CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Informática

Universidad Politécnica de Madrid
Escuela Técnica Superior de
Ingenieros Informáticos

TRABAJO FIN DE GRADO

Modelos y algoritmos del PageRank y su importancia en el funcionamiento de Google

Autor: Domínguez Puerto, Julio
Director: Mateos Caballero, Alfonso

Madrid, Junio 2018

Agradecimientos:

A Sofía, que me descubrió la pasión y el amor real por la vida.
A mi padres y hermanos, por hacer de mi vida un mundo lleno de oportunidades.
A mis amigos, Marcos , Guille y Ventu, siempre fieles y generosos.
A todos los profesores que ponen pasión en su profesión y cambian la vida de tantas personas.

ÍNDICE

1. RESUMEN 5

1.1 Español 5

1.2 English 6

2. INTRODUCCIÓN Y OBJETIVOS 7

2.1 Descripción del proyecto 7

2.2 Experiencia de usuario 10

2.3 Autoridad del sitio web 13

3. DESARROLLO: PAGERANK Y CADENAS DE MARKOV 18

3.1 Rastreo e indexación 20

3.2 PageRank y Cadenas de Markov 22

3.3 Ejemplos y análisis de distribución de PageRank 26

3.3.1 Ejemplo 1 26

3.3.2 Ejemplo 2 – PageRank Sculpting 28

3.3.3 Ejemplo 3 – Dangling node 31

3.4 Presente y futuro del PageRank 34

4. DESARROLLO DE APLICACIÓN 36

4.1 Tecnología empleada 36

4.2 Implementación 36

4.2.1 Base de datos 37

4.2.2 Crawler 38

4.2.3 Cálculo PageRank 42

5. CONCLUSIONES Y LINEAS FUTURAS DE ACTUACIÓN 45

6. BIBLIOGRAFÍA 46

ÍNDICE DE ILUSTRACIONES

FIGURA 1 – RANKING FACTORS 2018. SEMRUSH. 9

FIGURA 2 - ESTRUCTURA SITIO WEB TIPO JERÁRQUICA 14

FIGURA 3 - ESTRUCTURA SITIO WEB TIPO RED-JERÁRQUICA 14

FIGURA 4 – TRÁFICO Y ENLACES DE PERIÓDICOS DE ESPAÑA 15

FIGURA 5 - REGRESIÓN LINEAL - TRÁFICO VS ENLACES 16

FIGURA 6 - RESPUESTA GARY ILLYES, DISTRIBUCIÓN PAGERANK..... 20

FIGURA 7 – BASE DE DATOS “PAGES”..... 37

FIGURA 8 – BASE DE DATOS “LINKS” 38

1. RESUMEN

1.1 Español

En el año 1998 se publicaba el artículo "*The PageRank Citation Ranking: Bringing Order to the Web*" donde Larry Page y Sergey Brin sentaban las bases de lo que sería el buscador de Google:

"La importancia de una página web es un problema inherentemente subjetivo que depende del interés de los lectores, de su conocimiento y de sus inclinaciones. Este artículo describe PageRank, un método para valorar las páginas web de forma objetiva y mecánica".

Las primeras versiones del algoritmo de Google se basaron en el PageRank que se centraba en calcular la importancia de cada web a partir de los vínculos entre ellas, es decir, los enlaces entre URLs que conforman internet.

A lo largo de los años, han entrado en juego muchos más factores a la hora de ordenar los resultados de búsqueda a medida que Google ha sacado actualizaciones de su algoritmo. La tendencia actual se encuentra en darle relevancia a la experiencia del usuario, otorgándole importancia al contenido de calidad, a los sitios webs rápidos, a páginas adaptadas a dispositivos móviles, a entornos con estructuras claras, organizadas y definidas. Con el paso del tiempo, esta tendencia irá aumentando exponencialmente gracias a la ayuda del *machine learning* que permite cada día que pasa simular mejor el comportamiento de los usuarios.

A pesar de esto, el PageRank sigue siendo un factor fundamental. La meta principal de Google es ofrecer el contenido que mejor responda a la intención de búsqueda de los usuarios, esto hace que cada día que pasa tengan más peso todos esos enlaces entre URLs que refuercen el contenido y resuelvan las necesidades.

El presente proyecto tiene como principal objetivo analizar en profundidad el funcionamiento del PageRank, sumado a el desarrollo de un programa que tiene como meta calcular cómo se distribuye la autoridad interna en un sitio web seleccionado, dando soluciones de mejora reales sobre la página web.

1.2 English

In 1998, the paper "The PageRank Citation Ranking: Bringing Order to the Web" was published, in which Larry Page and Sergey Brin wrote the foundations for what would become Google's search engine:

"The importance of a website is an inherently subjective problem that depends on the readers interest, knowledge and inclinations. This article describes PageRank, a method for evaluating web pages in an objective and mechanical way, effectively measuring the human attention and interest directed towards each page".

The first versions of Google's algorithm were based on PageRank, which focused on calculating the importance of each website from the links between them.

Over the years, many more factors have come into play when it comes to sorting search results as Google has released updates to its algorithm. The trend is towards giving relevance to the user experience, giving importance to quality content, fast websites, pages adapted to mobile devices and, organized and defined structures. Soon this trend will increase exponentially with the help of machine learning.

With these new factors, PageRank has lost the supremacy it had over the rest of the factors, but it is still a fundamental factor. Google's primary goal of delivering content that best matches to user's search intentions, means that every day that passes, all those links between URLs that reinforce the content and improve the user experience will have more weight.

The main objective of this project is to analyze in depth the functioning of the PageRank and develop a program to calculate how the internal authority is distributed on a selected website, in order to provide solutions for improvement and better use of resources.

2. INTRODUCCIÓN Y OBJETIVOS

El propósito del presente trabajo es analizar con profundidad uno de los principales factores que usan los motores de búsqueda en internet para ordenar sus resultados, concretamente el conocido algoritmo PageRank desarrollado por Google. Las finalidades principales serán la definición de su funcionamiento y el desarrollo de una metodología que mejore la eficiencia de la distribución de autoridad para aplicarla a proyectos reales. La disciplina que alberga el análisis y la implementación en búsqueda de mejores resultados en los motores de búsqueda se denomina SEO (*Search Engine Optimization*).

2.1 Descripción del proyecto

Una de las razones por las que Google desde un inicio se convirtió en un motor de búsqueda tan eficaz es debido al algoritmo PageRank creado Larry Page y Sergey Brin. El PageRank está completamente relacionado con la estructura de enlaces en la que se basa la conocida *World Wide Web*.

El algoritmo de Google que ordena los resultados de las SERPS (*Search Engine Results Page*) destacó por encima del resto de buscadores porque le otorgó relevancia a los enlaces entre webs, pasando a interpretar internet como un gran grafo. Específicamente, las webs son los nodos y los enlaces entre ellas, las aristas, que son los que permiten llegar a los demás nodos y encontrar la información. En el caso concreto que nos atañe, Google puede rastrear los sitios webs gracias a hacer un *crawling* de los enlaces, luego los indexa y finalmente los clasifica. En esta clasificación de las distintas URLs que ha indexado, el orden lo determinan varios factores, entre los más relevantes, su PageRank, que viene dado exclusivamente del PageRank de las URLs que le apuntan a ésta, siendo éste a su vez transmisor del PageRank que acumula hacia el resto de webs.

El algoritmo para calcular el PageRank se basa en cadenas de Markov, lo que da como resultado que el PageRank es la probabilidad de que un usuario llegue a un sitio de manera aleatoria.

En el SEO no existe una fórmula única para conseguir objetivos y son muchos los factores que importan para conseguir buenos resultados en las SERPS. Estos factores se podrían englobar en dos grandes bloques: experiencia del usuario (muy ligado a los factores denominados *OnPage)* y la autoridad del sitio web.

En el año 2017, la popular herramienta online SEO *SEMrush* realizó un minucioso estudio sobre los factores más relevantes a la hora de ordenar los rankings en las SERPS de Google. Se analizaron más de 600.000 resultados de búsqueda entre muchos conjuntos de datos diferentes a nivel global observando los 100 primeros resultados para cada una de las palabras clave analizadas. Del estudio se sacan conclusiones muy concluyentes y concisas, aunque es cierto, que alguno de estos factores Google ha negado que se usen como parámetros a la hora de ordenar los resultados. Esta postura es muy común por la compañía, por lo que siempre habría que contextualizar e interpretar adecuadamente sus afirmaciones. Los factores que *SEMrush* determina que son de máxima relevancia, en orden de importancia, son:

1. *Visitas directas al sitio web*. Esto demuestra que el sitio es una autoridad en su sector y que el sitio web es útil.
2. *Tiempo en el sitio*. Ligado a una buena experiencia de usuario, dado que los visitantes invierten tiempo a lo largo de la página web.
3. *Páginas por sesión*. Guarda una estrecha relación con una navegación optimizada dentro del sitio web, constatando por tanto una buena acogida por parte de los usuarios y también una buena estructura interna de enlaces donde se distribuye la autoridad del sitio web en beneficio de los intereses del público.
4. *Tasa de rebote*. Acción que se produce cuando un usuario que navega por un sitio web después de haber visto una única página, abandona unos pocos segundos después. En este punto hay dos tipos de casuísticas. La primera y más común esta estrechamente relacionada con un rechazo por parte del usuario, normalmente por una mala experiencia dentro de la página web, esto puede deberse a que la página no se adapte a dispositivos móviles, a una mala velocidad de carga, a un contenido irrelevante, etc. Por otro lado encontramos distintos tipos de búsquedas donde la respuesta a la intención de búsqueda por

parte del usuario es muy concreta, se resuelve en poco tiempo sin apenas navegación. Estas dos situaciones tan dispares para respuestas de usuarios tan distintas hace que la tasa de rebote adquiera una relevancia según el contexto de la intención de búsqueda y respuesta del usuario a lo largo del sector en el que se sitúe una búsqueda determinada.

