Arquitectura de Google

UCR – ECCI

CI-2414 Recuperación de Información

Prof. Kryscia Daviana Ramírez Benavides

Introducción

- Google fue fundada el 7 de septiembre de 1998 por Larry Page y Sergey Brin (dos estudiantes de doctorado en Ciencias de la Computación de la Universidad de Stanford).
- La idea de formar Google nace con la necesidad de obtener mejores respuestas de los motores de búsqueda.
  - Hasta 1998 ya se había logrado el objetivo de hacer motores de búsqueda rápidos, era tiempo de enfocarse en la calidad.
- Brin y Page, se dieron cuenta de que solo indexar las páginas no es la solución a una mejor búsqueda.
  - Ocurría muy a menudo que los resultados realmente relevantes en una búsqueda, eran depreciados por resultados mediocres.

Introducción (cont.)

- El avance en los motores de búsqueda se debe enfocar en poner entre las primeras diez páginas lo que el usuario promedio está buscando.
- Toda esta necesidad de interacción con el usuario promedio, presenta un nuevo objetivo que debe ser alcanzado:
  - Construir un sistema que la mayoría de las personas puedan utilizar sin problemas.
- La arquitectura de Google está diseñada para guardar todos los documentos que se encuentren en el rastreo.

Características

- Google se está consolidando a grandes pasos como el preferido para realizar búsquedas.
- Sus principales ventajas se deben a que es muy rápido, y sus resultados son relevantes y bastante bien ordenados.
- Para jerarquizar sus páginas utiliza diversos factores tales como *modelo vectorial*, *texto de anchors*, *Page Rank*.

Características (cont.)

- Google indexa más de 3 mil millones de páginas Web, aunque ofrecen más resultados gracias a los "rastreos profundos".
- Hay varios "rastreadores" (*crawlers*):
  - El general (una vez al mes), que busca en la mayoría de la WWW.
  - El *Fresh*, que rastrea en las páginas que se actualizan frecuentemente.
  - El de noticias, que rastrea cada 10 minutos.
- El servidor Web que utilizan es personalizado, llamado *Google Web Server - GWS* (se sospecha que se trata del servidor Apache modificado); actualmente existe la versión 2.1.

Características (cont.)

- Hay 4 tipos de servidores en el clúster de Google, situados en paralelo del servidor Web:
  - **Los servidores índice:** Están divididos en fragmentos (por ejemplo, uno apunta a todo lo que comienza con la letra 'a'), y devuelve al servidor Web una lista con las IDs de documentos donde aparece una determinada palabra.
  - **Los servidores de documentos:** Contienen las copias caché de las páginas Web que se rastrean, el código HTML plano de los documentos está disponible en los almacenes de Google.
  - **Los servidores correctores de deletreo:** Son los que nos muestran el mensaje "Quiso decir: ...".
  - **Los servidores *AdWords*:** Muestran los enlaces patrocinados.

Características (cont.)

- Google analiza más de 100 factores para determinar la relevancia de una página Web. Entre ellos, destacan el texto del enlace (el *anchor text*), el tamaño de la fuente y la proximidad.
  - Mantiene información de la posición de los términos que aparecen dentro de los documentos indexados, lo que permite búsquedas por proximidad.
  - Mantiene información de la apariencia visual de los documentos (ej.: a las palabras marcadas en negrita o con un tamaño de letra mayor se les concede mayor peso al calcular la relevancia).

Características (cont.)

- Uno de los factores más importantes además de la alta calidad y facilidad de las búsquedas es el ***PageRank***, un método sofisticado para asignar la importancia a cada documento de la ***Word Wide Web***.
- Para calcular el valor del *PageRank* , Google utiliza la teoría de grafos, mediante una matriz de 30 billones de nodos. Cada uno de estos nodos tiene 10 arcos (o aristas) diferentes.

PageRank

- Con el propósito de impedir que los *webmaster* realizaran "páginas de entrada" que consisten en páginas optimizadas para un criterio de búsqueda en concreto, fue desarrollado el concepto de popularidad de ***PageRank***.
- Este concepto consiste básicamente en contar los links entrantes y salientes de un sitio Web, de esta forma mientras un sitio en la red tenga más links provenientes de otros sitios, este sitio Web generalmente tendrá más importancia.

PageRank (cont.)

