Análisis y Recuperación de Información

1er Cuatrimestre 2017

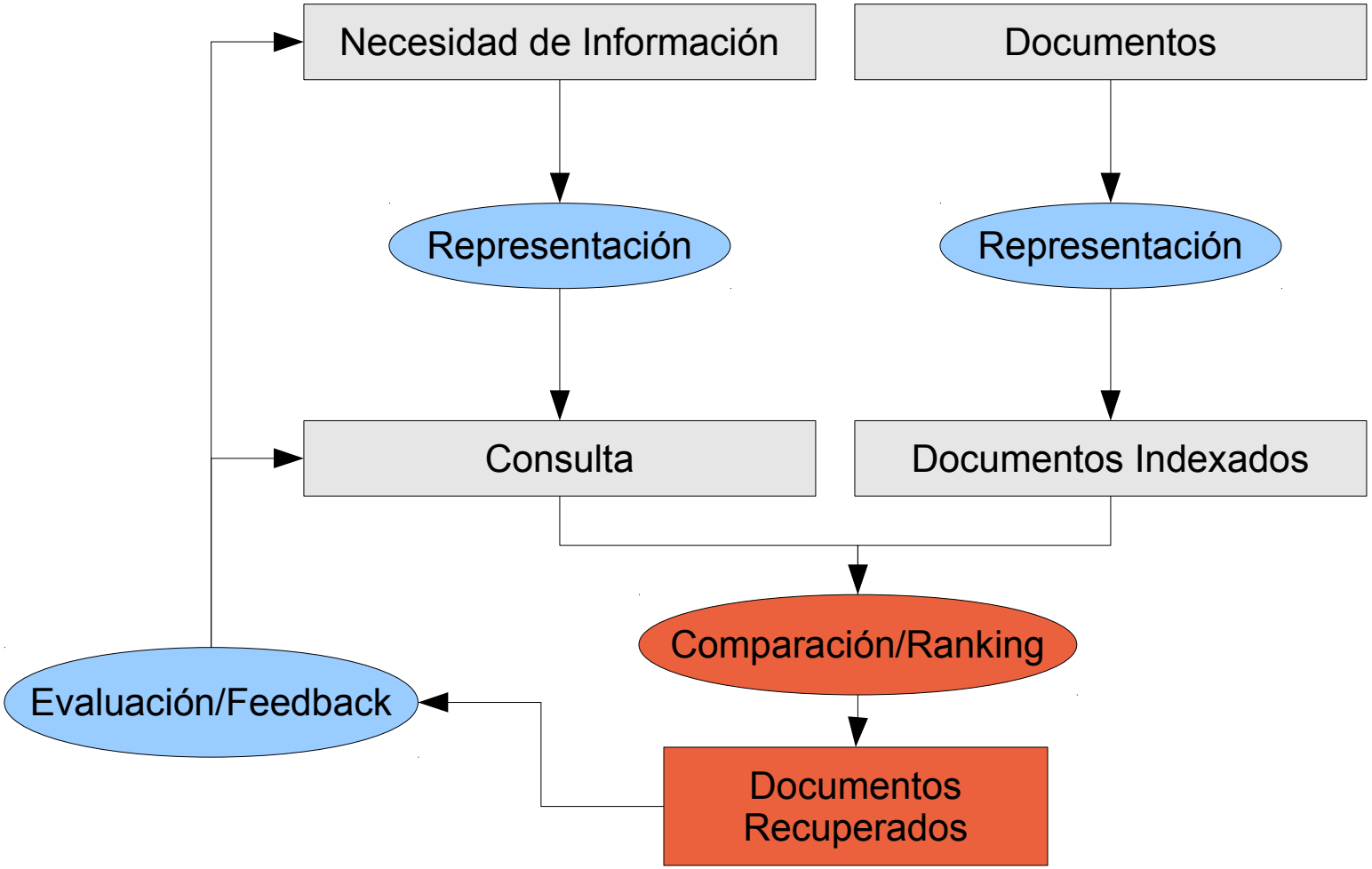
Página Web

http://www.exa.unicen.edu.ar/catedras/ayrdatos/

Prof. Dra. Daniela Godoy

ISISTAN Research Institute
UNICEN University
Tandil, Bs. As., Argentina
http://www.exa.unicen.edu.ar/~dgodoy
dgodoy@exa.unicen.edu.ar

Recuperación de Información



Ranking

- El problema de ranking se enfoca en la forma de ordenar los resultados de una búsqueda
- Cómo se rankean los resultados de una búsqueda?
  - frecuentemente usando la similitud entre la consulta y el documento
  - en un proceso iterativo se pueden rankear los documentos de acuerdo a su similitud a la consulta y a los documentos previamente marcados como relevantes o irrelevantes
  - otros factores que se pueden utilizar para calcular el ranking son las citaciones o referencias