5. *Número total de dominios de referencia, número total de backlinks, número total de IPs de referencia y número de enlaces de tipo follow.* Factor directamente relacionado con la transmisión de autoridad entre los sitios web. El estudio afirma que la relación es directamente proporcional entre ocupar los primeros resultados en las SERPS con el poseer un mayor número de dominios distintos de referencia y a medida que las *keywords* son de mayor competencia aumenta la proporción de dominios de referencia para encontrarse en primeras posiciones.
6. *Extensión del texto en el sitio web, sitio web seguro usando el protocolo HTTPS, uso adecuado de factores On-page como los son los títulos, y las metadescripciones.*

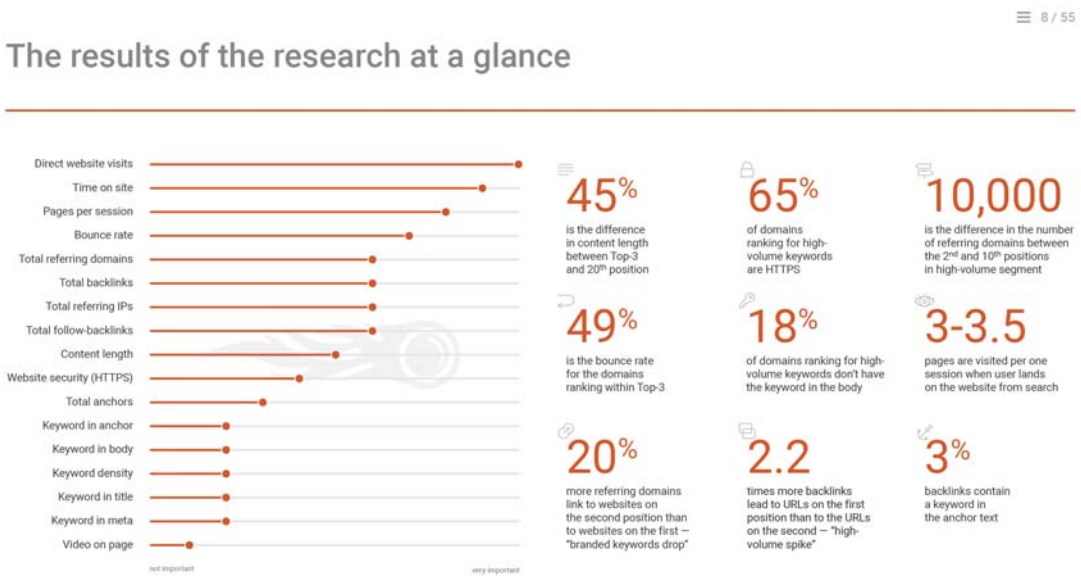


Figura 1 – Resultados de la investigación: Ranking Factors 2018. Fuente: SEMrush.

Como se puede apreciar el protagonismo principal de los factores con mayor peso se centran en el tráfico y su comportamiento en el sitio web, eso le transmite a Google que el dominio posee una elevada calidad y autoridad. Por otro lado, se observa como

los *backlinks* siguen teniendo una incidencia muy elevada en los resultados, impactando directamente en ellos siempre y cuando no solo se potencie este componente.

2.2 Experiencia de usuario

Si echamos la mirada hacia atrás, simplemente a finales del siglo pasado e inicios de los dos mil, se podría afirmar que el SEO era una disciplina simple y con metodologías de trabajo muy básicas pero pronto con el paso de los años se han ido uniendo un amplio abanico de factores que afectan a la organización de los resultados y muchos de ellos asociados al conocido *machine learning* e interpretación de la información relacionada con el comportamiento de los usuarios. Esto ha proporcionado que los resultados mostrados en las SERPS sean los más relevantes, no sólo en términos de contenido, sino acorde a la experiencia del usuario general dentro del sitio web.

Con los nuevos avances del algoritmo de Google la tendencia se encuentra en otorgar mayor visibilidad a aquellos sitios webs que aporten el mayor valor posible y una buena experiencia. Esta nueva realidad en los últimos años ha convertido que el mundo del SEO incorpore en su día a día la experiencia de usuario como uno de sus pilares más fundamentales, llegando a ser el más importante entre todos.

Para medir la calidad y el fácil uso de un sitio se pueden realizar varias preguntas que encaminen a una solución: ¿Cómo es de fácil de navegar la página web? ¿El contenido es relevante y atractivo para los visitantes y les hace interactuar dentro de la página y mostrar interés? ¿Es segura la página, rápida en velocidad de carga y adaptada para los teléfonos móviles?

Puede sonar extraño que a día de hoy siga teniendo mucha presencia en las distintas páginas el hándicap de una navegación compleja y poco intuitiva. Un sitio web ha de tener una estructura lógica y útil tanto para que los usuarios puedan interactuar de la mejor forma posible como para que los robots de los distintos buscadores, puedan rastrear adecuadamente el conjunto de URLS que forman el sitio web y comprender de una forma eficiente el mapa que conforma. Existen distintos tipos de estructuras en la actualidad que afectan en gran medida a la comprensión del sitio web, siendo las más comunes la estructura jerárquica, y de tipo red-jerárquica. Este es uno de los nexos más

fuertes entre la experiencia del usuario y la autoridad de un sitio web, donde confluye la distribución de la autoridad dentro de una web y el flujo de la interacción de los usuarios dentro de un sitio. Más adelante se explicarán las características de cada uno de los tipos de estructuras.

Hoy en día la gran mayoría de los usuarios no entran al sitio web a través de su página de inicio, lo que significa que ha de ser consistente a lo largo de todas sus URLS indexables por los buscadores, por lo que las distintas secciones de la página adquieren mayor relevancia, cobrando especial importancia el trabajo de personas que ejercen como UX(*user experience*) como los UI(*user interface*) que consiguen hacer entornos intuitivos, eficientes y atractivos.

Las distintas acciones que tomen los usuarios dentro de una página web serán un rastro de señales que Google tomará para ordenar sus resultados.

La velocidad del sitio web, también conocida como WPO (*Web Performance Optimization*) es un factor de clasificación de los resultados y seguirá siendo más importante a partir de ahora dado que Google ha anunciado que a partir del mes de Julio de 2018 la velocidad será un factor de posicionamiento web para las búsquedas móviles, así dice su directiva: "La *Actualización de velocidad*, como así la llamamos, sólo afectará a las páginas que ofrecen la experiencia más lenta a los usuarios y sólo afectará a un pequeño porcentaje de las consultas. Aplica el mismo estándar a todas las páginas, independientemente de la tecnología utilizada para construir la página. La intención de la consulta de búsqueda sigue siendo una señal muy fuerte, por lo que una página lenta puede tener un alto rango si tiene un gran contenido relevante."[1]

Por lo tanto no solo adquiere una importancia que un sitio web cargue velozmente, sino que su experiencia en versión móvil también lo sea y lo aparente. La conclusión una vez más sobre esta última directiva de Google es que ningún factor marca la diferencia, pero todos tienen su relevancia. Frecuentemente los anuncios de Google van asociados a la publicación de herramientas para mejorar las páginas web en pro de las nuevas directivas. En este caso encontramos la herramienta online *Google PageSpeed*

[1] https://webmasters.googleblog.com/2018/01/using-page-speed-in-mobile-search.html

Insights[2] y también *Lighthouse*[3], que permiten detectar incidencias de rendimiento relacionadas con la velocidad del sitio web, ya sea en versión *desktop* o en versión móvil.

En los últimos años el porcentaje del tráfico móvil se ha elevado hasta alcanzar un 70% de media dejando así en un segundo puesto al tráfico generado desde portátiles o ordenadores de mesa. Este hecho ha supuesto que Google le de máxima relevancia a la versión móvil tomando varias acciones como han sido el denominados *Mobile-first idexing*[4] y *Speed Update*[5].

El *Mobile-first indexing* ha entrado en vigor en marzo de 2018 con la principal intención de organizar el índice de resultados de Google a través del rastreo de los sitios webs con un *boot* específico para dispositivos móviles. Por lo tanto, lo que se pretende es simular la navegación de un sitio web a través de un dispositivo móvil pudiendo así comprobar el tamaño correcto de la letra, que los enlaces y botones se encuentren en las dimensiones y con el espacio correcto y que todo el contenido se visualice correctamente en un navegador móvil.

El *Speed Update* anteriormente mencionado, se centra exclusivamente en los dispositivos móviles.

Dentro de la experiencia del usuario entra el contenido. Un buen contenido será aquel que responda la intención de los usuarios y hará en la mayoría de los casos que los usuarios permanezcan más tiempo en el sitio web y naveguen alrededor de él, aumentando también la posibilidad de que los contenidos sean enlazados desde otros sitios, lo que le otorgará más autoridad al dominio.

Por último, la seguridad de los sitios webs bajo el protocolo HTTP. En los últimos anuncios de Google ha publicado una nueva advertencia, “A secure web is here to stay”[6], indicando que a partir del mes de julio de 2018 todos aquellos sitios web que no

[2] https://developers.google.com/speed/pagespeed/insights/
[3] https://developers.google.com/web/tools/lighthouse/run/
[4] https://webmasters.googleblog.com/2018/03/rolling-out-mobile-first-indexing.html
[5] https://webmasters.googleblog.com/2018/01/using-page-speed-in-mobile-search.html
[6] https://security.googleblog.com/2018/02/a-secure-web-is-here-to-stay.html

contengan el protocoló HTTP bajo el cifrado HTTPS, serán marcados como sitios web no seguros, provocando así una mala imagen de cara los usuarios aumentando muy probablemente la tasa de rebote.

2.3 Autoridad del sitio web

La autoridad de un sitio web, engloba dos escenarios, la distribución de la autoridad de un sitio web a nivel interno y por otro lado la autoridad que tiene el sitio web a nivel externo.