- Numerosos *webmaster* intentan eludir este sistema, creando muchos links entrantes desde otros sitios Web que en muchos casos suelen ser de poca importancia.
- Pero el sistema de popularidad de ***PageRank*** no solo está basado en el número de links entrantes, también tiene en cuenta la importancia del sitio que tiene el link.

PageRank (cont.)

- **Algoritmo:**

$$PR(i) = d \cdot \sum_{j \in B(i)} \frac{PR(j)}{N(j)} + \frac{(1-d)}{m}$$

  - *PR(i)* es el *PageRank* de la página *i*.
  - *PR(j)* es el *PageRank* de las páginas *j* que tienen un link a la página *i*.
  - *N(j)* es el número de enlaces salientes de la página *j*.
  - *B(i)* es el número de páginas que apuntan a la página *i*.
  - *d* es un factor de decaimiento (entre 0 y 1).
  - *m* es el número total de nodos en el grafo.

PageRank (cont.)

- Fácilmente calculable con algoritmos iterativos.
- Características del "navegante" aleatorio:
  - El *PageRank* es la probabilidad de que este "navegante" acabe en una determinada página Web partiendo de una de entrada.
  - El factor d se puede ver como la probabilidad de que el "navegante" se aburra.
- El *PageRank* para una página será alto:
  - Si existen muchas páginas apuntándola.
  - O aunque la apunten pocas páginas, éstas tienen *PageRank* alto.

PageRank (cont.)

- El atributo *nofollow* fue creado por Google y luego aceptado por otros buscadores para dar la posibilidad a los *webmasters* de indicar a los rastreadores o bots que esos enlaces no deberían ser seguidos y que no tenían importancia en una determinada página Web.
- Al momento de calcular el *PageRank* de las páginas internas de una determinada página, Google no repartía o heredaba (el llamado juicio *PageRank*) este valor a las páginas con el atributo *nofollow*.

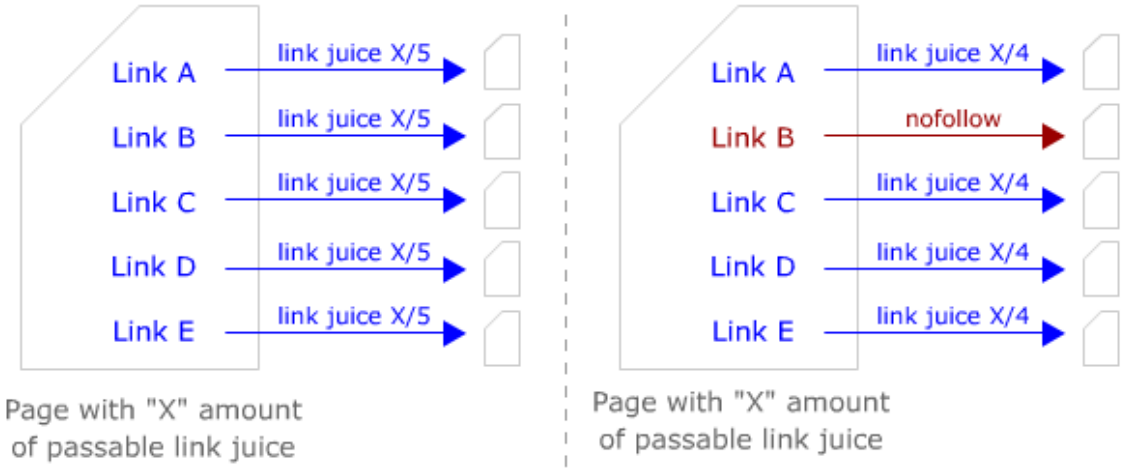
PageRank (cont.)

- De esta manera, los *webmasters* aprovecharon este fenómeno para **canalizar** el juicio *PageRank* a las páginas que tenían más importancia.
- Esté método se llamó ***PageRank Sculpting***, o esculpir el *PageRank*.
- Matt Cutts menciona que el *PageRank* ya no existe en su forma clásica, como lo dieron a conocer los fundadores de Google.

