- ✓ Envíe un e-mail donde indique los tres tipos de necesidad de información y el directorio con los URL's que recopiló y el archivo URLS.txt, junto con los documentos WORD y EXCEL. El e-mail debe ser mandado con el *subject* "Etapa I – Equipo #", y los archivos que deben adjuntar deben estar comprimidos en un archivo *zip* con el nombre "EtapaI_Equipo#.zip".
- ✓ Presente la documentación respectiva impresa (documento WORD) y, un CD con el directorio con los URL's que recopiló y el archivo URLS.txt, junto con los documentos WORD y EXCEL, el día indicado a la hora de la clase.

II ETAPA: I PARTE DEL INDEXADOR

Fecha de Entrega: **24 / 04 / 2020**

Valor: **10%**

Enunciado:

Debe implementar un programa que lea el archivo llamado "URLS" dado por la profesora (este archivo será la unión de cada archivo "URLS" entregados por los grupos de proyecto de la etapa anterior) y; utilizando este archivo y el conjunto de reglas dado (debe explicar el propósito de cada una) (puede agregar otras reglas y no realizar algunas de las reglas propuestas (justificar)), producir como resultado un archivo de texto con extensión *.tok* por cada documento *html* procesado (si los nombres de los archivos fueran números: 1.html produce *1.tok*, etc.). Producir además un archivo de texto con el vocabulario de la colección (el nombre del archivo debe ser "Vocabulario" para estandarizar soluciones).

Posibles reglas para procesamiento de documentos en HTML son:

1. Eliminar etiquetas <> con todo y sus atributos.
2. Extraer términos de tamaño máximo 30.
3. Pasar el término a minúsculas.
4. Símbolos válidos a-z, A-Z, 0-9 y _.
5. Se eliminan tildes y quedan vocales simples, por ejemplo: í se pasa a una i.
6. Convertir la ñ en n.
7. Todo término inicia con a-z no numérico (sólo para palabras que tienen números y letras).
8. Guardar algún rango de números. Por ejemplo: 0 a 10000.

Los formatos de salida propuestos son (todos los archivos deben estar ordenados alfabéticamente):

- Para los documentos *.tok* cada línea define un término con su respectiva frecuencia (número de veces que aparece en el documento) y frecuencia normalizada. Se recomienda que el ancho de la columna sea fijo y corresponda a (puede hacer las modificaciones justificadas que considere conveniente):
  - 30 caracteres para el término.
  - 1 espacio en blanco.
  - 12 espacios para la frecuencia de aparición del término en el documento (*freq*).
  - 1 espacio en blanco.
  - 20 espacios para la frecuencia normalizada del término dentro del documento (*tf*).
  - Cambio de línea.
- En el archivo "Vocabulario" cada línea define un término de la colección con su respectiva frecuencia total y número de documentos diferentes donde aparece cada término (puede hacer las modificaciones justificadas que considere conveniente):
  - 30 caracteres para el término.
  - 1 espacio en blanco.

- 12 espacios para el número de documentos donde aparece.
- 1 espacio en blanco.
- 20 espacios para la frecuencia inversa del término dentro de los documentos de la colección (*idf*).
- Cambio de línea.

Puede utilizar cualquier lenguaje de programación orientado a objetos, preferiblemente Java.

Ejecute su programa con los documentos recolectados en la I Etapa del Motor de Búsqueda, recuerde que deben ser al menos 200 URL's. Conservar los archivos URLS, *.tok* y Vocabulario para ser usados en la siguiente etapa del proyecto.

Entregue el código fuente del programa, el ejecutable, la colección de documentos usada, los archivos obtenidos: URLS, *.tok* y Vocabulario. Además, una pequeña documentación de esta II Etapa del Motor de Búsqueda, donde incluya:

- Descripción general del programa.
- Los pasos a seguir al realizar la II Etapa.
- Las reglas utilizadas para formatear los documentos y la explicación del propósito de cada una, la justificación de agregar o eliminar alguna regla.
- Los formatos de salida de los archivos, la justificación de cambios hechos al formato establecido.
- La descripción de cada una de las clases, las variables y los métodos utilizados en cada clase (UML).
- Los problemas surgidos, tanto resueltos como no resueltos, soluciones y mejoras para la siguiente etapa.
- La división de trabajo.