La distribución de la autoridad a nivel interno es la distribución del PageRank de un sitio web que vendrá determinado por la arquitectura interna en la que se base. Una buena organización supondrá división optimizada de la autoridad interna, dándole relevancia a aquellas partes del sitio web que lo requieren. En la actualidad encontramos dos tipos de arquitecturas habitualmente usadas que ayudan a una correcta distribución: jerárquica o de tipo red-jerárquica.

La estructura jerárquica, véase Figura 2, guarda muchas similitudes a la estructura de un árbol, donde la página de inicio es el nodo raíz de todo el árbol, a partir de ahí se crean distintos niveles, primer nivel (Nivel 1, por ejemplo: midominio.com/categoria1), segundo nivel (Nivel 2, por ejemplo: midominio.com/categoria1/subcategoria1), tercer nivel (Nivel 3, midominio.com/categoria1/subcategoria1/articulo), etc. Estas divisiones del sitio web son las que a día de hoy se conocen como *clusters*, agrupaciones de contenido estrictamente relacionados aumentando la densidad de contextualización de un grupo de URLs, aumentando así la relevancia entorno a una respuesta de usuario concreta.

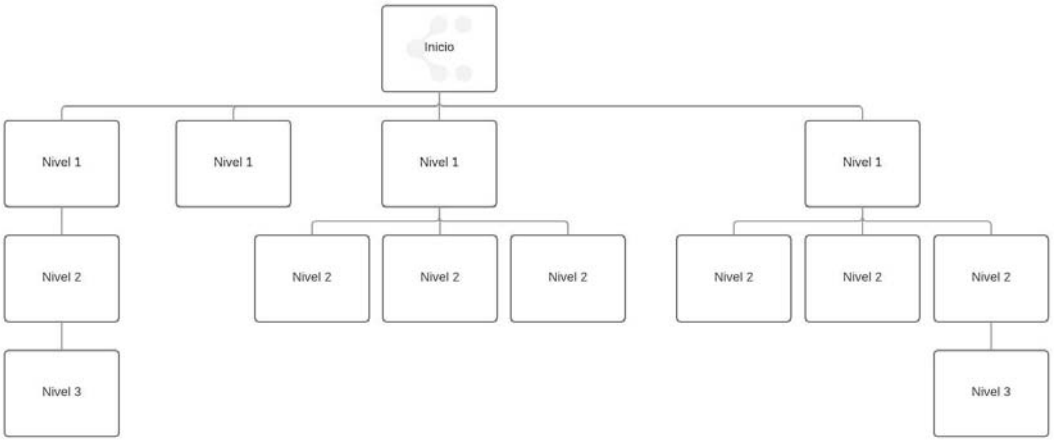


Figura 2 - Estructura sitio web tipo jerárquica

Este tipo de diseño de URLs agrupadas favorece a los lectores a la comprensión por parte de los usuarios, conociendo la organización del sitio web y donde se encuentran, sabiendo que la información más general se encuentra en los niveles superiores.

La arquitectura de tipo red-jerárquica, véase Figura 3, guarda mucha relación con la analizada anteriormente, con el añadido de que existen enlaces entre los distintos *clusters* siempre que el contenido guarde relación, con la intención de mejorar la interacción por parte de los usuarios, ofreciendo la posibilidad de una navegación más rica y tráfico a lo largo de las distintas secciones. Sin perder la lógica y segmentación lograda al establecer las jerarquías.

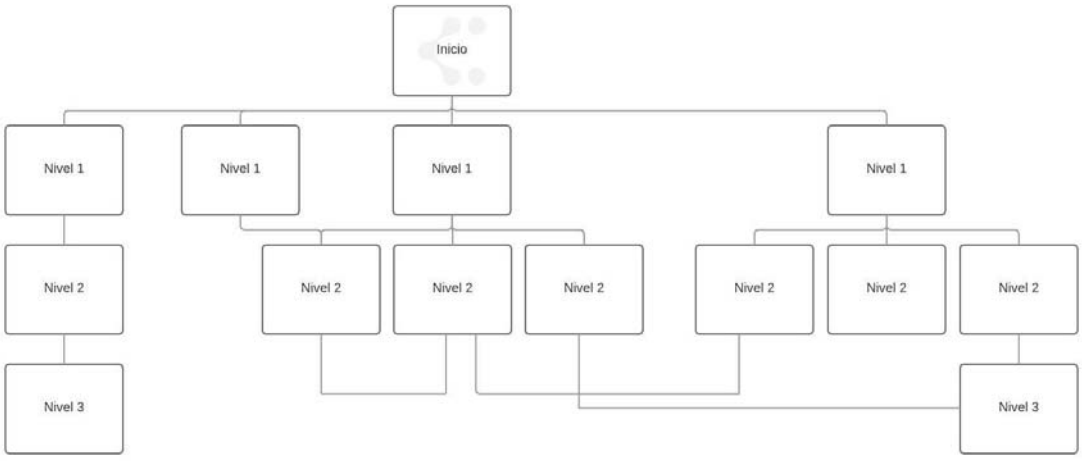


Figura 3 - Estructura sitio web tipo red-jerárquica

En ambos casos y comúnmente en el segundo más, se consigue una tasa de rebote inferior. Un sitio web más rico y funcional, por lo que acaba repercutiendo en las señales enviadas a Google como positivas a nivel de UX (*User Experience*), aquí es donde convergen la distribución de autoridad y la experiencia del usuario.

También cabe reseñar que la existencia de una arquitectura bien clasificada ayuda notablemente a la comprensión del sitio web por parte de los robots de los buscadores, haciendo así su trabajo más fácil y exprimiendo al máximo los recursos destinados para su rastreo.

En lo referente a la autoridad a nivel externo, que es como realmente Google clasifica los distintos nodos existentes en internet, existe una correlación directa entre la visibilidad de cada una las URLs de un sitio web y su tráfico y el número de *backlinks* que recibe un dominio.

Para demostrar este hecho se ha realizado un estudio de 10 periódicos de España con un elevado tráfico y el número de *backlinks* entrantes al dominio raíz (estimación de tráfico al mes y *backlinks* obtenidos mediante la herramienta *Ahrefs*).

Periódico	**Tráfico estimado *Ahrefs***	**Enlaces *Ahrefs***
El País	36	27.5
Eldiario.es	3	4.2
20 minutos	7.6	4.9
La Vanguardia	16.9	5.4
El Mundo	17.2	16.3
El Confidencial	7.7	2.4
ABC	7.1	2.5
La Voz de Galicia	2.2	1.8
La Razón	0.7	2.2
El Periódico	5.3	13.3

Figura 4 – Tráfico y enlaces de periódicos de España

*datos en millones

Con esta información se procede a obtener el coeficiente de correlación de Pearson, una medida de relación lineal entre variables aleatorias cuantitativas, que otorga una correlación de forma independiente a la escala de medida de variables.

Coeficiente correlación Pearson

$$r = \frac{S_{XY}}{S_X S_Y}$$

Desviación típica

$$S_X = \sqrt{\frac{\sum_i (Xi - \bar{X})^2}{n}}$$

Covarianza

$$s_{xy} = \frac{\sum x_i \cdot y_i}{n} - \bar{x} \cdot \bar{y}$$

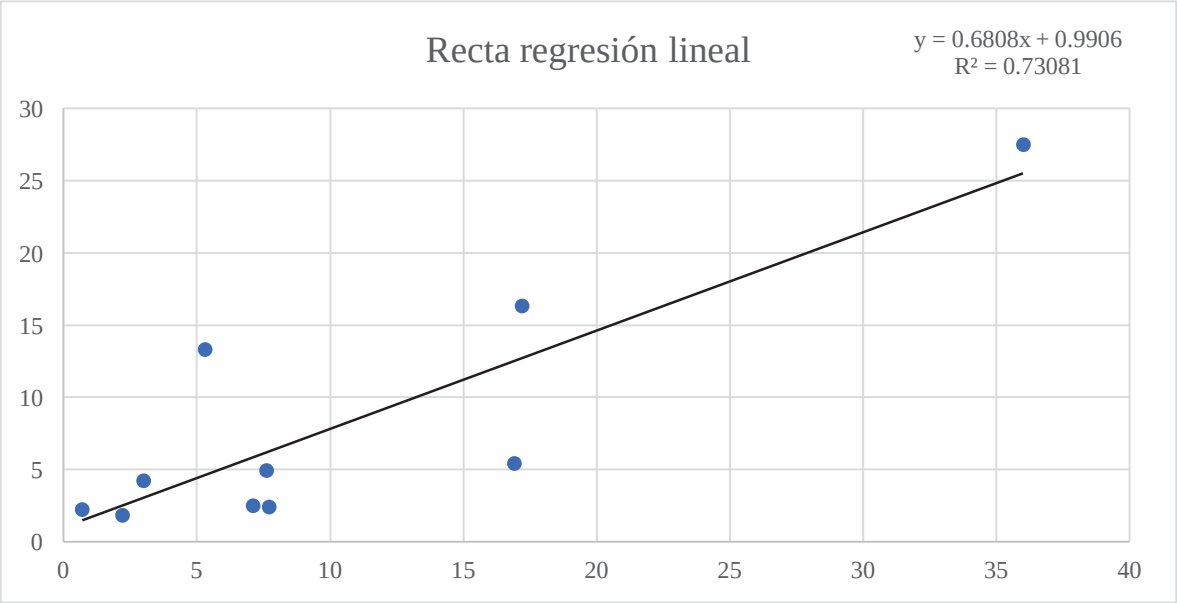


Figura 5 - Regresión lineal - Tráfico vs Enlaces

S_{xy}	68.8895
S_x	10.05962723
S_y	8.010649162
r	0.854876607

El valor del coeficiente de correlación, r = 0.854876607, al estar muy próximo a a uno porque la vinculación entre ambos hechos es muy fuerte, lo que nos lleva a afirmar que existe una correlación de tal forma que cuando aumenta una aumenta la otra.