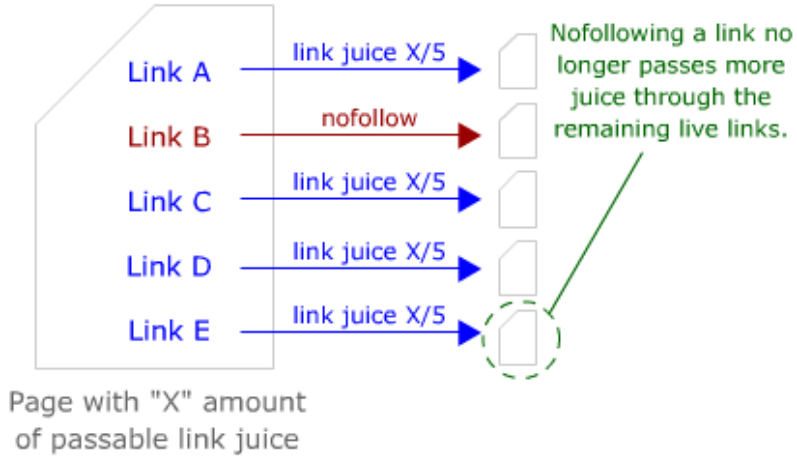
PageRank (cont.)

- Para evitar la propagación infinita del *PageRank* en enlaces que retornan a la misma página, Larry y Sergei definieron un factor de decrecimiento de unos 10% - 15% de modo que el valor del *PageRank* iba cayendo de página a página hasta desaparecer.
- Con la introducción del atributo *nofollow*, antes, en Google estos enlaces NO pasan el *PageRank* y tampoco pasan el *anchor text*.
- Pero se cambió este algoritmo, y los enlaces con el atributo *nofollow* **absorben** el valor del *PageRank* que les debería tocar.

Old PageRank Algorithm & Effect of Nofollow Sculpting



New PageRank Algorithm & Effect of Nofollow Sculpting



PageRank (cont.)

- Esculpir el *PageRank* de un sitio Web con el atributo *nofollow*, ya no funciona, y los *webmasters* NO se habían dado cuenta de esto.
- Esculpir o modelar el *PageRank* interno de una web, todavía es posible.
- Lo que aún no está claro, es qué se debe hacer con los comentarios en un blog que contienen el atributo *nofollow*. El *PageRank* de estos enlaces ahora se desvanece y se pierde sin poder usarse en las otras páginas del blog.

Texto en los Enlaces

- El texto anclado, es el texto al que se le hace clic cuando se hace clic a un *hyperlink*. No es el texto del link, si no la máscara que hace más descriptivo este link.
- La mayoría de los buscadores asocian el texto de un enlace (*anchor text*) con la página en la que aparece.
- Google asocia el texto del enlace con la página a la que apunta.

Texto en los Enlaces (cont.)

- Ventajas:
  - El texto de los enlaces, con frecuencia, proporciona descripciones más acertadas de las páginas web que las páginas mismas.
  - Pueden existir enlaces a documentos (imágenes, programas, direcciones de *e-mail*, bases de datos, etc.) que no pueden ser indexados por motores de búsqueda textuales.
  - Permite devolver documentos en las búsquedas que no han sido rastreados.

Texto en los Enlaces (cont.)

- Desventajas:
  - Pueden devolver páginas inexistentes.
  - Es posible que se retorne páginas que no han sido indexadas, por lo tanto es posible que nunca hayan sido revisadas. En este caso, puede suceder que la página nunca existiera, pero este problema puede llegar a ser solucionado si ordenamos los resultados en base a *PageRank*.
  - Es muy difícil implementar el uso de los enlaces eficientemente, ya que pueden apuntar a datos que son bastante grandes; estos datos deben ser procesados.

Hardware

- Google tiene muchos sitios Web indexados y los cálculos los hace en poco tiempo, ya que las búsquedas son generadas en menos de un segundo.
- Pues se ha hecho un estudio sobre el hardware de Google y las cifras son exorbitantes, ronda un gasto de unos 250 millones de dólares en *hardware*:
  - Entre 45.000 y 80.000 servidores.
  - 69.000 las máquinas y 539 *racks* (el *rack* es algo así como: 88 dual-CPU 2Ghz Intel Xeon servers con 2Gbytes de RAM y un disco duro de 80Gbytes).

Hardware (cont.)