Ranking

- El ranking tiene que ser realizado con acceso sólo a los índices y no el texto
- Características locales
  - Frecuencia, font, font size relativo, tags, etc.
- Características externas
  - Citaciones de los documentos: que tan frecuentemente es citada una página, la importancia de los documentos que la citan
  - Ubicación el sitio Web: altura en la estructura de directorios o distancia en links
  - Popularidad de las páginas en consultas similares

Análisis de links

- Idea fundamental:
  - los links contienen información o juicios de relevancia de una página
  - cuanto más links entrantes a una página, más puede ser considerada importante
- Bray 1996
  - la visibilidad de un sitio se puede medir por el número de sitios que apuntan a él
  - la luminosidad de un sitio puede medirse como el número de sitios a los cuales apunta
  - falla en capturar la importancia relativa de los diferentes sitios que apuntan o son apuntados

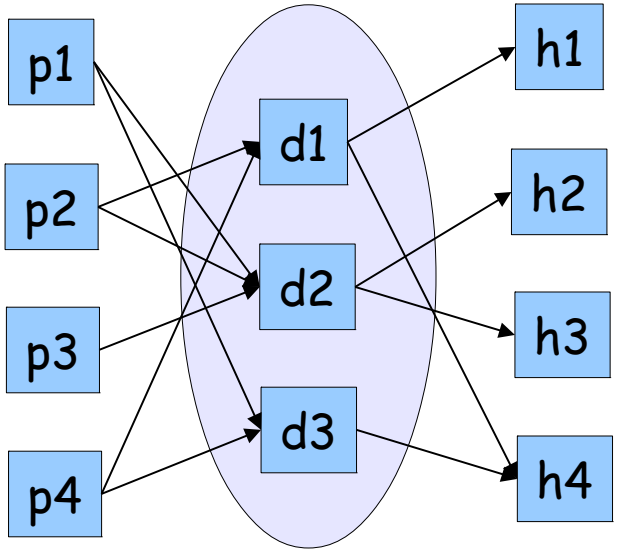
Análisis de links

- El análisis de links tiene por objetivo rankear páginas sacando ventaja de la estructura de links en la Web
- Enfoques
  - Estático
    - se usan los links para calcular un ranking de las páginas off-line (Google) → PageRank
  - Dinámico
    - se usan los links en el resultado de la búsqueda para determinar el ranking (CLEVER de IBM) → HITS

HITS

- HITS – Hypertext Induced Topic Selection
- Objetivo: obtener el ranking para una consulta particular (enfoque dinámico) en lugar de toda la Web

- existe un buscador que puede devolver un conjunto de páginas *S* que coinciden con la consulta
- encontrar todas las páginas que apuntan a *S* (padres) y aquellas a las que *S* apunta (hijos)
- estas páginas se agregan al conjunto de documentos



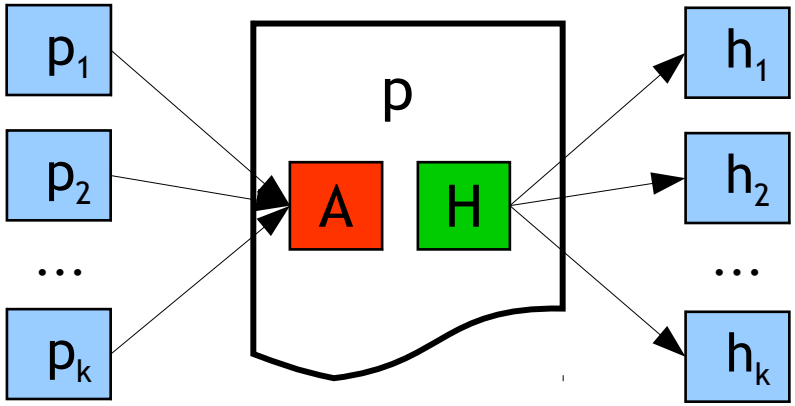
Resultados *S*

HITS

- Para cada consulta se trata de encontrar:
  - *Authorities*: autoridades en el tópico (que poseen la información real)
  - *Hubs*: sitios que apuntan a la mejor información en el tópico (generalmente sitios de links)

HITS

- Intuición:
  - la *autoridad* está dada por los links entrantes
  - los *hubs* provienen de los links salientes
- Reforzando la intuición:
  - mayor autoridad vienen de links entrantes de son buenos *hubs*
  - ser mejor *hub* proviene de links salientes a buenas autoridades



