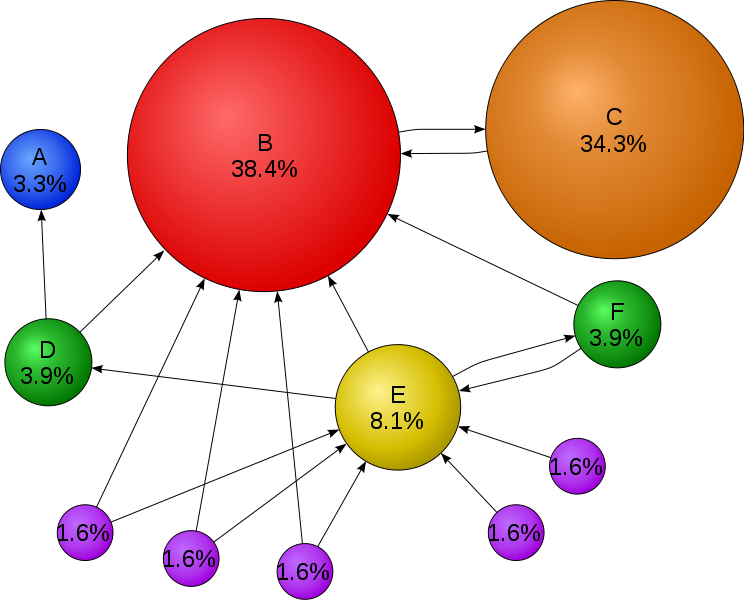
PageRank ha tomado su modelo del Science Citation Index (SCI) elaborado por Eugene Garfield para el Instituto para la Información Científica (ISI) en los Estados Unidos durante la década del 50. El SCI pretende resolver la asignación objetiva de méritos científicos suponiendo que los investigadores cuyo factor de impacto (número de publicaciones y/o referencias bibliográficas en otros trabajos científicos) es más alto, colaboran en mayor medida con el desarrollo de su área de investigación. El índice de citación es un elemento determinante para seleccionar qué investigadores reciben becas y recursos de investigación

El **algoritmo del PageRank de Google**, es decir, la fórmula que utiliza para calcular el **PageRank** de una página es uno de los mayores y más deseados secretos de Internet.

El algoritmo PageRank

[](http://upload.wikimedia.org/wikipedia/commons/f/fb/PageRanks-Example.svg)

El **algoritmo original del PageRank** ha sido descrito en varias ocasiones por Lawrence Page y Sergey Brin. Es el siguiente:

PR(A) = (1-d) + d (PR(T1)/C(T1) + … + PR(Tn)/C(Tn))

Las variables son: El [algoritmo](http://es.wikipedia.org/wiki/Algoritmo) **inicial** del PageRank lo podemos encontrar en el documento original donde sus creadores presentaron el [prototipo](http://es.wikipedia.org/wiki/Prototipo) de [Google](http://es.wikipedia.org/wiki/Google): “The Anatomy of a Large-Scale Hypertextual Web Search Engine":[[2]](http://es.wikipedia.org/wiki/PageRank#cite_note-1)

**PR(A) = (1-d) + d  * \sum_{i=1}^n {PR(i) \over C(i)}**

Donde:

* **PR(A)** es el PageRank de la página A.
* **d** es un factor de amortiguación que tiene un valor entre 0 y 1.
* **PR(i)** son los valores de PageRank que tienen cada una de las páginas **i** que enlazan a A.
* **C(i)** es el número total de enlaces salientes de la página i (sean o no hacia A).
* PR(A) Es el PageRank de la página A,
* PR(Ti) Es el PageRank de las páginas Ti que enlazan a A,
* C(Ti) Es el número de links salientes de la página Ti
* d es un factor variable que puede estar entre 0 y 1.

Así­ pues, lo primero de todo, vemos que **el PageRank no se aplica a un sitio Web en su totalidad, sino a sus páginas individualmente**. Además, el PageRank de la página A se define de forma recursiva por el PageRanks de las páginas que enlazan a esa página A.

El **PageRank** de las páginas Ti que enlazan a la página A no afecta al PageRank de la página A uniformemente. Dentro del algoritmo del PageRank, el PageRank de una página T se calcula por el número de links salientes C (T) en la página T. **Esto significa que cuantos más enlaces salientes tenga la página T, menos beneficiará el anlace de la página A al enlazar a la página T**.

El **PageRank transmitido** de las páginas Ti se suma. La consecuencia de esto es que otros enlaces adicionales a la página A aumentarán siempre el PageRank de la página A.