- El hardware original que utilizaba Google al poco tiempo de su fundación, cuando aún estaba en la Universidad de Stanford, incluía los siguiente:
  - Sun Ultra II con procesador de 200MHz dual y 256 MB de RAM. Esta era la máquina principal del sistema original.
  - Dos servidores Pentium II duales a 300 MHz donados por Intel que incluían 512 MB de RAM y 9 discos de 9 GB entre los dos servidores. Era en estos servidores donde se ejecutaba la parte principal de la búsqueda.
  - F50 IBM RS/6000 donado por IBM que incluía cuatro procesadores, 512 MB de memoria y ocho discos duros de 9 GB.

Hardware (cont.)

- El hardware original que utilizaba Google al poco tiempo de su fundación, cuando aún estaba en la Universidad de Stanford, incluía los siguiente (cont.):
  - Dos armarios adicionales incluían tres discos duros de 9 GB y seis de 4 GB respectivamente que estaban conectados al servidor Sun Ultra II.
  - Un armario de expansión de discos de IBM con otros ocho discos duros de 9 GB donados por IBM.
  - Armario de disco duros casero que contenía 10 discos duros de 9 GB SCSI.

Hardware (cont.)

Primera oficina de Google.
Consiguió hacer funcionar varias máquinas totalmente diferentes con un impresionante rendimiento.

Una máquina hecha a medida, con techo de LEGO.
Go lego!!! ⇒ Google!!!

Hardware (cont.)

Puede ser uno de los primeros logotipos de Google, en 1997, un año antes del sitio web de Google.

El primer logotipo de Google fue creado por Sergey Brin, utilizando el programa de gráficos libre GIMP. Un signo de exclamación se añadió, imitando al logotipo de Yahoo!.

Hardware (cont.)

- Ahora viene la parte más interesante, si hacen los cálculos la baja, según el estudio de Tln sería:
  - 539 *racks*.
  - 47,432 máquinas.
  - 94,864 CPUs.
  - 189,728 GHz de poder de procesamiento.
  - 94,864 GB de RAM.
  - 3,705 TB de espacio en el disco duro.

Hardware (cont.)

- La mitad de las cuentas darían algo así:
  - 719 *racks*.
  - 63,262 máquinas.
  - 126,544 CPUs.
  - 253,088 GHz de poder de procesamiento.
  - 126,544 GB de RAM.
  - 4,943 TB de espacio en el disco duro.

Hardware (cont.)

- Y redondeando hacia arriba queda algo de otro mundo:
  - 899 *racks*.
  - 79,112 máquinas.
  - 158,224 CPUs.
  - 316,448 GHz de poder de procesamiento.
  - 158,224 GB de RAM.
  - 6,180 TB de espacio en el disco duro.

Hardware (cont.)

- En el estudio se dice que Google estará en el primer puesto de las máquinas más rápidas del mundo, ya que haciendo cálculos a la baja, tendría una capacidad de cálculo de 189 *teraflops*, un cálculo medio daría como resultado 253 *teraflops* y si somos generosos y calculamos redondeando hacia arriba nos quedaríamos con 316 *teraflops* de potencia.

Hardware (cont.)

- Actualmente, según datos, los servidores son ordenadores personales x86 que utilizan una versión personalizada de Linux.
  - El objetivo es comprar procesadores que ofrezcan el mejor rendimiento por unidad de energía, no los procesadores más potentes directamente.
  - Se estima que se necesitan 20 MW para poder alimentar a 450.000 servidores, lo que podría tener un coste asociado de unos 2 millones de dólares americanos al mes en facturas de electricidad.

Hardware (cont.)

- Especificaciones del *hardware* del año 2003:
  - Más de 15.000 servidores con velocidades comprendidas entre el Intel Celeron de 533 MHz y el Pentium III a 1,4 GHz dual (a fecha de 2003). Según Paul Strassman, Google tendría en 2005 unos 200.000 servidores mientras que algunas fuentes indican que el número de servidores podría haber alcanzado los 450.000 en 2006.
  - Uno o más discos duros de 80 GB por servidor (en 2003).
  - Entre 2 y 4 GB de memoria por máquina.

Hardware (cont.)

- El tamaño exacto de los centros de datos que Google utiliza es desconocido, y las cifras oficiales se mantienen poco precisas intencionadamente.
- Según una estimación del año 2000, la granja de servidores de Google estaba compuesta por 6000 procesadores, 12.000 discos duros IDE (dos por máquina) en cuatro centros físicos: dos en Silicon Valley y dos en Virginia.
- Cada centro tenía una conexión de fibra óptica de 2488 Mbit/s y otra de 622 Mbit/s. Los servidores ejecutan un software llamado Google Web Server.