HITS

- Es un algoritmo iterativo para gradualmente converger a un conjunto de *hubs* y *autoridades*
- Para cada página p ∈ S:
  - valor de autoridad: a_p (vector $\boldsymbol{a}$)
  - valor de hub: h_p (vector $\boldsymbol{h}$)
- Inicializar todos los $a_p = h_p = 1$
- Mantener los valores normalizados:

$$\sum_{p \in S} \left(a_p\right)^2 = 1 \qquad \sum_{p \in S} \left(h_p\right)^2 = 1$$

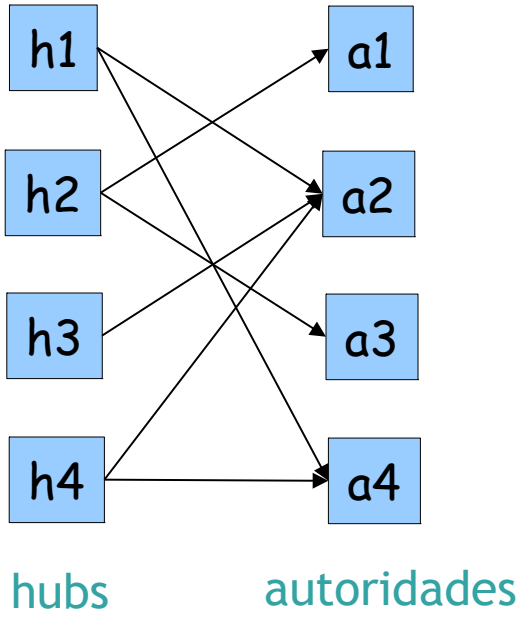
HITS

- Reglas de actualización
  - el vector ***a*** tiene un peso para cada documento en *D* para indicar que tan buena autoridad es

$$a_p = \sum_{q:q \to p} h_q$$

  - el vector ***h*** tiene un peso para cada documento en *D* para indicar que tan buen *hub* es

$$h_p = \sum_{q:p \to q} a_q$$



HITS

Inicialización

For all $p \in S$: $a_p = h_p = 1$

For i = 1 to k:

For all $p \in S$: $a_p = \sum_{q:q \to p} h_q$ (actualizar *autoridades*)

For all $p \in S$: $h_p = \sum_{q:p \to q} a_q$ (actualizar *hubs*)

For all $p \in S$: $a_p = a_p / c$ c:

For all $p \in S$: $h_p = h_p / c$ c:

$$\sum_{p \in S} (a_p)^2 = 1$$ (normalizar ***a***)

$$\sum_{p \in S} (h_p)^2 = 1$$ (normalizar ***h***)

HITS

- El algoritmo converge a un número de punto fijo si itera indefinidamente, en la práctica 20 iteraciones producen un resultado estable
- Resultados:
  - Autoridades para la consulta: “Java”
    - java.sun.com
    - comp.lang.java FAQ
  - Autoridades para la consulta: “search engine”
    - Yahoo.com
    - Excite.com
    - Lycos.com
    - Altavista.com
  - Autoridades para la consulta: “Gates”
    - Microsoft.com

HITS

- Muchos links no son correctos (no son recomendaciones)
  - muchos links de un mismo autor
  - muchos links generados automáticamente
  - una solución podría ser pesar los links
- Topic drift
  - los hubs y autoridades tienden a moverse hacia algo general, en lugar de más específico
    - "jaguar AND cars" resulta en páginas acerca de autos en general
  - una página muy importante puede ser superficial al tema
  - una solución podría ser analizar el contenido y asignar pesos a los nodos de acuerdo al tópico (por similitud con la consulta)

HITS

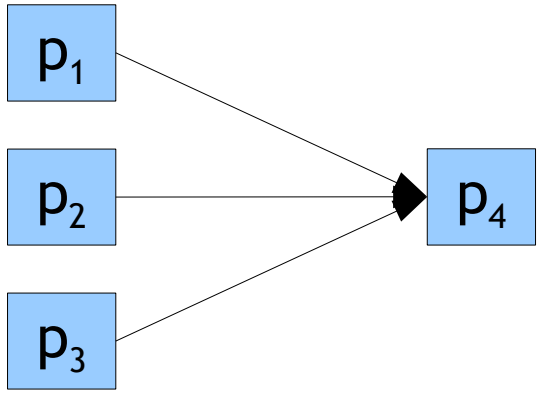
- Requiere cálculos y recuperación *on-the-fly*
- Limitaciones para reducir el costo computacional:
  - limitar el número de paginas de las que se extraen links a las primeras *N* recuperadas para la consulta
  - eliminar links puramente navegacionales (por ejemplo en el mismo sitio)

