|  |
| --- |
| Instituto Tecnológico de Costa Rica |
| Documentación TP2 |
| Recuperación de Información Textual |

|  |
| --- |
| Daniel Cortés Sáenz, Isaac Ramírez Solano  24/06/2013 |

Contenido

[Introducción 2](#_Toc359749548)

[Ambiente de Desarrollo 2](#_Toc359749549)

[Instrucciones para ejecutar los programas 2](#_Toc359749550)

[Archivo Invertido 2](#_Toc359749551)

[Consultas 2](#_Toc359749552)

[Page Rank 2](#_Toc359749553)

[Corridas 3](#_Toc359749554)

[Archivo Invertido 3](#_Toc359749555)

[Consultas 3](#_Toc359749556)

[Page Rank 3](#_Toc359749557)

[Comentarios Finales 3](#_Toc359749558)

# Introducción

La segunda tarea programada consiste en la implementación de un archivo invertido y la realización de búsquedas usando ese archivo invertido. Para ello, se debe implementar la estructura del archivo invertido que consta del diccionario, documentos y archivo de postings.

Además, se debe implementar un algoritmo para calcular el Page Rank de la colección. Este algoritmo recibe como parámetros la diferencia esperada y la cantidad de iteraciones. Al final muestra el Page Rank de los documentos de forma descendente.

# Ambiente de Desarrollo

Para esta segunda tarea programada, se usó Perl (5.16.3) y Java (1.7.25). El sistema operativo usado fue Windows 8. Se decidió usar Perl porque es un lenguaje de scripting bastante rápido. Java se encarga de procesar los documentos que genera Perl ya que las matrices y arreglos son más fáciles de manejas que Perl.

Además, se utilizó Github como manejador de versiones. El repositorio de la tarea está disponible públicamente en [este](https://github.com/dcortes92/rit2.git) enlace.

# Instrucciones para ejecutar los programas

## Archivo Invertido

## Consultas

## Page Rank

Para correr el Page Rank, se debe ingresar alguno de los siguientes comandos:

1. perl parser.pl pr
2. perl parser.pl 0.0003 500

El primer comando asume que la diferencia esperada y la cantidad iteraciones serán 0.0001 y 100. En el segundo, se especifica que la diferencia esperada será 0.0003 y 500.

Es importante aclarar que este programa debe ser ejecutado **después** de haber creado el archivo invertido.

# Corridas

## Archivo Invertido

## Consultas

## Page Rank

Se adjunta el archivo ResultadoPageRank.txt dentro del directorio “Documentación” de la tarea que tiene el Page Rank de los documentos con 1 iteración.

# Comentarios Finales

La tarea se terminó completamente. Se pudo implementar tanto el archivo invertido como las consultas y el Page Rank. Dentro de los problemas encontrados se pueden mencionar los siguientes:

1. No se sabía cómo extraer los URL de las etiquetas de <a> ni cómo formatear el HTML a que solo se procese el texto. Para esto, se usó el HTML Parser que trae Perl.
2. Los arreglos multidimensionales son difíciles de usar el Perl. Para este caso, se decidió usar Java para que procese él estas partes y Perl se encargue de todo lo demás.

En general a todos los problemas encontrados se les encontró una solución. La única limitación es que el Page Rank requiere el archivo invertido para funcionar.