Hardware (cont.)

- Actualmente Google está desarrollando un supercomputador en un centro de datos en Dallas.
- El proyecto se llama **Proyecto O2** y se espera que incremente sustancialmente la capacidad de su red global actual, permitiendo ejecutar miles de millones de búsquedas al día y un catálogo de otros servicios que cada vez crece más.
- El nuevo complejo tiene el tamaño de dos campos de fútbol con torres de refrigeración de cuatro pisos.

Topología de Red

- A pesar de que no se conocen las cifras exactas, se estima que Google mantiene más de 450.000 servidores, ordenados en *racks* de *clusters* en varias ciudades del mundo.
  - Los principales centros se encuentran en Mountain View (California), Virginia, Atlanta y Dublín. Hay otras instalaciones en construcción en Dallas y Saint-Ghislain..
- En 2009 Google tiene previsto inaugurar otra instalación ecológica en Council Bluffs, cerca de una fuente abundante de energía eólica y de una red de fibra óptica.

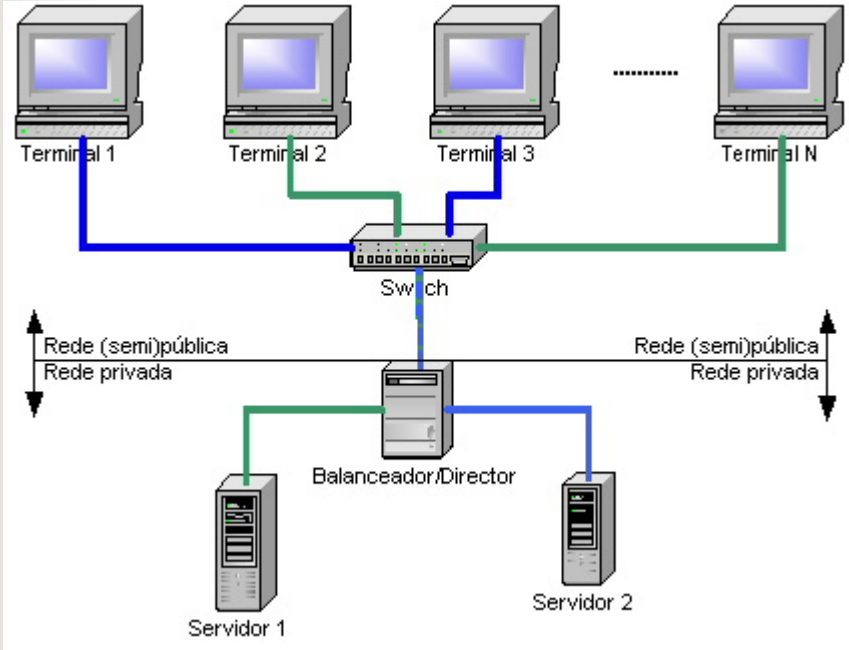
Topología de Red (cont.)

- Gracias a la dispersión geográfica de sus servidores, Google puede ofrecer un servicio más rápido a los usuarios, lo cual es vital teniendo en cuenta que en el año 2005 Google había indexado 8.000 millones de sitios Web.
- Cuando se hace conexión a Google, los servidores DNS traducen la dirección www.google.com a varias IP's distintas, permitiendo que se distribuya la carga entre varios *clusters*.
  - Cuando un dominio tiene varias IP's, como en el caso de Google, el orden en que los servidores DNS traducen las direcciones IP se calcula mediante el sistema de planificación *round-robin*.

Topología de Red (cont.)

- Cada *cluster* tiene miles de servidores, por lo que cuando alguien se conecta a un *cluster*, se distribuye la carga de nuevo mediante el *hardware* del *cluster* para enviar la consulta al servidor Web que esté menos ocupado en ese momento.
- Los *racks* de Google están hechos a medida y pueden contener entre 40 y 80 servidores.
  - Cada *rack* tiene una conexión *ethernet* a un *router* local que a su vez se conecta al *router* central utilizando una conexión de 1 Gigabit.

Topología de Red (cont.)



Arquitectura típica de un balanceador de carga.

Un centro de datos donde se pueden ver varios *racks*.