PageRank

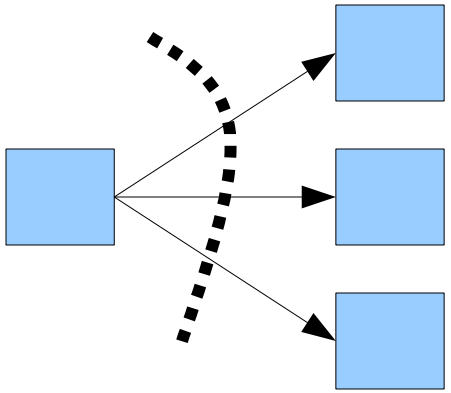
- Hace uso de la estructura de links de la Web para calcular un ranking de calidad (PageRank) para cada página
- Cada página tiene un único PageRank, independientemente de la consulta
- El valor de PageRank no expresa la relevancia de la página a la consulta
- Intuición:
  - la importancia de una página puede decidirse por el número de páginas que apuntan a ella
  - una implementación simple seria contar estas páginas para cada página
  - se puede engañar fácilmente generando muchas páginas que nada más a una pagina dada

PageRank

- El prestigio de una página es proporcional a la suma de los prestigios de las páginas que la citan
- Medida estándar de influencia en bibliometría
- El algoritmo simula un paseo aleatorio en la Web para calcular el prestigio de todas las páginas
- Ordena las respuestas que coinciden con la consulta por orden decreciente de prestigio



Seguir un link aleatoriamente



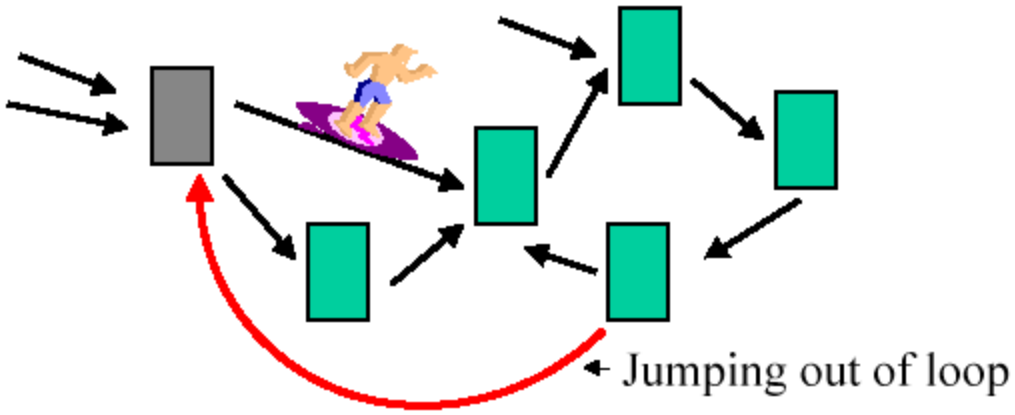
PageRank

- Para comenzar cada página tiene un peso inicial, supongamos 1
- En cada iteración, cada página propaga su peso actual *W* a todos sus *N* vecinos hacia adelante, cada uno de ellos recibe un peso *W/N*
- Entonces, una página acumula los pesos de sus vecinos hacia atrás
- El algoritmo itera hasta que todos los pesos convergen, usualmente 6 o 7 veces es suficiente
- El peso final de cada página es su importancia