Con esta correlación queda ilustrada la fuerte vinculación entre tráfico de un sitio web y los enlaces entrantes que recibe, un indicio más que demuestra que sí son un factor esencial de posicionamiento en los resultados de búsqueda.

El presente proyecto tiene como principal función conocer a fondo el funcionamiento del PageRank e intentar dar respuesta y soluciones a proyectos reales en la distribución de autoridad dentro de un sitio.

El Capitulo 2 consiste en un análisis minucioso de la fórmula del PageRank y el desarrollo de ejemplos que simulen posibles casos reales.

El Capitulo 3 presenta y explica el programa desarrollado para poder analizar la autoridad de sitios webs reales.

Por último el Capitulo 4 es una exposición de todos los aprendizajes obtenidos gracias al desarrollo del proyecto y una puerta abierta a posibles líneas de investigación.

3. DESARROLLO: PAGERANK Y CADENAS DE MARKOV

El PageRank es uno de los factores principales en la historia del SEO, teniendo aún en la actualidad una relevancia muy importante en el orden de los resultados de búsqueda.

Concebimos internet como un gran grafo, donde los nodos serán las URLs y las aristas los enlaces que unen estas URLs. Por lo tanto, serán los nodos donde se aloje la información y las aristas los hipervínculos que serán los "caminos" que permiten llegar a esa información. Debito a esto, una URL que no reciba enlaces no tendrá mucha relevancia ya que se limitan notablemente las vías de acceso a su información.

Se debe diferenciar 2 tipos de PageRank, el primero esta asociado al dominio en su conjunto, PageRank externo, siendo reflejo de la autoridad que tiene con respecto al resto de páginas web en la red. El segundo tipo sería a nivel interno, que ilustra la distribución de autoridad por cada una de sus URLs internas de un sitio web. Este último caso se calcula a partir de la obtención del grafo formado por la estructura interna de URLs de un dominio.

El PageRank se calcula mediante una fórmula que representa la distribución de una probabilidad en un proceso estocástico discreto. Cabe destacar que Google lo aplica sobre todo los nodos que conforman internet, esto no quiere decir que no se pueda aplicar simplemente sobre un único dominio, y obtener resultados concluyentes y útiles.

$$PR_A = (1 - d) + d * [(PR_{T1} / C_{T1}) + \ldots + (PR_{Tn} / C_{Tn})]$$

siendo:

- PR_A = PageRank de la página *A*
- *d* = *Damping factor* (0.85)
- *Tn* = Página *n* que apunta a *A* (citaciones[7])
- PR_{Tn} = PageRank de la página *n* que cita a *A*

[7] En 1998 en el artículo académico "The Anatomy of a Large-Scale Hypertextual Web Search Engine" a aquellos sitios que apuntan a una URL en concreto se les denominan citaciones: http://infolab.stanford.edu/~backrub/google.html

- C_{Tn}= Número de enlaces salientes de la página *n*.

La anterior fórmula representa una distribución de probabilidad, donde una variable aleatoria es una función que asigna a cada suceso definido sobre la variable aleatoria la probabilidad de que dicho suceso ocurra. En este caso, se puede afirmar que PageRank es uno, tanto si hablamos de todo internet como si nos referimos al conjunto de nodos formado por un único sitio web.

El valor del *damping factor* publicado en el documento académico[8] del año 1998, tiene asignado un valor de 0.85, con la intención de cubrir todos aquellos nodos en internet que se encuentran aislados del resto y no tienen ningún camino conector. Es muy probable que Google haya actualizado este valor tras el transcurso de más de 20 años, pero lo relativamente importante de este valor, no es el valor en sí, sino como condiciona el resto de resultados.

¿Cómo calcula Google exactamente el PageRank? Google utiliza todas las URLs que tiene en su índice y es ahí donde radica su máximo éxito, consiguiendo calcular el PageRank de todos los nodos al mismo tiempo. Este factor es muy importante dado que el PageRank adquiere la misma importancia tanto si nos referimos a nivel interno como externo. Información contrastada gracias a la respuesta de Gary Illyes que ocupa el cargo de *Webmaster Trends Analyst* en Google, siendo uno de los personajes públicos con más relevancia en las distintas actualizaciones del algoritmo.

[8] “The Anatomy of a Large-Scale Hypertextual Web Search Engine”

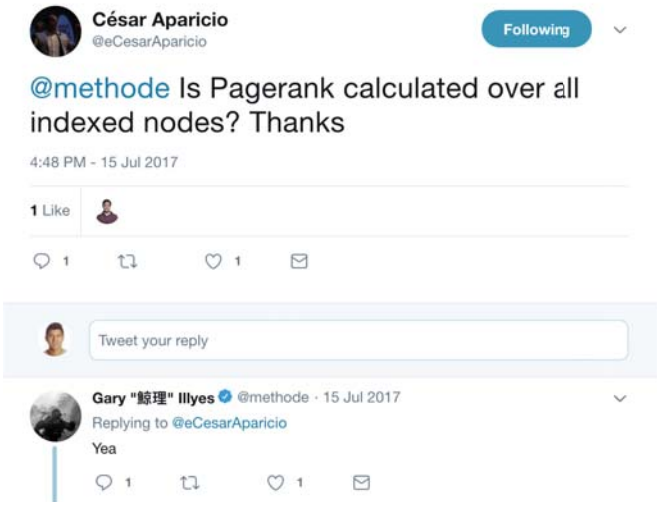


Figura 6 - Respuesta Gary Illyes, distribución del PageRank

3.1 Rastreo e indexación

Los enlaces entre páginas web son uno de los factores principales de posicionamiento web, se puede ver internet de tal forma que las URLs son los nodos que contienen información y las aristas los enlaces entre ellas que permiten el acceso y navegación. ¿Cómo rastrea Google por lo tanto los sitios web? Google realiza un *crawling* de todos los enlaces existentes en internet, calculando el PageRank sobre el total, para a continuación realizar una indexación y finalmente una clasificación del conjunto de los resultados encontrados.

Para llevar a cabo el rastreo, que es el primer paso de todos, son necesarias las conexiones entre URLs, los enlaces.

Los enlaces en su semántica guardan una información muy relevante, que supone una distribución correcta del *link juice* (distribución de PageRank entre los distintos enlaces existentes en una URL) y del *crawl budget* (tiempo destinado de rastreo a un sitio web).

Los enlaces a nivel semántico, se pueden clasificar de dos maneras, follow o nofollow, mediante el etiquetado HTML (por ejemplo: <a href="https://ejemplo.com"

rel="follow">). Los enlaces follow indican al GoogleBoot[9] que siga el enlace, en cambio en el caso de los enlaces con la etiqueta nofollow bloquea el rastreo a través de ese enlace.

El *link juice* es el PageRank que se puede distribuir a través de una URL, que se reparte entre todos los enlaces existentes en un determinado nodo. El uso del etiquetado afecta de forma directa a esta distribución, dado que todos los enlaces internos, sean de tipo follow o nofollow, hacen que el *link juice* se reparta de una manera equitativa por cada uno de los enlaces, con la única diferencia que los enlaces nofollow pierden la transmisión de la autoridad, por lo que se estaría desaprovechando y perdiendo parte de la transmisión de PageRank posible. En consecuencia, serán casos muy específicos en los que sea beneficioso usar la etiqueta nofollow en enlaces a nivel interno.

Google nos recomienda una serie de casos en los que sí aplicar esta etiqueta: cuando el contenido externo enlazado no es confiable en su totalidad, cuando son enlaces de pago (anuncios, enlaces de afiliación[10], etc.) y priorización del rastreo bloqueando el acceso a URLs poco relevantes para el robot de Google siempre y cuando se tenga en cuenta la pérdida de *link juice*. A partir de esta circunstancia, surgen diferentes técnicas como la ofuscación de enlaces, para tratar de evitar el desaprovechamiento de *link juice* a nivel interno, más adelante se ahondará en este tema.

El *crawl budget* o presupuesto de rastreo es el tiempo que destina Googleboot a rastrear un sitio web. Este presupuesto de rastreo vendrá determinado por las dimensiones del grafo que conforme un dominio y de su autoridad a nivel externo. Para sitios web muy grandes en ocasiones el *boot* tiene muchas complicaciones para rastrearlo en su totalidad, por lo que será importante hacer uso del archivo .htaccess[11] para bloquear una o varias rutas del dominio, con la intención de que se rastree el contenido relevante para los usuarios, siempre teniendo en cuenta que esto puede afectar

[9] Robot de búsqueda usado por la empresa Google que realiza el rastreo de los sitios web para su posterior indexación
[10] Enlaces que referencian a productos o servicios de otros sitios web con la intención de obtener una comisión por venta, registro o clic conseguido.
[11] Archivo de configuración de directorios de Apache.

a la autoridad de los nodos del sitio web, dado que también bloqueará la transmisión de autoridad.

Tras el rastreo del sitio indexa las distintas URLs marcadas para ese cometido y crea un índice, que funciona de forma similar al de un libro, donde las URLs quedan ordenadas y clasificadas en función al PageRank calculado justo en el momento anterior.

3.2 PageRank y Cadenas de Markov

El PageRank se puede definir como un ranking de URLs que tiene en cuenta el número de enlaces que recibe cada una de las URLs, teniendo un peso muy importante en el cálculo final de este ranking desde donde se reciben estos enlaces.

El PageRank de un nodo/URL depende exclusivamente de los PageRanks de los nodos/URLs que le enlazan de forma directa, tanto nodos/URLs internos o externos. Se calcula por lo tanto a partir de todos los nodos/URLs visibles en internet. Hablamos de nodos visibles para todos aquellos que reciben un enlace desde un nodo que ha podido ser rastreado previamente, lo que significa que ese nodo también esté enlazado desde algún otro lado.

El algoritmo del PageRank funciona en base a lo que es conocido comúnmente como cadenas de Markov. Las cadenas de Markov son un proceso estocástico (variables aleatorias que evolucionan en función de otras variables aleatorias) discreto (divisible en un número finito) en el que la probabilidad de un evento depende del inmediatamente anterior. Lo que significa en este caso que el PageRank de una URL depende exclusivamente de las URLs que le apuntan de forma directa y no de otras.