Tipos de Servidores

- La infraestructura de servidores de Google esta dividida en varias categorías, cada una con un propósito diferente:
  - Los distribuidores de carga aceptan la petición del cliente y la reenvían a uno de los servidores Web de Google a través de servidores proxy Squid.
  - Los servidores proxy Squid aceptan la petición y devuelven el resultado desde la caché local si es posible y si no reenvían la petición al servidor web.

Tipos de Servidores (cont.)

- La infraestructura de servidores de Google esta dividida en varias categorías, cada una con un propósito diferente (cont.):
  - Los servidores Web coordinan la ejecución de las consultas enviadas por los usuarios y formatean el resultado utilizando el lenguaje HTML. La ejecución consiste en enviar peticiones a servidores de índices, fusionar los resultados, calcular su rango utilizado *PageRank*, elaborar un resumen para cada resultado, preguntar por posibles sugerencias a los servidores de ortografía y finalmente obtener una lista de anuncios del servidor de publicidad.
  - Los servidores de recolección de datos navegan por Internet al estilo araña. Van actualizando el índice y las bases de datos de documentos con las páginas Web que van encontrando, y aplican los algoritmos de Google para calcular el rango de cada página.

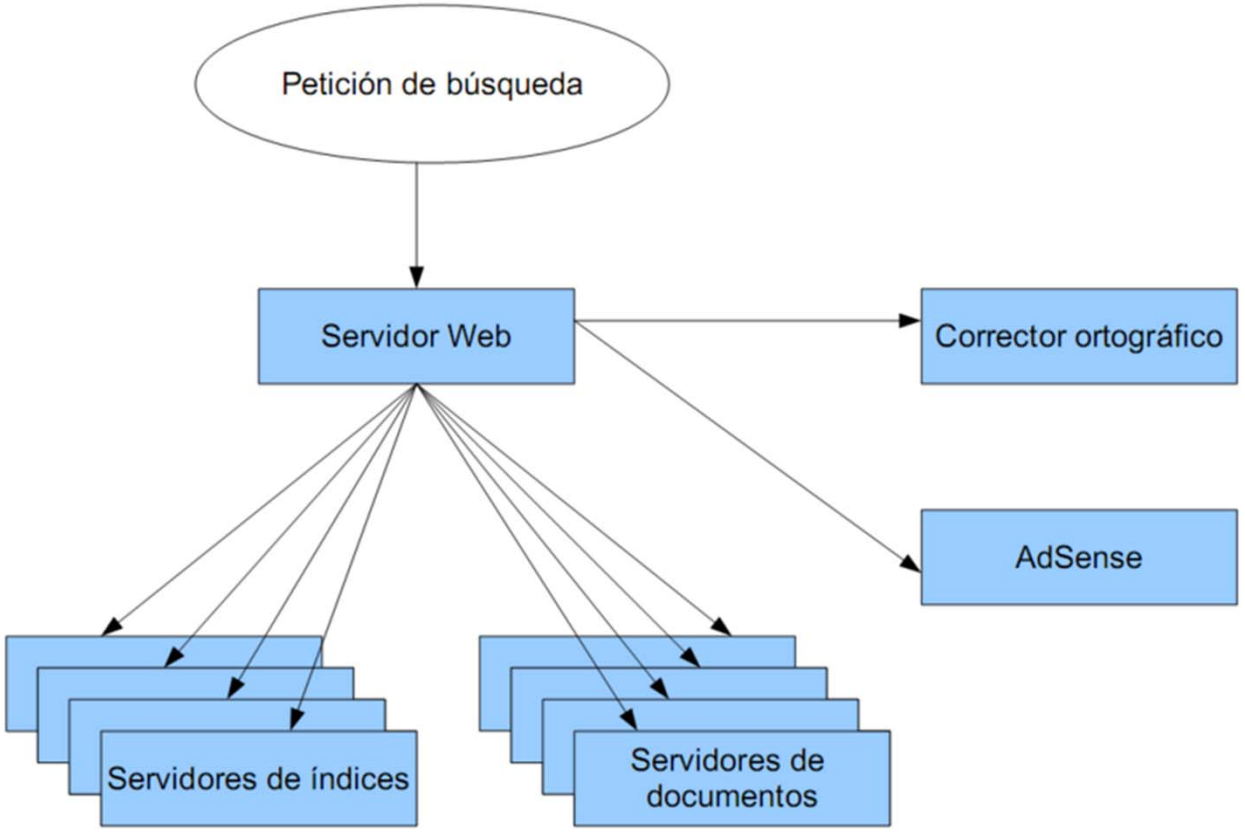
Tipos de Servidores (cont.)