Instrucciones:

- ✓ Realizar el programa cuya funcionalidad se explicó anteriormente en el enunciado de esta etapa.
- ✓ Realizar la documentación que incluya lo explicado en el enunciado de esta etapa. Siga el ejemplo para presentar documentos, dado en la Etapa I.
- ✓ Envíe un e-mail donde anexe los archivos obtenidos: URLS.txt, *.tok* y Vocabulario.txt, y la documentación. El e-mail debe ser mandado con el *subject* "Etapa II – Equipo #", y los archivos que deben adjuntar deben estar comprimidos en un archivo *zip* con el nombre "EtapaII_ Equipo#.zip".
- ✓ Presente la documentación respectiva impresa y, un CD con el código fuente del programa, el ejecutable, los archivos obtenidos: URLS.txt, .tok y Vocabulario.txt, y la documentación, el día indicado a la hora de la clase.

III ETAPA: II PARTE DEL INDEXADOR

Fecha de Entrega: **29 / 05 / 2020**

Valor: **10%**

Enunciado:

Debe agregarle al programa de la Etapa II diferentes cálculos y la creación de otros archivos necesarios para el buen desempeño del motor de búsqueda. Las nuevas funciones son:

- Crear un archivo de pesos con extensión *.wtd* por cada documento *.tok*, donde se agrega cada término y el peso del término en el documento (*w*).
- Crear un archivo llamado "Indice", que contiene todos los términos del vocabulario, la posición inicial y la cantidad de entradas dentro del archivo de "Postings".
- Crear el archivo llamado "Postings", que contiene todos los términos del vocabulario, el alias del archivo URL y el peso del término en el documento.

Los formatos de salida propuestos son (todos los archivos deben estar ordenados alfabéticamente):

- Para los documentos *.wtd* cada línea define un término con su respectivo peso. Se recomienda que el ancho de la columna sea fijo y corresponda a (puede hacer las modificaciones justificadas que considere conveniente):
  - 30 caracteres para el término.
  - 1 espacio en blanco.
  - 20 espacios para el peso del término en el documento (*w*).
  - Cambio de línea.
- En el archivo "Indice" cada línea define un término del vocabulario, la posición inicial y la cantidad de entradas dentro del archivo de "Postings". Se recomienda que el ancho de la columna sea fijo y corresponda a (puede hacer las modificaciones justificadas que considere conveniente):
  - 30 caracteres para el término.
  - 1 espacio en blanco.
  - 12 espacios para la posición inicial dentro del archivo "Postings" (número de línea).
  - 1 espacio en blanco.
  - 12 espacios para la cantidad de entradas dentro del archivo de "Postings" (el número de documentos donde aparece).
  - Cambio de línea.
- En el archivo "Postings" cada línea un término del vocabulario, el alias del archivo URL y el peso del término en el documento. Se recomienda que el ancho de la columna sea fijo y corresponda a (puede hacer las modificaciones justificadas que considere conveniente):
  - 30 caracteres para el término.
  - 1 espacio en blanco.

- 15 caracteres para el alias.
- 1 espacio en blanco.
- 20 espacios para el peso del término en el documento (*w*).
- Cambio de línea.

Puede utilizar cualquier lenguaje de programación orientado a objetos, preferiblemente Java. Conservar los archivos URLS, *.tok*, *.wtd*, Vocabulario, Indice y Postings para ser usados en la siguiente etapa del proyecto.

Entregue el código fuente del programa, el ejecutable, la colección de documentos usada, los archivos obtenidos: URLS, *.tok, .wtd*, Vocabulario, Indice y Postings. Además, agregarle a la documentación de la II Etapa la documentación de esta III Etapa del Motor de Búsqueda, donde incluya:

- Documentación de la II Etapa.
- Descripción general del programa.
- Los pasos a seguir al realizar la II Etapa y la III Etapa.
- Los formatos de salida de los archivos, la justificación de cambios hechos al formato establecido.
- La descripción de cada una de las clases, las variables y los métodos utilizados en cada clase (UML).
- Los problemas surgidos, tanto resueltos como no resueltos, soluciones y mejoras para la siguiente etapa.
- La división de trabajo.