Una cadena de Markov es homogénea cuando la probabilidad de pasar del estado *i* al *j* no depende del instante de tiempo en el que nos encontramos:

$$P(X_{n+1} = j \mid X_n = i) = P(X_2 = j \mid X_1 = i).$$

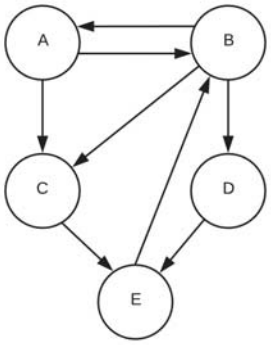
De esta forma las probabilidades de transitar entre un estado *i* a otro *j* se definen como la matriz de transición *T*, donde:

$$T_{ij} = P(X_{n+1} = j \mid X_n = i).$$

Una distribución estacionaria es una distribución de probabilidad donde la probabilidad *p*, dentro del conjunto de estados posibles, es: $pT = p$. El punto importante en una distribución estacionaria, se encuentre en que la cadena de Markov siempre tendrá a converger hasta dicha distribución, sin guardar ningún tipo de relación con el marco inicial en el que se encuentre o se considere.

Finalmente, para calcular la matriz estacionaria asociada a la distribución de PageRank, es necesario elevar la matriz de transición inicial a un número determinado de repeticiones hasta que los valores de la matriz se mantengan homogéneos a pesar de añadir más interacciones. De esta forma, se obtiene una representación real de la relevancia que tiene un nodo indiferentemente de que un usuario llegue a una URL u otra dentro de un sitio web. Quedando como resultado final la "probabilidad" de que un usuario llegue a un determinado nodo. A este valor final, habría que sumarle la aplicación del *damping factor*, que alterará el valor del peso final de todos los nodos.

Supongamos que existe una página web formada por 5 URLs y que su enlazado interno viene representado por el siguiente grafo:



Donde la distribución de autoridad[12] en una URL *i* se define calculándose de forma recursiva e instantánea sobre todos los nodos, siguiendo la fórmula:

$$A_i = ((A_{T1} / C_{T1}) + \ldots + (A_{Tn} / C_{Tn})).$$

[12] Autoridad de una URL: peso de una URL con respecto a otras sin aplicar el valor del *damping factor*, que dará a posteriori el valor real del PageRank.

siendo:

- A_i = la autoridad de la página *i*.
- $\{A_{Tn}\}$ = conjunto de URLs que enlazan a la página *i*.
- A_{Tn} = la autoridad de la página T_n.
- C_{Tn} = número de páginas enlazadas desde T_n.

Dado que este proceso es recursivo, en un estado inicial, se asigna una autoridad uniforme para todas las URLs, siendo: $A_i^0 = 1 / n$ donde i = 1, 2, 3…n. A posteriori, se itera de forma recursiva sobre todos los nodos con la fórmula anteriormente expuesta. Este proceso, se puede ver como una cadena de Markov en el que la matriz de transición sería:

$$\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Al ser una matriz estocástica, todas las filas han de sumar 1.

Una vez obtenida la matriz de transición, habrá que realizar un número suficiente de iteraciones hasta que deje la matriz deje de converger y se mantenga estable, resultando una matriz final, denominada estacionaria. Donde se otorga a cada nodo la probabilidad de que sea visitado. Estas iteraciones lo que van hacer es conseguir representar la relevancia de los nodos independientemente de que un usuario llegue a uno u otro. Es importante especificar que la distribución de probabilidad ha sido rellenada por filas.

0.1111	0.3333	0.1667	0.1111	0.2778
0.1111	0.3333	0.1667	0.1111	0.2778
0.1111	0.3333	0.1667	0.1111	0.2778
0.1111	0.3333	0.1667	0.1111	0.2778
0.1111	0.3333	0.1667	0.1111	0.2778

Finalmente, sobre este ranking final de autoridad se aplica el *damping factor*, valor que simula la probabilidad de que exista un enlace proveniente desde un nodo totalmente aislado, para ello se considera una matriz *D* constante con valor a 1/n, siendo *n* el número de nodos que forman el grafo y donde *A* es la matriz estacionaria:

$$PR = 0.85D + 0.15A$$

0.1244	0.3133	0.1717	0.1244	0.2661
0.1244	0.3133	0.1717	0.1244	0.2661
0.1244	0.3133	0.1717	0.1244	0.2661
0.1244	0.3133	0.1717	0.1244	0.2661
0.1244	0.3133	0.1717	0.1244	0.2661

Por lo tanto el vector estacionario resultante donde se asocia el valor definitivo de PageRank a cada nodo es:

$$\pi^{PR} = (0.1244, 0.3133, 0.1717, 0.\ 1244, 0.2661).$$

Las cadenas de Markov ofrecen la posibilidad de hallar la probabilidad de encontrarnos en un nodo/URL al azar sin haber tenido en cuenta sucesos anteriores. Por lo que en función de los enlaces entrantes y la relevancia que tengan estos enlaces que recibe, este sitio tiene más probabilidades de ser visitado por un usuario al azar.

Es muy común que dentro de muchas estrategias SEO se promulgue la obtención de enlaces en grandes cantidades, debido a que cuantos más caminos lleguen a un sitio web aumentará la probabilidad de que éste sea encontrado. Esto hace que se haga *spam* en

muchos de los casos o dicho de otra forma, un uso inadecuado de creación de enlaces sin pensar aportar valor al usuario. Este tipo de prácticas es perseguida por el algoritmo de Google gracias al *machine learning* que trabaja siempre en beneficio de la experiencia de los usuarios siendo capaz de detectar qué enlaces realmente no son "forzados". Más adelante, se desarrollará de forma más extendida esta situación y como Google define la relevancia de los distintos enlaces, otorgándoles a unos mayor importancia y los que no son considerados naturales como irrelevantes o inexistentes.

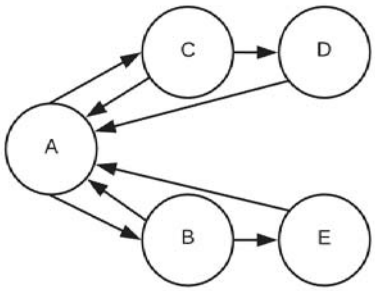
Las URLs que contengan información poco útil, duplicada o con *thin content* estarán diluyendo significativamente el PageRank dentro de un sitio web, lastrando por lo tanto a aquellas que sí contienen una información rica y útil para los usuarios. Entonces sí se consigue orientar y optimizar al máximo esta distribución de autoridad dentro de un sitio web, éste se verá notablemente beneficiado, dado que las URLs más importantes tendrán una alta autoridad que es lo que se persigue en todos los proyectos.

En definitiva, podemos resumir que el PageRank es la probabilidad de que un usuario llegue a una página al azar, por lo que mayor probabilidad significará mayor PageRank.

3.3 Ejemplos y análisis de distribución de PageRank

A continuación se presentan tres ejemplos, ejemplificando casos singulares como el *Pagerank Sculping* con el uso de enlaces nofollow o el *Pagerank death*.

3.3.1 Ejemplo 1



$$\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

El ejemplo 1 representa el enlazado interno de un único dominio donde se representa una estructura jerárquica y donde la página de inicio recibe enlaces desde todas las URLs, hecho que acostumbra a ser muy habitual en las páginas web. Tras aplicarle el *damping factor* a la matriz estacionaria:

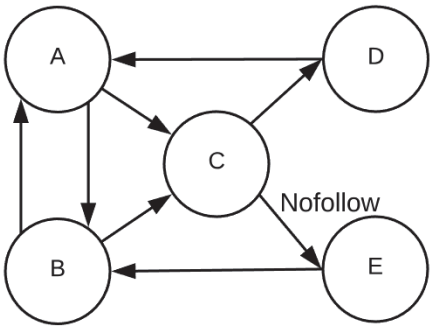
$$\begin{pmatrix} 0.2300 & 0.2000 & 0.2000 & 0.1850 & 0.1850 \\ 0.2300 & 0.2000 & 0.2000 & 0.1850 & 0.1850 \\ 0.2300 & 0.2000 & 0.2000 & 0.1850 & 0.1850 \\ 0.2300 & 0.2000 & 0.2000 & 0.1850 & 0.1850 \\ 0.2300 & 0.2000 & 0.2000 & 0.1850 & 0.1850 \end{pmatrix}$$

π^{PR} = (0.2300, 0.2000, 0.2000, 0.1850, 0.1850).

El ejemplo 1 es una clara representación de un sitio web con una estructura de URLs jerárquica muy definida y así lo demuestra el PageRank de sus nodos, donde cada uno de los niveles del sitio web tiene un PageRank distinto. Lo más común en una página web es que la página de inicio sea la que mayor PageRank tenga, dado que comúnmente recibe enlaces desde el header[13] o footer[14]. Y a medida que se van bajando niveles la autoridad disminuye.

[13] Es un elemento HTML que representa una sección donde se determina la navegación del sitio web. Puede contener elementos de encabezado, como un logo o menú.
[14] Es un elemento HTML que representa el pie de página. Típicamente contiene información del autor, menú de navegación del sitio web, enlaces relacionados, etc.

3.3.2 Ejemplo 2 – PageRank Sculpting



$$\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

El Ejemplo 2 representa un caso de uso del enlace de tipo nofollow a nivel interno. Este hecho condiciona la transmisión de PageRank, diluyendo la transmisión de autoridad por ese enlace, afectando la distribución de la autoridad a lo largo de todo el grafo. Es por este motivo que la casilla que representa la casilla de la conexión *C-E* vale 0 y por lo tanto la suma de la fila *C* no es igual a 1.

Este tipo de enlazado interno es conocido como *PageRank Sculpting*. Esta práctica fue muy utilizada años atrás (hay quien sigue usándola incorrectamente) por SEOs con la intención de manipular el flujo de autoridad interna de un dominio.