- La infraestructura de servidores de Google esta dividida en varias categorías, cada una con un propósito diferente (cont.):
  - Los servidores de índices contienen un conjunto de trozos de índice. Devuelven una lista de id's de documentos, llamados “docid”, de forma que los documentos a los que identifican contienen la palabra que el usuario está buscando. Estos servidores necesitan menos espacio en disco, pero en cambio soportan un carga de procesador bastante elevada.

Tipos de Servidores (cont.)

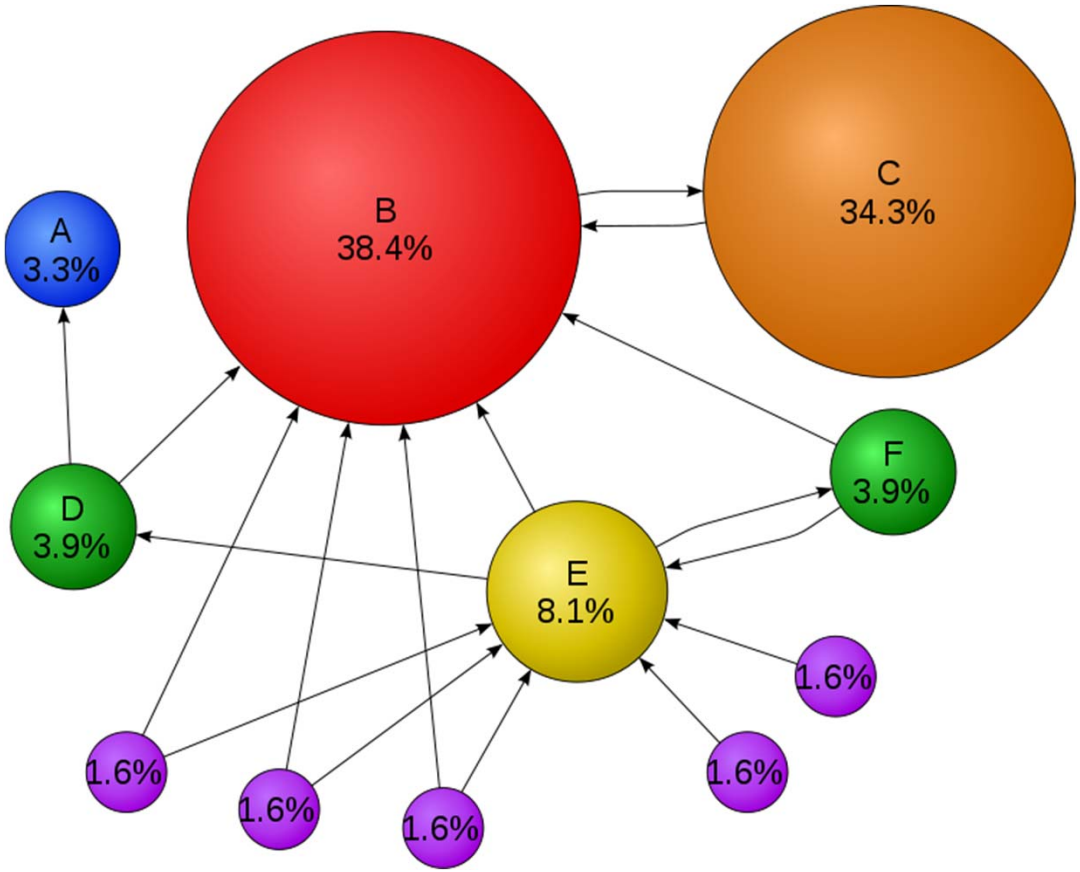
- La infraestructura de servidores de Google esta dividida en varias categorías, cada una con un propósito diferente (cont.):
  - Los servidores de documentos sirven para almacenar los documentos; cada documento se almacena en docenas de servidores de documentos. Cuando alguien realiza una búsqueda, el servidor de documentos devuelve un resumen de la página basado en las palabras buscadas por el usuario. También puede devolver el documento entero directamente si se lo solicitan. Estos servidores requieren bastante espacio de disco.
  - Los servidores de anuncios (*ad servers*) gestionan la publicidad de los servicios *AdWords* y *AdSense*.