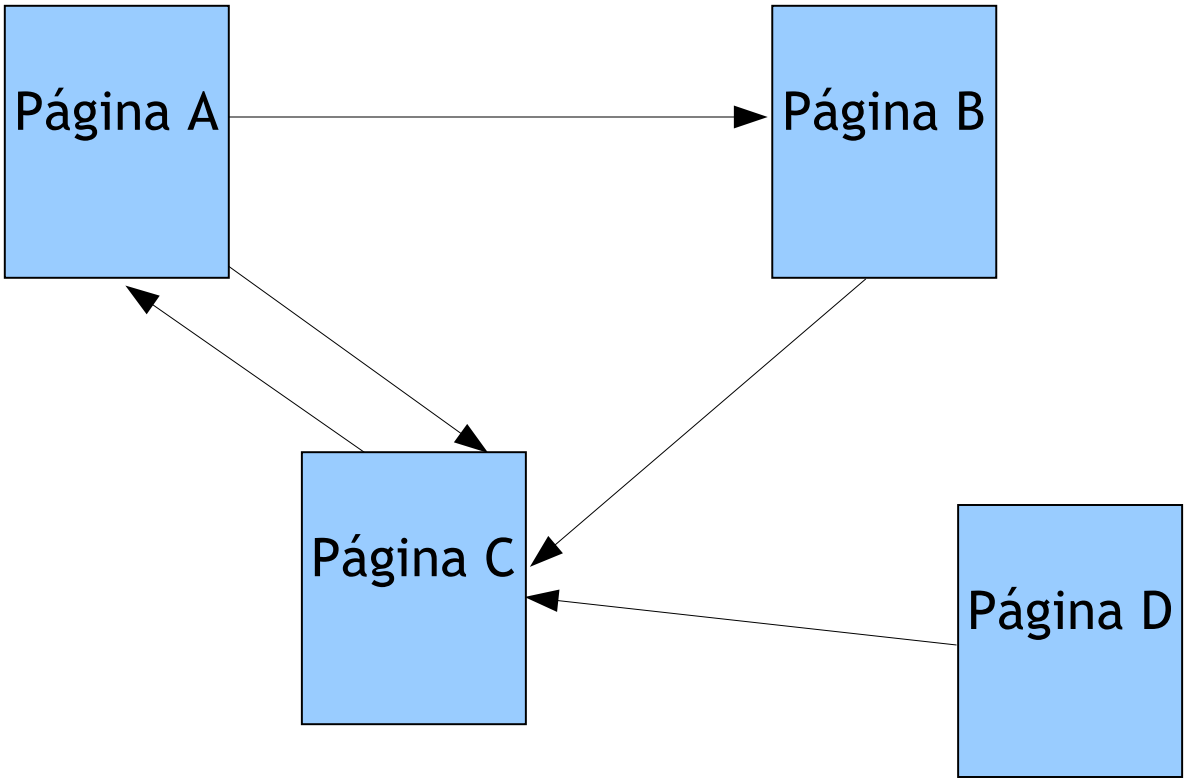
PageRank

$$PR(a) = q + (1-q)\sum_{i=1}^{n} \frac{PR(p_i)}{C(p_i)}$$

- PageRank simula un usuario que navega aleatoriamente en la Web, quien salta a una página aleatoria con probabilidad *q* o que sigue un link aleatorio (en la página actual) con probabilidad 1 - *q*
- Sea C(*a*) el número de enlaces de salida de una página *a* y suponga que la página está apuntada por páginas p_1 a p_n