En 2009, Matt Causs, quién fue director del departamento contra el spam en web de Google, dejó claro en una entrevista que esta práctica es incorrecta: “Entonces, ¿qué sucede cuando tienes una página con diez puntos de PageRank y diez enlaces salientes, y cinco de esos enlaces son nofollow? Originalmente, los cinco enlaces sin nofollow habrían fluido dos puntos de PageRank cada uno (en esencia, los enlaces no seguidos no contaban hacia el denominador al dividir PageRank por el grado de la página). Hace

más de un año, Google cambió el flujo del PageRank para que los cinco enlaces sin nofollow fluyeran un punto de PageRank cada uno.”

La introducción de este tipo de enlaces dentro de la estructura de un sitio web, supone la transformación de la transmisión de autoridad tanto al nodo que enlaza directamente, en este caso sería el nodo E, dado que el enlace que recibe desde C no le transmite autoridad, como indirectamente también al nodo B porque recibe un enlace desde D y en consecuencia una cambio global de la autoridad de los nodos relacionados.

Si consideramos este mismo caso, pero siendo la URL entre C y E sea de tipo follow, el resultado final de la matriz tras aplicar la fórmula del PageRank sería:

$$\begin{pmatrix} 0.2425 & 0.2425 & 0.2425 & 0.1363 & 0.1363 \\ 0.2425 & 0.2425 & 0.2425 & 0.1363 & 0.1363 \\ 0.2425 & 0.2425 & 0.2425 & 0.1363 & 0.1363 \\ 0.2425 & 0.2425 & 0.2425 & 0.1363 & 0.1363 \\ 0.2425 & 0.2425 & 0.2425 & 0.1363 & 0.1363 \end{pmatrix}$$

Sin el enlace nofollow: π^{PR} = (0.2425, 0. 2425, 0. 2425, 0.1363, 0.1363).

En el caso de la existencia nofollow, la matriz de transición inicial que representa el grafo contiene una fila en la que la suma no es igual a 1(en este ejemplo, igual a 1/2), al realizar la distribución estacionaria sobre la matriz, acaba colapsando. Esta situación significa que la matriz que representa la distribución estacionaria del grafo converge tendiendo a 0 como resultado final en todas sus posiciones.

Para solucionar esta circunstancia, se remplazan todas las casillas de la fila (la fila relacionada con el enlace nofollow, es decir, la fila de *C*) que valen 0, por el valor restante hasta que la fila sume 1 dividido por el número de nodos en esa fila que valgan 0. A continuación la matriz de transición resultante:

0	1/2	1/2	0	0
1/2	0	1/2	0	0
1/8	1/8	1/8	1/2	1/8
1	0	0	0	0
0	1	0	0	0

Este ajuste esta justificado dado que representa el modelado del comportamiento de un usuario que navega por la red. Si la probabilidad de que continúe por alguno de los enlaces disponibles no es igual a 1, habrá que distribuir el porcentaje restante de probabilidad entre todos los nodos disponibles de forma equitativa. Este modelo es la propuesta más solida hasta el momento para esta singularidad, tomando los *N* nodos restantes de forma aleatoria con la misma probabilidad.

Es importante recordar en este punto que Google ve todo internet como un único nodo, por lo que realmente los *N* nodos en este caso serían todos aquellos que conforman el grafo de internet, lo que haría que se diluyese más aún, tendiendo a cero. Este concepto guarda muchas similitudes con el *damping factor*.

La matriz final de PageRank asociada al grafo conformado por el enlace nofollow:

0.2819	0.2189	0.2819	0.1559	0.0615
0.2819	0.2189	0.2819	0.1559	0.0615
0.2819	0.2189	0.2819	0.1559	0.0615
0.2819	0.2189	0.2819	0.1559	0.0615
0.2819	0.2189	0.2819	0.1559	0.0615

Con el enlace nofollow: π^{PR} = (0.2819, 0.2189, 0. 2819, 0.1559, 0.0615).

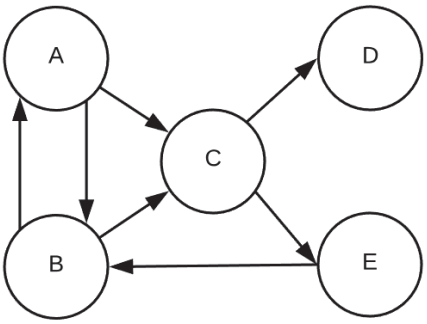
Si se compara el la distribución del PageRank entre las distintas URLs, se puede observar como pasan los nodos *A* y *C* a ser los de mayor autoridad de manera muy significativa, disminuyendo de manera indirecta la autoridad de *B*.

Esta circunstancia, en un proyecto real, aumenta notablemente la complejidad de conocer con exactitud la distribución de la autoridad a nivel interno. Esta complejidad aumentada se debe a que se incremente el número de factores aleatorios, teniendo como consecuencia final la disolución de la autoridad entre más nodos.

Debido a estas características, se recomienda evitar el uso de esta práctica para enlazado interno. En muchos casos se quiere hacer uso de los enlaces nofollow para evitar que Googleboot rastree una zona determinada de la web, con el fin de optimizar el *crawl budget*, pero existen métodos alternativos para no tener que hacer uso de la etiqueta nofollow.

Mediante Javascript, se pueden integrar enlaces una vez el *boot* de Google haya obtenido la información resultante del texto HTML, lo que hace que este enlace sea visual para los usuarios pero no cuenta para la distribución de PageRank. Esta técnica se llama "ofuscación de enlaces". A día de hoy se ha demostrado en varios estudios[15] que si se implementa correctamente este tipo de creación de enlaces mediante JavaScript, se puede optimizar el presupuesto de rastreo.

3.3.3 Ejemplo 3 – Dangling node



[15] Luis Villanueva, profesional SEO, realiza un experimento muy ilustrativo de ofuscación de enlaces en: http://luismvillanueva.com/seo/enlaces-nofollow-realidades-y-mitos-experimento.html

$$\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

El Ejemplo 3 representa un caso de *PageRank death*, concretamente el nodo *D* no transmite autoridad a ningún otro dado que no existen enlaces salientes desde esta URL. Este tipo de URLs que no tienen enlaces salientes son denominadas *dangling nodes*.

En esta situación, una vez el usuario llegue al nodo *D*, no tendrá salida a nivel de navegación interna dentro del sitio web, por lo tanto, la autoridad de ese nodo no se propagará a un conjunto concreto de nodos.

Simulación del PageRank del Ejemplo 3 con un enlace de *D* a *A*:

$$\begin{pmatrix} 0.2425 & 0.2425 & 0.2425 & 0.1363 & 0.1363 \\ 0.2425 & 0.2425 & 0.2425 & 0.1363 & 0.1363 \\ 0.2425 & 0.2425 & 0.2425 & 0.1363 & 0.1363 \\ 0.2425 & 0.2425 & 0.2425 & 0.1363 & 0.1363 \\ 0.2425 & 0.2425 & 0.2425 & 0.1363 & 0.1363 \end{pmatrix}$$

Con el enlace entre *D* y *A*: π^{PR} = (0.2425, 0. 2425, 0. 2425, 0.1363, 0. 1363)

Al igual que en el Ejemplo 2, basado en el mismo concepto del *damping factor*, se reemplazan todos los valores de la fila (en este caso la asociada al nodo *D*) por el valor de 1/*N* siendo *N* el número de nodos total que conforman el grafo de un sitio web. Este ajuste permite modelar la distribución de autoridad de este grafo, pero es necesario volver a puntualizar que Google lo realiza sobre todos los nodos de Internet, por lo que *N* en este caso aumentaría de manera superlativa, por lo que en un caso real, los nodos internos verían su autoridad mínimamente su autoridad, tendiendo este aumento a 0, debido a la cantidad de nodos que conforman internet.

La matriz resultante tras aplicar este nuevo modelado sería:

$$\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Finalmente la matriz resultante tras aplicar la distribución estacionaria y la fórmula del PageRank:

$$\begin{pmatrix} 0.1725 & 0.2602 & 0.2425 & 0.1628 & 0.1628 \\ 0.1725 & 0.2602 & 0.2425 & 0.1628 & 0.1628 \\ 0.1725 & 0.2602 & 0.2425 & 0.1628 & 0.1628 \\ 0.1725 & 0.2602 & 0.2425 & 0.1628 & 0.1628 \\ 0.1725 & 0.2602 & 0.2425 & 0.1628 & 0.1628 \end{pmatrix}$$

$$\pi^{PR} = (0.1725,\ 0.\ 2602,\ 0.\ 2425,\ 0.1628,\ 0.\ 1628)$$

Una vez más, se puede observar como cambia la distribución de autoridad a nivel interno con la existencia de un *dangling node* dentro de un grafo. La autoridad proveniente del nodo *D* se distribuye hacia el resto de nodos del grafo (en un caso real esta autoridad se diluiría por todos los nodos de internet), a diferencia del caso donde hay un enlace de *D* a *A*, donde la autoridad de *A* es mucho mayor, dado que toda la autoridad se transmite de forma directa a este nodo.

Dado que se pierde la posibilidad de transmitir mayor autoridad a nivel interno con la existencia de *dangling nodes*, se recomienda la eliminación de esta tipología de enlazado interno.

3.4 Presente y futuro del PageRank

Google trabaja a diario por combatir el spam o distintos tipos de técnicas conocidas como *Black Hat*[16] que tienen como principal función alterar los resultados de búsqueda por una vía más rápida o con metodologías, en muchos casos, cuestionables. Google persigue la detección de estos sitios web con la intención de penalizarlos o simplemente no tenerlos en cuenta, y pasar a entender su contenido como irrelevante.

En el caso que nos concierne, la transmisión de autoridad a través de enlaces, Google usa distintas técnicas, cada día las perfecciona más para tratar de conocer la naturalidad de los mismos y si realmente son una opción interesante para el usuario. Porque una forma correcta de hacer *link building*[17] es hacer más atractivo el contenido y mejorar la experiencia del usuario.