Finalmente, la suma del PageRank transmitido de las páginas Ti se multiplica por un factor aleatorio d que puede estar entre 0 y 1. De este modo, **la transmisión del PageRank que beneficia a una página que es enlazada por otra se reduce.**

El Modelo Navegación Aleatoria

Lawrence Page y Sergey Brin justifican de una forma simple e intuitiva el funcionamiento del algoritmo PageRank. Consideran el PageRank como un modelo del comportamiento del usuario, donde una persona que navega cliquea en enlaces al azar sin reparar en el contenido.

**El usuario que visita una página puede cambiar o influir en el PageRank de la página**. La probabilidad de que el ususario presione un enlace solo depende del número de enlaces de la página. **Esta es la razón por la cual el PageRank transmitido a una página que se enlaza disminuye, ya que se divide entre los links de la página.** Así­ pues, la probabilidad para el usuario de presionar sobre ese link en concreto depende del número de links de la página. Ahora, esta probabilidad se reduce por el factor aleatorio **d**. La explicación de este “Modelo de Navegación Aleatoria”, por lo tanto, es que el usuario no puede presionar sobre un número infinito de links, pero acaba cansándose y se va a otra página.

La probabilidad de que el usuario no deje de presionar el enlace viene dada por el factor **d**, que está, dependiendo del grado de probabilidad por lo tanto, entre 0 y 1. El **d** más alto es la probabilidad más alta de que el usuario presione en el enlace. Puesto que el usuario sale de la página cuando deja de presionar sobre los enlaces, la probabilidad se define como la constante (1-d) en el algoritmo. Sin tener en cuenta los enlaces entrantes, **la probabilidad de que el usuario presione el link es siempre (1-d)**, así­ que **una página siempre tiene el PageRank mí­nimo**.

Otra versión del algoritmo PageRank

Lawrence Page y Sergey Brin have han publicado varias versiones del algoritmo del PageRank. Otra formulación determina el PageRank de A según la fórmula:

PR(A) = (1-d) / N + d (PR(T1)/C(T1) + … + PR(Tn)/C(Tn))

* N es el número total de páginas del sitio

La segunda versión del algoritmo no se diferencia mucho de la primera. Con respecto al Modelo de Navegación Aleatoria, El PageRank de esta versión es la probabilidad real de que el usuario presione sobre el link entre todos los que existen. El **PageRank** es, entonces, **la probabilidad uniforme entre todas las páginas, así­ que la suma de los PageRanks de todas las páginas será el mismo.**

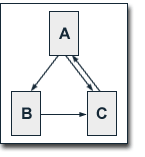
Por el contrario, en la primera versión del algoritmo la probabilidad de presionar el enlace viene dado por el número total de páginas. En esta versión, el PageRank es un valor previsto para el usuario cuando entra tan a menudo como páginas tiene el sitio. Es decir, **si un sitio tiene 100 páginas y una página tiene un PageRank 2, el usuario tendrá el doble de probabilidades de llegar a esa página si repite la llegada a la página 100 veces**.

Como se menciona arriba, no hay muchas diferencias entre la dos versiones. El PageRank calculado con la segunda versión del algoritmo tiene que ser multiplicado por el número total de páginas del sitio para que coincida con el **PageRank** calculado con la primera versión del algoritmo. Los autores reivindican el uso del primer algoritmo en su obra más famosa “The Anatomy of a Large-Scale Hypertextual Web Search Engine” para calcular el PageRank de un sitio como **la suma de los PageRanks de todas las páginas**.

A partir de ahora, utilizaremos la primera versión del algritmo para hacer cálculos ya que de esta forma evitamos el parámetro del número total de páginas.

Caracterí­sticas del PageRank

Las caracterí­sticas del PageRank pueden explicarse con un pequeño ejemplo:

Tenemos un sitio con tres páginas, A, B y C, donde A enlaza B y C, B enlaza C, y C enlaza A. De acuerdo con el algoritmo, el factor aleatorio d es de 0.85, pero para simplificar los cálculos lo redondeamos a 0.5. Este factor d sabemos que afecta al PageRank, pero no afecta a los principios del PageRank. Así­ que hagamos unos cálculos:

PR(A) = 0.5 + 0.5 PR(C)  
PR(B) = 0.5 + 0.5 (PR(A) / 2)  
PR(C) = 0.5 + 0.5 (PR(A) / 2 + PR(B))

Estas ecuaciones son sencillas. Nos queda el siguiente PageRank para cada página:

PR(A) = 14/13 = 1.07692308  
PR(B) = 10/13 = 0.76923077  
PR(C) = 15/13 = 1.15384615

Como es obvio, el PageRank de la suma de las páginas es 3, igual que el número de páginas. Este resultado no es especí­fico de nuestro sencillo ejemplo.