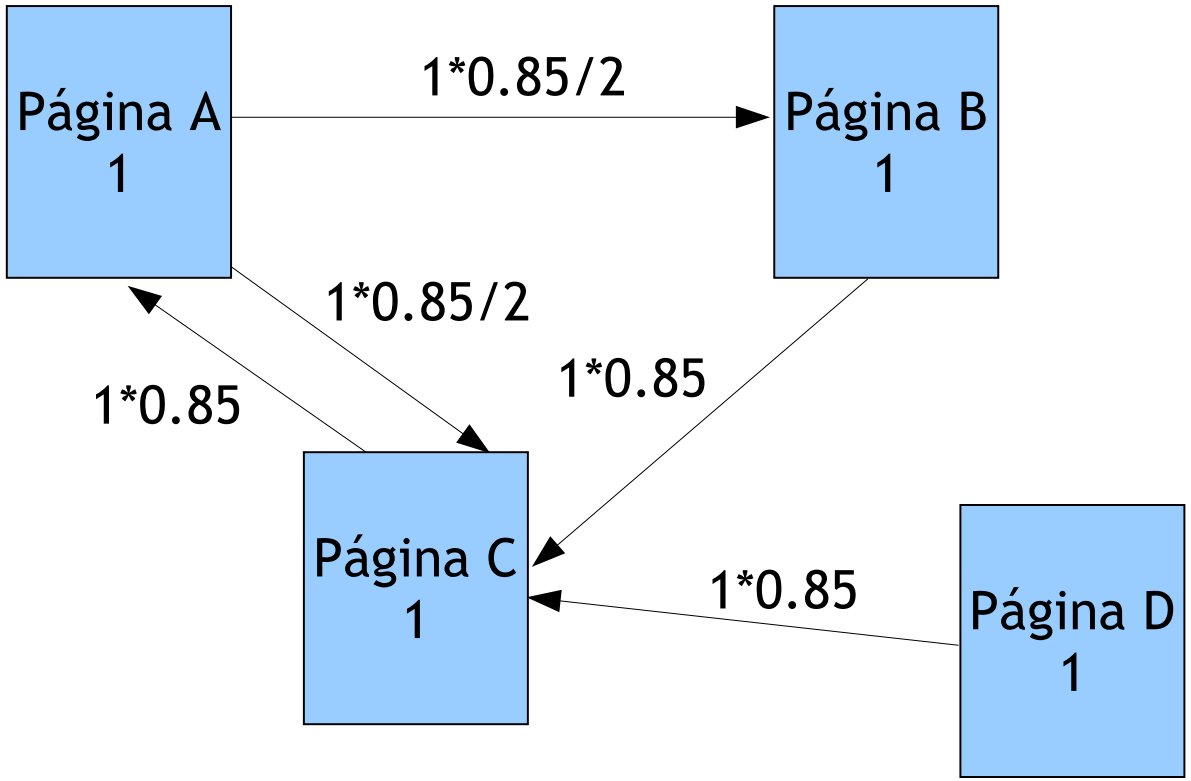


PageRank



Paso 1

PageRank



Paso 2

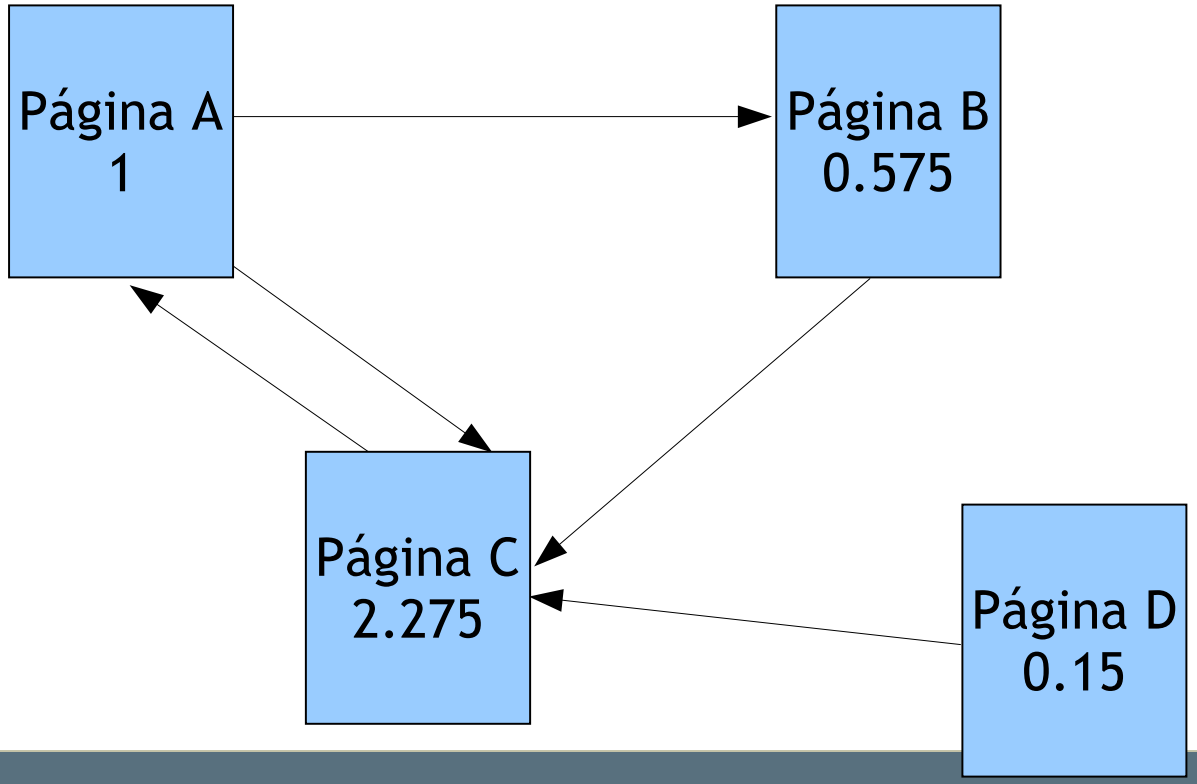
PageRank

Página A: 0.85 (de C) + 0.15 = 1

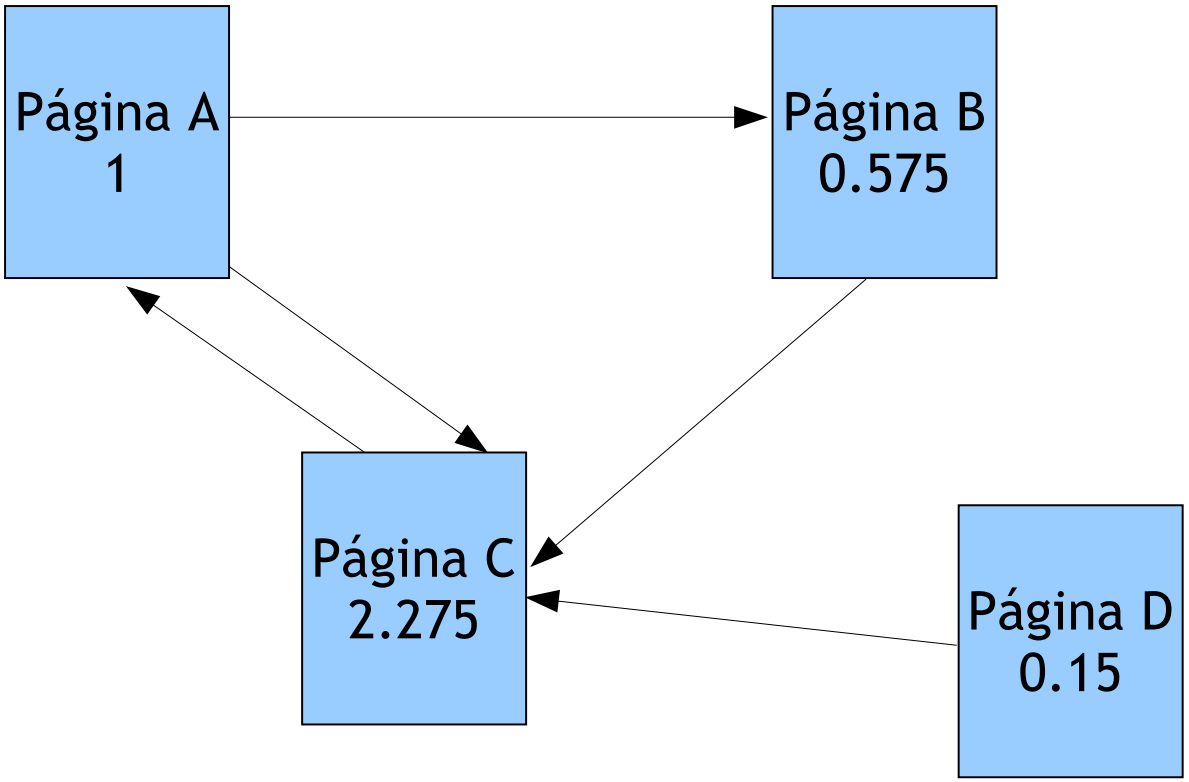
Página B: 0.425 (de A) + 0.15 = 0.575

Página C: 0.85 (de D) + 0.85 (de B) + 0.425 (de A) + 0.15 = 2.275

Página D: no recibe nada + 0.15 = 0.15



PageRank



Paso 3

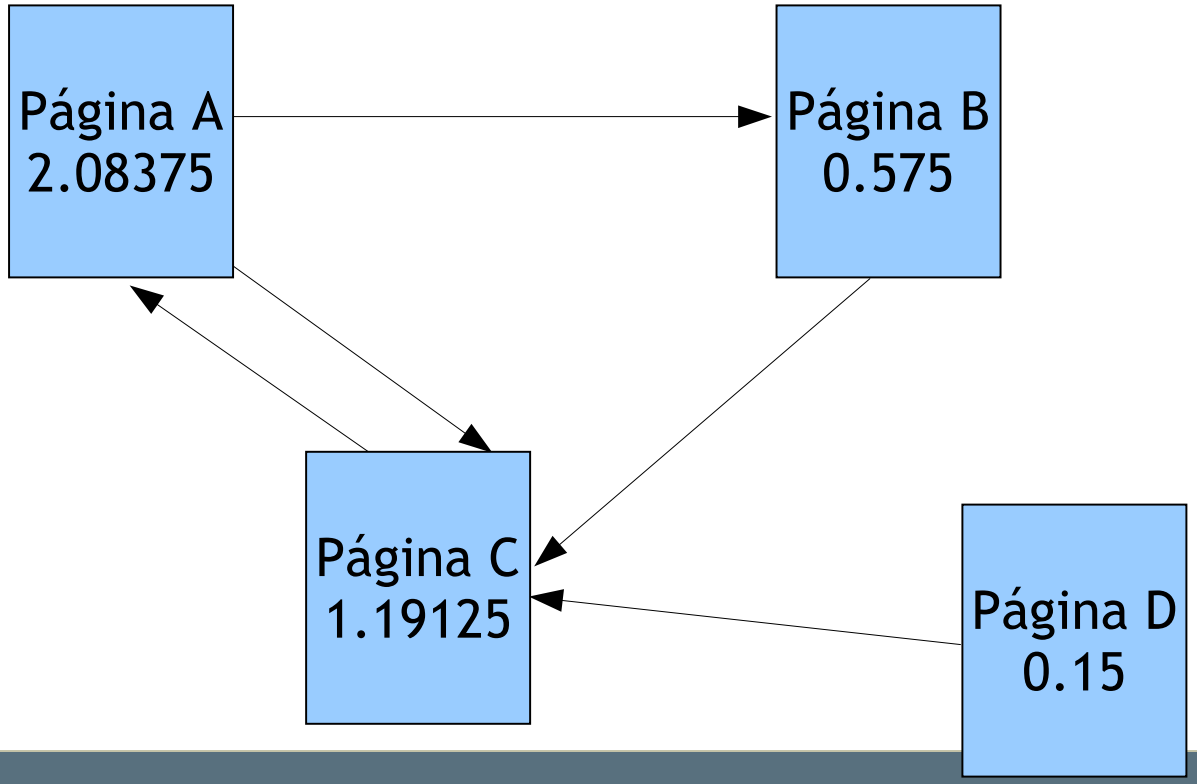
PageRank

Página A: 2.275*0.85 (de C) + 0.15 = 2.08375

Página B: 1*0.85/2 (de A) + 0.15 = 0.575

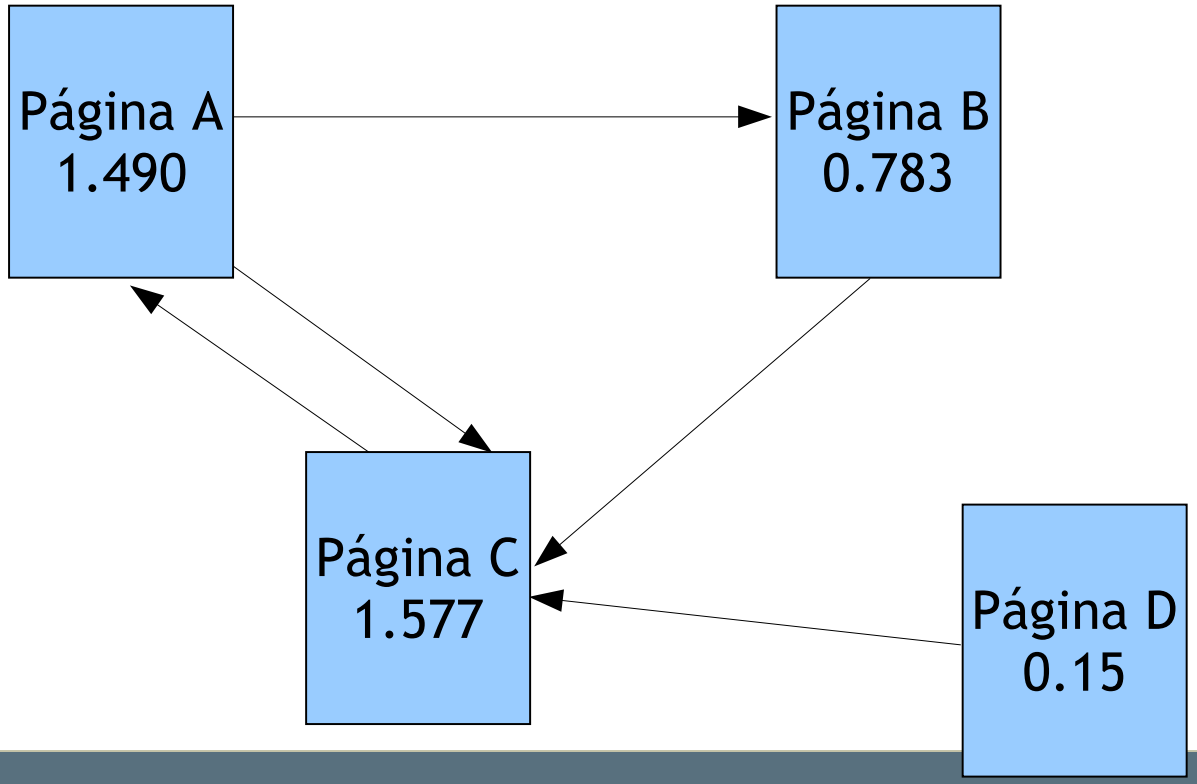
Página C: 0.15*0.85 (de D) + 0.575*0.85(de B) + 1*0.85/2 (de A) +0.15 = 1.19125

Página D: no recibe nada + 0.15 = 0.15



PageRank

- Luego de 20 iteraciones, la página C tiene el PageRank más alto y la página A tiene el siguiente más alto, la página C tiene la mayor importancia en el grafo



PageRank

- En los experimentos iniciales de Google se usaron 322 millones de links
- PageRank convergió, con una pequeña tolerancia, en alrededor de 52 iteraciones
- El número de iteraciones requeridos para converger es empíricamente O(log *n*), sonde *n* es el número de links
- En conclusión el cálculo es bastante eficiente

PageRank

- El sistema de ranking de Google (basado en publicaciones previas a su comercialización) consideraba:
  - similitud en el espacio de vectores
  - HTML tags con diferentes pesos (títulos, etc.)
  - PageRank
- El análisis de links usa información de la estructura del grafo que forma la Web para ayudar en la búsqueda
- Es una de las mayores innovaciones en lo que refiere a la búsqueda Web y una de las razones del éxito de Google