

# PageRank de Google explicado: cómo funciona el algoritmo transformó internet

12 de Enero de 2026

## PageRank de Google explicado: cómo funciona el algoritmo transformó internet

Te cuento cómo funciona el algoritmo PageRank de Google, cómo valora la calidad de las páginas web y qué factores influyen en el SEO y posicionamiento de tu sitio

23 September 2025

¿Alguna vez te has preguntado por qué unas páginas aparecen antes que otras en Google?

El posicionamiento mueve miles de millones cada año, y el SEO se ha convertido en una auténtica batalla por la visibilidad.

El año pasado tuve que estudiar a fondo uno de los algoritmos clave detrás de todo esto para un trabajo del máster, y me pareció tan interesante que quería compartirlo por aquí. Se trata del **PageRank**, el famoso algoritmo que hizo despegar a Google y que todavía hoy marca cómo se organiza buena parte de internet.

Quiero compartir contigo de forma sencilla cómo funciona, porque fue la base del éxito de Google y de muchísimas empresas que dependen de internet.

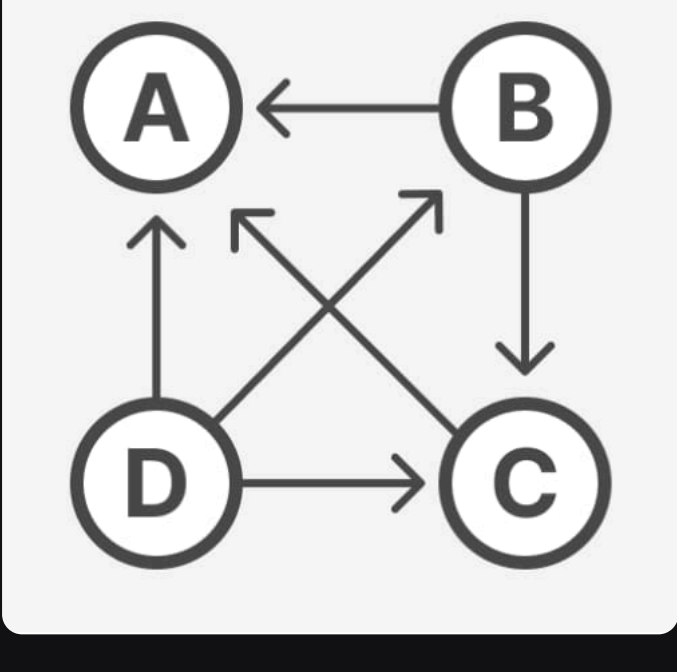
### ¿Qué es el PageRank?

El PageRank no es solo un algoritmo, sino una familia de ellos que fueron desarrollados por los cofundadores de Google, Larry Page y Serguéi Brin. De hecho, el nombre viene del apellido de **Larry Page**. Se patentó en 1999 con el objetivo de mejorar y optimizar los resultados de búsqueda en Google.

Básicamente, el algoritmo de PageRank **valora la calidad de una página web en función de la cantidad de enlaces que recibe y que da**. Se basa en la idea de que las páginas más importantes son las que son enlazadas por muchas otras páginas. Es como si cada enlace fuera un "voto" de confianza de una página a otra.

### La versión original del algoritmo

Para entenderlo de forma sencilla, la versión original del PageRank **entiende el conjunto de la web como un grafo dirigido**, donde cada página es un nodo y cada enlace entre páginas es una arista.



Ejemplo de grafo dirigido

Al principio, todas las páginas tienen la misma puntuación (un 1 en las primeras versiones). La puntuación de una página se reparte a partes iguales entre todas las páginas a las que enlaza. El **PageRank final de una página es la suma de las puntuaciones que le han pasado todas las páginas que la enlazan**.

Luego, se incluyeron variables más complejas, como el **damping factor** ("d"), que es la probabilidad de que un usuario deje de hacer clic en los enlaces de una web. Este factor se estima en 0.85, aunque es algo que solo Google sabe con exactitud.

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

Fórmula original PageRank

Donde PR(A) es el PageRank de la página que queremos calcular, d es el damping factor, PR(i) es el PageRank de las páginas que enlazan a A y C(i) es el número de enlaces salientes de esas páginas.

### Evolución y mejoras

La versión original tenía una gran limitación: era fácil de manipular. Empresas creaban *granjas de enlaces* para comprar miles de enlaces artificiales y mejorar su posicionamiento para engañar a Google.

Para evitar esto, el algoritmo ha ido evolucionando y hoy en día es una familia de algoritmos. Algunas de las **mejoras** más importantes han sido:

**Contexto semántico:** Ahora, el algoritmo tiene en cuenta el contexto y la calidad de la página de origen de un enlace, además de la relación entre ambas páginas.

**RankBrain:** Este es otro algoritmo clave que usa machine learning para analizar las consultas de los usuarios y medir si los resultados los satisfacen, basándose en el tiempo que pasan en las páginas después de hacer clic.

**Etiquetas *nofollow*:** Se incluyeron etiquetas como "nofollow" para que enlaces de publicidad o comentarios (considerados spam) no transfieran PageRank.

### Mi proyecto del máster: PageRank con Spark

Como parte de mi máster, me tocó profundizar en este tema y hacer una implementación práctica. Para ello, usé Scala y la librería GraphX de Spark, que es ideal para trabajar con grafos.

Preparé los datos: Creé un script en Python que generaba un dataset de 100 páginas y 1000 enlaces aleatorios entre ellas, que guardé en un archivo CSV. Mi objetivo no era replicar internet, sino tener una estructura para probar el algoritmo.

```
import csv
import random

numero_webs = 100
numero_enlaces = 1000
enlaces = set()

while len(enlaces) < numero_enlaces:
    web_origen = random.randint(0, numero_webs - 1)
    web_destino = random.randint(0, numero_webs - 1)
    if web_origen != web_destino:
        enlaces.add((web_origen, web_destino))

with open('dataset.csv', mode='w', newline='') as fichero:
    writer = csv.writer(fichero)
    writer.writerow(['web_origen', 'web_destino'])
    for enlace in enlaces:
        writer.writerow(enlace)
```

Para la implementación, con el dataset creado, usé Scala para leer el CSV y crear un grafo con GraphX. Luego, apliqué el algoritmo de PageRank para calcular la puntuación de cada una de las 100 páginas.

**Resultados:** Finalmente, ordené las páginas por su puntuación de PageRank y mostré las 10 con mayor y menor puntuación. La tolerancia que usé fue de 0.001, un valor que permite que el algoritmo itere hasta que la diferencia entre dos cálculos consecutivos sea muy pequeña.

Si tienes curiosidad escríbeme y te mando los scripts en Python y Scala para que puedas replicar los ejemplos y hacer pruebas con más o menos páginas.