Instrucciones:

- ✓ Realizar el programa cuya funcionalidad se explicó anteriormente en el enunciado de esta etapa.
- ✓ Realizar la documentación que incluya lo explicado en el enunciado de esta etapa. Siga el ejemplo para presentar documentos, dado en la Etapa I.
- ✓ Envíe un e-mail donde anexe los archivos obtenidos: URLS.txt, *.tok*, *.wtd*, Vocabulario.txt, Indice.txt y Postings.txt, y la documentación. El e-mail debe ser mandado con el *subject* "Etapa III – Equipo #", y los archivos que deben adjuntar deben estar comprimidos en un archivo *zip* con el nombre "EtapaIII_ Equipo#.zip".
- ✓ Presente la documentación respectiva impresa y, un CD con el código fuente del programa, el ejecutable, los archivos obtenidos: URLS.txt, .tok, .wtd, Vocabulario.txt, Indice.txt y Postings.txt, y la documentación, el día indicado a la hora de la clase.

IV ETAPA: MOTOR DE BÚSQUEDA E INTERFACE

Fecha de Entrega: **30 / 06 / 2020 – 03 / 07 / 2020**

Valor: **10%**

Enunciado:

Debe agregarle al programa de la Etapa III el motor de búsqueda y la interfaz del usuario:

- El motor de búsqueda es el que recibirá la consulta hecha por el usuario, deberá calcular la similitud entre la consulta y cada uno de los documentos respectivos (según lo indique el Índice) y devolverá los resultados correspondientes ordenados del más al menos relevante.
- La interfaz del usuario deberá ser Web, o sea, una página HTML donde el usuario digite las consultas y se desplieguen los resultados devueltos (parecido a cualquier buscador).
    - La interfaz debe tener para digitar la consulta como mínimo: el nombre del SRI, el cuadro de texto para digitar la consulta y el botón para buscar.
    - La interfaz debe tener para desplegar los resultados devueltos como mínimo: el nombre del SRI, el cuadro de texto con la consulta digitada, el botón para buscar, el total de documentos que se recuperaron, el tiempo transcurrido para resolver la consulta, los resultados (en grupos de 10 por página) y, botones de anterior y/o siguiente (según corresponda).
    - Los resultados deben aparecer, 10 por página, que incluya los siguientes datos: alias del documento recuperado (que tendrá el link al URL respectivo del documento) y una opción "En Caché" (que tendrá el link a la ubicación local del documento recuperado, donde está guardado el documento localmente), por ejemplo: 1.html (En caché).

Puede utilizar un Servlet, JSP o un CGI-BIN. Además, puede utilizar cualquier lenguaje de programación orientado a objetos, preferiblemente Java.

Entregue el código fuente de los programas, los ejecutables, la colección de documentos usada, los archivos obtenidos: URLS, *.tok, .wtd*, Vocabulario, Indice y Postings. Además, agregarle a la documentación de la III Etapa la documentación de esta IV Etapa del Motor de Búsqueda, donde incluya:

- Documentación de la III Etapa.
- Descripción general del programa.
- Los pasos a seguir al realizar la II Etapa, la III Etapa y IV la Etapa.
- La descripción de cada una de las clases, las variables y los métodos utilizados en cada clase (UML).
- Los problemas surgidos, tanto resueltos como no resueltos, soluciones y mejoras.
- La división de trabajo.
- El manual de usuario.

Instrucciones:

✓ Realizar el programa cuya funcionalidad se explicó anteriormente en el enunciado de esta etapa.

- ✓ Realizar la documentación que incluya lo explicado en el enunciado de esta etapa. Siga el ejemplo para presentar documentos, dado en la Etapa I.
- ✓ Envíe un e-mail donde anexe los archivos obtenidos: URLS.txt, *.tok*, *.wtd*, Vocabulario.txt, Indice.txt y Postings.txt, y la documentación. El e-mail debe ser mandado con el *subject* "Etapa IV – Equipo #", y los archivos que deben adjuntar deben estar comprimidos en un archivo *zip* con el nombre "EtapaIV_ Equipo#.zip".
- ✓ Presente la documentación respectiva impresa y, un CD con el código fuente del programa completo, los ejecutables, los archivos obtenidos: URLS.txt, .tok, .wtd, Vocabulario.txt, Indice.txt y Postings.txt, y la documentación, el día indicado a la hora de la clase.