En nuestro caso es facil calcularlo directamente ya que solo tenemos 3 páginas. Pero actualmente los sitios están formados por miles de páginas, luego es necesario algún otro sistema de cálculo.

El método iterativo del PageRank

Debido al tamaño actual de la Web, **el buscador de Google usa un valor iterativo aproximado de PageRank**. Esto significa que a cada página se le asigna un valor inicial de PageRank, y después el PageRank de todas las páginas se calcula con cálculos cí­clicos basados en la fórmula del algoritmo de PageRank. Este cálculo iterativo puede representarse para nuestro ejemplo de tres páginas con un valor inicial de pageRank para cada página de 1:

|  |  |  |  |
| --- | --- | --- | --- |
| Iteration | PR(A) | PR(B) | PR(C) |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 0.75 | 1.125 |
| 2 | 1.0625 | 0.765625 | 1.1484375 |
| 3 | 1.07421875 | 0.76855469 | 1.15283203 |
| 4 | 1.07641602 | 0.76910400 | 1.15365601 |
| 5 | 1.07682800 | 0.76920700 | 1.15381050 |
| 6 | 1.07690525 | 0.76922631 | 1.15383947 |
| 7 | 1.07691973 | 0.76922993 | 1.15384490 |
| 8 | 1.07692245 | 0.76923061 | 1.15384592 |
| 9 | 1.07692296 | 0.76923074 | 1.15384611 |
| 10 | 1.07692305 | 0.76923076 | 1.15384615 |
| 11 | 1.07692307 | 0.76923077 | 1.15384615 |
| 12 | 1.07692308 | 0.76923077 | 1.15384615 |

**Puede verse una buena aproximación del valor del PageRank real tras solo unas pocas repeticiones**. El número mí­nimo de Iteraciones necesario para que el PageRank sea válido es de 100 en todo el sitio.

También, por medio del cálculo iterativo, **la suma de PageRanks de todas las páginas converge con el número total de Web pages**. El PageRank medio de una página es 1. El PageRank mí­nimo de una página se asigna por (1-d). Por lo tanto, **hay un PageRank máximo para una página que se asigna por dN+ (1-d), donde el número N es el total de páginas**. Este máximo puede ocurrir teóricamente, si todas las páginas enlazan solamente a una página, y esta página también se enlaza a sí­ misma.

**Traducción del artí­culo**: [The PageRank Algorithm](http://pr.efactory.de/e-pagerank-algorithm.shtml)

# Algoritmo HITS

De Wikipedia, la enciclopedia libre

Saltar a: [navegación](http://es.wikipedia.org/wiki/Algoritmo_HITS#mw-head), [búsqueda](http://es.wikipedia.org/wiki/Algoritmo_HITS#p-search)

El **algoritmo HITS** ([acrónimo](http://es.wikipedia.org/wiki/Acr%C3%B3nimo) del [inglés](http://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s) ***H****ypertext* ***I****nduced* ***T****opic* ***S****election*) es un [algoritmo](http://es.wikipedia.org/wiki/Algoritmo) diseñado por [Jon Kleinberg](http://es.wikipedia.org/wiki/Jon_Kleinberg) para valorar, y de paso clasificar, la importancia de una [página web](http://es.wikipedia.org/wiki/P%C3%A1gina_web).

**HITS** usa dos indicadores para hacer esta valoración, definiendo recursivamente cada uno a partir del otro:

1. el *authority*, que valora cuán buena es la página como recurso de información; para su cálculo se usa una suma ponderada de valores *hub* de los enlaces que apuntan hacia esta página.
2. el *hub*, que dice cuán buena es la información que se consigue siguiendo los enlaces que tiene a otras páginas; se calcula como una suma ponderada de valores *authority* de las páginas a las que apunta ésta. Algunas implementaciones del algoritmo también consideran cuánta es la relevancia de las páginas enlazadas.

Pseudocódigo

1 S := semilla de páginas inicial

2 para cada página p en S haga

3 p.aut = 1 // p.aut es el puntaje de autoridad de la página p

4 p.hub = 1 // p.hub es el puntaje de hub de la página p

5 función HubsYAutoridades(S)

6 para iteración 1 a k haga// ejecuta el algoritmo k veces

7 para cada página p en S haga // actualiza todos los valores de autoridad primero

8 para cada página q en p.vecinosQueLlegan haga// p.vecinosQueLlegan es el grupo de páginas que están vinculadas a p

9 p.aut += q.hub

10 para cada página p en S haga // ahora se actualizan los valores de hub

11 para cada página r en p.vecinosQueSalen haga // p.vecinosQueSalen es el grupo de páginas que están vinculadas desde p

12 p.hub += r.aut