Estas mejoras que Google implementa recurrentemente con distintas actualizaciones en su algoritmo, pasan por darle mayor peso a aquellos enlaces que transmitan más tráfico (siempre que este sea de calidad y real), a enlaces que se encuentren destacados o estén dentro de un contexto con una elevada relación al contenido de la página, y complementen al contenido mostrado.

¿Cómo consigue Google saber cómo de importante es un enlace dentro de un contexto? Sobre este asunto no hay nada concreto públicamente y a día de hoy hay mera especulación. Mucha gente afirma que el propio Google usa la información recogida por su herramienta Google Analytics[18], pero en numerosas ocasiones la compañía ha afirmado que eso no está dentro de sus políticas y serían muchas las limitaciones si eso fuese así.

Otras teorías más sólidas sobre como Google valora cómo de importante es un enlace dentro de un contexto, se basan a través de técnicas de *machine learning*. Esto

[16] Intento de mejorar el posicionamiento en buscadores de una página web mediante técnicas poco éticas o que contradicen las directrices de Google.
[17] El *link building* es la técnica conocida como la generación de enlaces hacia un sitio web. Dentro de esta técnica encontramos factores importantes como el *anchor text(texto del enlace)* o la tipología del enlace, follow o nofollow.
[18] Herramienta de analítica web. Ofrece información agrupada del tráfico que llega a los sitios web según la audiencia, la adquisición, el comportamiento y las conversiones.

les permite poder simular y estimar cómo los usuarios van a comportarse ante la singularidad de cada *backlink*[19].

[19] Enlaces entrantes que apuntan desde otras páginas a una determinada página.

4. DESARROLLO DE APLICACIÓN

Se ha desarrollado una aplicación en Python que tiene como principal función auditar un dominio y conocer como se distribuye su PageRank a nivel interno, permitiendo conocer como se reparte la autoridad entre sus URLs y la detección de posibles errores que pueden estar penalizando la distribución de autoridad dentro del sitio web seleccionado.

4.1 Tecnología empleada

El programa está desarrollado en Python 3.5. Se ha seleccionado este lenguaje por su sencillez y facilidad en la integración con la base de datos SQLite. También cabe destacar la multitud de librerías disponibles en este lenguaje que facilitan mucho el desarrollo eficiente del programa.

La información es almacenada en una base de datos relacional SQLite. Este tipo de base de datos se diferencia de los típicos gestores de base de datos en que el motor de SQLite no es un proceso independiente con el que el programa principal se comunica, la base de datos pasa a formar parte del mismo proceso. El programa utiliza la funcionalidad de SQLite a través de simples llamadas. Con esto se consigue reducir la latencia, dado que las llamadas a funciones son más eficientes que la comunicación entre distintos procesos. Toda la información relativa a la base de datos es almacenada en la máquina host.

4.2 Implementación

El programa consta de dos partes principales. La primera tiene la misión de rastrear el sitio web seleccionado al completo, “spyder.py”, lo que es conocido como un *crawler*, para así almacenar la información detallada acerca de las URLs y las relaciones entre ellas. La segunda parte, “sprank.py” , se encargará de filtrar toda la información obtenida tras el rastreo y así poder calcular el PageRank de cada nodo para posteriormente indicar incidencias y posibles mejoras a realizar dentro del sitio web.

4.2.1 Base de datos

Toda la información recogida tras rastrear el sitio web y calcular su PageRank es almacenada en una base de datos relacional compuesta por 2 tablas:

- *Pages*: en esta tabla se almacenan las URLs rastreadas, cada una de ellas asociadas a un id único, que está marcado como *primary key*, columna "id". Una columna llamada "url" donde se guarda el *string* asociado a la URL del nodo. Sobre cada URL se almacena el código respuesta HTTP, "status_code". En la columna "is_file" se guarda si la URL es un archivo o no (imágenes, pdf, doc…), y por en "n_enlaces" el número de enlaces internos salientes desde esa URL y su PageRank una vez esté calculado en "new_rank".

	id ▲	url	status_code	new_rank	is_file	n_enlaces
	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro
1	1	https://globaliaespacios.com	200	0.0789	No	14
2	2	https://globaliaespacios.com/reformas...	200	0.151	No	14
3	3	https://globaliaespacios.com/reformas...	200	0.151	No	12
4	4	https://globaliaespacios.com/reformas...	200	0.151	No	9
5	5	https://globaliaespacios.com/contacto	200	0.0789	No	9
6	6	https://globaliaespacios.com/reformas...	200	0.0232	No	11
7	7	https://globaliaespacios.com/reformas...	200	0.0273	No	11
8	8	https://globaliaespacios.com/reforma-...	200	0.0241	No	9
9	9	https://globaliaespacios.com/reformas...	200	0.0292	No	9

Figura 7 - Base de datos "Pages"

- *Links*: esta tabla tiene como principal función guardar la relación de enlazado interno de un sitio web, lo que va a permitir posteriormente crear la matriz de transición. Esta conformada por 4 columnas, las dos primeras columnas, "form_id" y "to_id", indican el id de la URL origen y la URL destino. De cada una de las relaciones se almacena el tipo de enlace que les une, es decir, si el enlace es follow o nofollow, columna "is_follow". Por último, el código de respuesta de la URL destino, "status_code".

	from_id ▲	to_id	is_follow	status_code
	Filtro	Filtro	Filtro	Filtro
1	1	1	follow	200
2	1	2	follow	200
3	1	3	follow	200
4	1	4	follow	200
5	1	5	follow	200
6	1	6	follow	200
7	1	7	follow	200
8	1	8	follow	200
9	1	9	follow	200

Figura 8 - Base de datos "Links"

4.2.2 Crawler

La aplicación cuando es lanzada solicita la inserción de un dominio a rastrear. A partir de este punto se hacen comprobaciones para contrastar que la URL suministrada es correcta y rastreable. Si es correcto, se guarda en base de datos el nodo raíz.

```
1. starturl = input('Escribe el nombre de la página web a rastrear: ')
2.   if ( len(starturl) < 1 ) : starturl = 'https://globaliaespacios.com'

3.   if ( starturl.endswith('/') ) : starturl = starturl[:-1]
4.   web = starturl
5.   if ( starturl.endswith('.htm') or starturl.endswith('.html') ) :
6.       pos = starturl.rfind('/')
7.       web = starturl[:pos]
8.
9.   if ( len(web) > 1 ) :
10.      cur.execute('INSERT OR IGNORE INTO Webs (url) VALUES ( ? )', ( w
   eb, ) )
11.      cur.execute('INSERT OR IGNORE INTO Pages (url, new_rank) VALUES
   ( ?, NULL)', ( web, ) )
12.      conn.commit()
```

A partir de este momento comienza a abrir las URLs que maneja en la tabla *Pages* y aún no ha rastreado, hasta un máximo de 500. Obtiene el código respuesta de la URL y lee su contenido HTML.

```
1. many = 500
2. while True:
3.     if ( many < 1 ) :
4.         break
```

```
5.
6.      many = many - 1
7.
8.      cur.execute('SELECT id,url FROM Pages WHERE status_code is NULL OR
   DER BY RANDOM() LIMIT 1')
9.      try:
10.         row = cur.fetchone()
11.         # print row
12.         fromid = row[0]
13.         url = row[1]
14.     except:
15.         print('No se encuentran más URLs para rastrear')
16.         many = 0
17.         break
```

Una vez sabido si la URL tiene como respuesta 200, y su contenido HTML es legible, se procede a analizar el contenido, en búsqueda de etiquetas HTML de tipo enlace como “Mi dominio”. Esto se logra gracias al uso de la biblioteca BeautifulSoup, que permite identificar distintos tipos de etiquetas HTML en el contenido.

```
1.  try:
2.          document = urlopen(url, context=ctx)
3.
4.          html = document.read()
5.          status_code = document.getcode();
6.          print(status_code)
7.          cur.execute('UPDATE Pages SET status_code=? WHERE url=?', (sta
   tus_code, url) ) # Se almacena en bbdd el tipo de error
8.
9.          soup = BeautifulSoup(html, "html.parser") #Biblioteca de pytho
   n para obtener información de archivos HTML.
10.     except KeyboardInterrupt:
11.         print('')
12.         print('Programa interrumpido por el usuario')
13.         break
14.     except urllib.error.HTTPError as e:
15.             if hasattr(e,'code'):
16.                 cur.execute('UPDATE Pages SET status_code=? WHERE url=
   ?', (e.code, url) )
17.             if hasattr(e,'reason'):
18.                 print(e.reason)
19.
20.             cur.execute('UPDATE Pages SET n_enlaces=? WHERE url=?', (0
   , url) )
21.             print('HTTPError')
22.     except:
23.         print("Ha sido imposible rastrear la página")
24.         cur.execute('UPDATE Pages SET status_code=-
   1 WHERE url=?', (url, ) )
```

```
25.         conn.commit()
26.         continue
27.
28.     cur.execute('INSERT OR IGNORE INTO Pages (url, new_rank) VALUES (
    ?, NULL )', ( url, ) )
29.
30.     conn.commit()
```

Sobre las distintas URLs obtenidas, se rastrea una a una analizando el tipo de enlace que conectan el nodo origen y la página destino, información almacenada en la variable "isFollow". Seguidamente, se analiza si la página destino es de tipo archivo, en la variable "is_file" junto con la obtención de su código respuesta, obtenido en "status_code_live.getcode(); " para así saber si estamos ante un caso de enlazado interno con *PageRank death.*

Al mismo tiempo, durante la recursividad, para analizar los enlaces salientes desde el nodo origen, se va sumando cada iteración en un contador, llamado "count" con la finalidad de saber cuantas enlaces salientes hay desde cada nodo, pieza fundamental en la creación de la matriz que representa el grafo del sitio web.

En el caso que se encuentre ante un nodo origen que de un código respuesta final distinta a 200, al saltar una excepción, no entrará en el bloque de código anterior y no iterará buscando dentro del contenido HTML dado que no existe. Debido a esto, al final del código se pone la condición de que únicamente, en el caso de que el nodo tenga un código respuesta con valor 200, actualizará el valor "count" para que así no sobrescriba el valor anteriormente actualizado a cero. Esto sucederá para todos los casos en el que el nodo origen tenga código respuesta distinto a 200.

```
1. tags = soup('a')
2.     count = 0
3.
4.     #Busqueda en el html del enlazado interno
5.     for tag in tags:
6.
7.         isFollow = tag.get('rel', None)
8.
9.         if not isFollow :
10.            isFollow = 'follow'
11.        else :
12.            for i in isFollow :
13.                if i == 'nofollow' :
14.                    isFollow = 'nofollow'
```

```
                    break

        if isFollow != 'nofollow' :
            isFollow = 'follow'

        href = tag.get('href', None)
        if ( href is None ) : continue
        up = urlparse(href)
        if ( len(up.scheme) < 1 ) :
            href = urljoin(url, href)
        ipos = href.find('#')
        if ( ipos > 1 ) : href = href[:ipos]

        #Se comprueba si el enlazado interno es hacia un archivo lo que provocaría pérdida de linkjuice
        if ( href.endswith('.jpg') or href.endswith('.png') or href.endswith('.gif') or href.endswith('.pdf') or href.endswith('.doc') or href.endswith('.mp4')) :
            is_file = 'Yes'
            cur.execute('UPDATE Pages SET is_file=? WHERE url=?', (is_file, url) )
        else:
            is_file = 'No'
            cur.execute('UPDATE Pages SET is_file=? WHERE url=?', (is_file, url) )
        if ( href.endswith('/') ) : href = href[:-1]
        if ( len(href) < 1 ) : continue

        found = False
        for web in webs:
            if ( href.startswith(web) ) :
                found = True
                break
        if not found : continue

        cur.execute('INSERT OR IGNORE INTO Pages (url, new_rank) VALUES ( ?, 1.0 )', ( href, ) )
        count = count + 1
        conn.commit()

        cur.execute('SELECT id FROM Pages WHERE url=? LIMIT 1', ( href, ))
        try:
            row = cur.fetchone()
            toid = row[0]
        except:
            print('No ha encontrado el id')
            continue

        try:
            status_code_live = urlopen(href, context=ctx)
            statusCode = status_code_live.getcode();
        except urllib.error.HTTPError as e:
           statusCode = e.code
```

```
62.
63.         cur.execute('INSERT OR IGNORE INTO Links (from_id, to_id, is_f
    ollow, status_code) VALUES ( ?, ?, ?, ? )', ( fromid, toid, isFollow,
    statusCode) )
64.
65.     cur.execute('SELECT status_code FROM Pages WHERE url=? LIMIT 1', (
     url, ))
66.     try:
67.         row = cur.fetchone()
68.         status_code = row[0]
69.     except:
70.         print('No ha encontrado el id')
71.         continue
72.
73.     if status_code == 200 :
74.         cur.execute('UPDATE Pages SET n_enlaces=? WHERE url=?', (count
    , url) )
```

4.2.3 Cálculo PageRank

Una vez obtenidos todos los nodos rastreables, con toda su información relacionada, se puede proceder al cálculo del PageRank de cada una de las URLs.

El primer paso es obtener de base de datos y guardarlo en una lista, la relación existente entre el nodo origen y el nodo destino, junto con el tipo de enlace que los une y el número de enlaces salientes desde el origen.

```
1.  # Se crea una lista, con la relación de donde sale el enlace y quien l
    o recibe
2.  # from_id
    el nodo desde donde sale el enlace y to_id para el que receptor
3.  links = list()
4.  cur.execute('''''SELECT from_id, to_id, is_follow, n_enlaces FROM Link
    s INNER JOIN Pages on Links.from_id = Pages.id''')
5.  for row in cur:
6.      from_id_related = row[0]
7.      to_id_related = row[1]
8.      is_follow = row[2]
9.      n_links_row = row[3]
10.     links.append(row) # Se almacena la relacion en la lista del enlaza
    do interno en links
```

A continuación, es el momento de construir la matriz transitoria inicial que representa el grafo del dominio. El tamaño de la matriz viene marcado por el número total de nodos del dominio rastreados, variable almacenada en “count”.

```
1.  cur.execute('SELECT count (DISTINCT to_id) FROM Links')
2.  row = cur.fetchone()
```

```
count = row[0]

def is_not_follow(p, rows, matrix):
    value = 1/p[3] * 1/rows
    for i in range(rows):
        matrix[p[0] - 1, i] = matrix[p[0] - 1, i] + value

def is_equal_zero(p, rows, matrix):
    for i in range(rows):
        matrix[p[0] - 1, i] = matrix[p[0] - 1, i] + 1/rows

matrix_damping_factor = np.full((n_columns, m_rows), 1/count)

matrix = numpy.zeros(shape=(m_rows,n_columns))

for p in links:
    if p[3] is not None:
        if p[2] == 'nofollow':
            is_not_follow(p, m_rows, matrix)
        elif p[3] == 0: # caso que sea un dangling node
            is_equal_zero(p, m_rows, matrix)
        elif matrix[p[0] - 1][p[1] - 1] == 0.00:
            matrix[p[0] - 1][p[1] - 1] = 1/p[3]
        else: # si existe más de un enlace a una determinada casilla
            matrix[p[0] - 1][p[1] - 1] = matrix[p[0] - 1][p[1] -
    1] + 1/p[3]
```

Una vez formada la matriz inicial se procede a calcular la distribución estacionaria, para ello se ha hecho uso de la biblioteca “discreteMarkovChain” que dispone de una función que devuelve el vector de la distribución estacionaria si se le pasa la matriz transitoria.

Por último, se aplica a este vector resultante la fórmula que incluye el *damping factor*.

```
def pagerank_value(stationary_distribution, n_columns):
    for x in range(n_columns):
        stationary_distribution[x] = (stationary_distribution[x] * 0.8
   5) + (1/n_columns * 0.15)
        stationary_distribution[x] = str.format('{0:.4f}', stationary_
   distribution[x])

        cur.execute('UPDATE Pages SET new_rank=? WHERE id=?', (station
   ary_distribution[x], x+1))
        cur.execute('SELECT url FROM Pages WHERE id=? LIMIT 1', ( x+1,
    ))
        try:
            row = cur.fetchone()
            url = row[0]
        except:
```

```
12.             print('No ha encontrado el id')
13.             continue
14.
15.         print(url, stationary_distribution[x], end='\n')
```

Como resultado final se obtiene el PageRank asociado a cada una de las URLs.

5. CONCLUSIONES Y LINEAS FUTURAS DE ACTUACIÓN

Como resultado final a la realización de este TFG se han conseguido los objetivos marcados, desarrollando un análisis profundo de cómo los enlaces entre URLs afectan en la distribución de autoridad y tras este estudio, el desarrollo de un programa que puede auditar sitios web para poder proponer implementaciones dentro de la página con el fin de mejorar la autoridad de sus nodos.

Una evolución interesante del proyecto sería dar un paso más allá, teniendo en cuenta el enlazado externo que apunta al sitio web. Realizando una estimación del *link juice* transmitido y PageRank de esas URLs externas gracias al uso de herramientas como *Ahrefs*, que almacenan ese tipo de información y disponen de APIs para esos recursos. *Ahrefs* almacena todos los enlaces entrantes a un sitio web y les asigna bajo una escala creada por ellos la autoridad tanto del dominio como de la propia URL en concreto. A partir de aquí se podría hacer una estimación y cruzado de datos entre la autoridad externa e interna, para así identificar los nodos con mayor autoridad. Esta funcionalidad adquiriría mucho valor para aquellos casos que se quiera comparar la autoridad entre distintos dominios.

El proyecto ha sido una oportunidad que me ha aportado grandes beneficios, debido a que me ha permitido poner en práctica los distintos tipos de competencias adquiridos a lo largo de la carrera. Competencias técnicas como la programación y la probabilidad, permitiendo el sumergirme en un lenguaje nuevo para mí como es Python y resolver casos singulares de la parte de probabilidad. Por otro lado, otro tipo de competencias transversales que permiten planificar proyectos, tener capacidad resolutiva, y generar propuestas de valor y útiles.

6. BIBLIOGRAFÍA

[1] Using page speed in mobile search ranking – Web oficial de Google de indexación

https://webmasters.googleblog.com/2018/01/using-page-speed-in-mobile-search.html

[2] Cómo usar rel="nofollow"– Web oficial de ayuda a desarrolladores de Google

https://support.google.com/webmasters/answer/96569?hl=es

[3] Patente Google

https://patentimages.storage.googleapis.com/78/90/34/61ae272aecc94b/US9305099.pdf

[4] The Anatomy of a Large-Scale Hypertextual Web Search Engine

http://infolab.stanford.edu/~backrub/google.html

[5] SQLite - Wikipedia

https://es.wikipedia.org/wiki/SQLite

[6] Cómo funciona Google: cadenas de Markov y valores propios - Blog Proyecto Klein

http://blog.kleinproject.org/?p=1605&lang=es

[7] PageRank Algorithm

https://www.researchgate.net/publication/314235791_PageRank_Algorithm

[8] PageRank Algorithm - Kenneth Shum

http://home.ie.cuhk.edu.hk/~wkshum/papers/pagerank.pdf

[9] A Modified Algorithm to Handle Dangling Pages using Hypothetical Node

https://pdfs.semanticscholar.org/6ad1/ca4a8cf78f7a689cf20ff615304ee4c58fdf.pdf

[10] Discrete Markov Chain in Python library

https://github.com/gvanderheide/discreteMarkovChain

Este documento esta firmado por

Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Wed Jun 06 21:08:55 CEST 2018
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)