Tipos de Servidores (cont.)



Arquitectura básica de Google.

Tipos de Servidores (cont.)



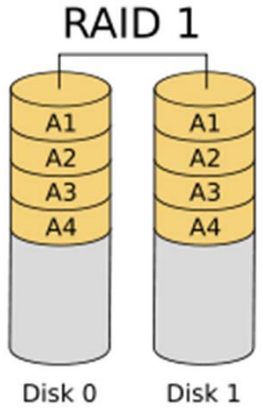
Ejemplo del algoritmo *PageRank*.

Operación de los Servidores

- La mayoría de operaciones son de solo lectura; cuando se necesita una actualización de datos, las consultas se envían a otros servidores, para simplificar los problemas de consistencia.
- Las consultas se dividen en subconsultas, cada una de ellas se envía por diferentes canales en paralelo, reduciendo así el tiempo de latencia.

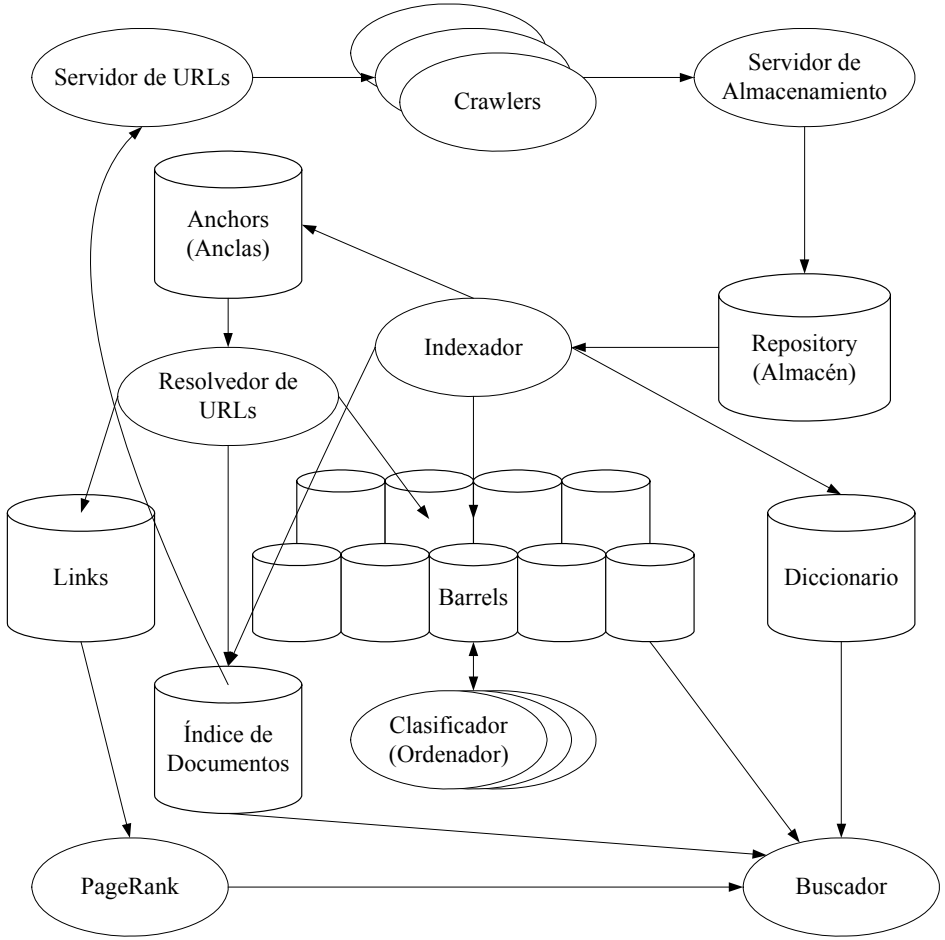
Operación de los Servidores (cont.)

- Para reducir los efectos de un posible fallo de hardware, los datos almacenados en los servidores se duplican utilizando tecnología RAID (febrero de 2008).
- El software también está diseñado para gestionar los fallos. Por lo tanto, cuando un servidor se cae, los datos todavía están disponibles en otros servidores.

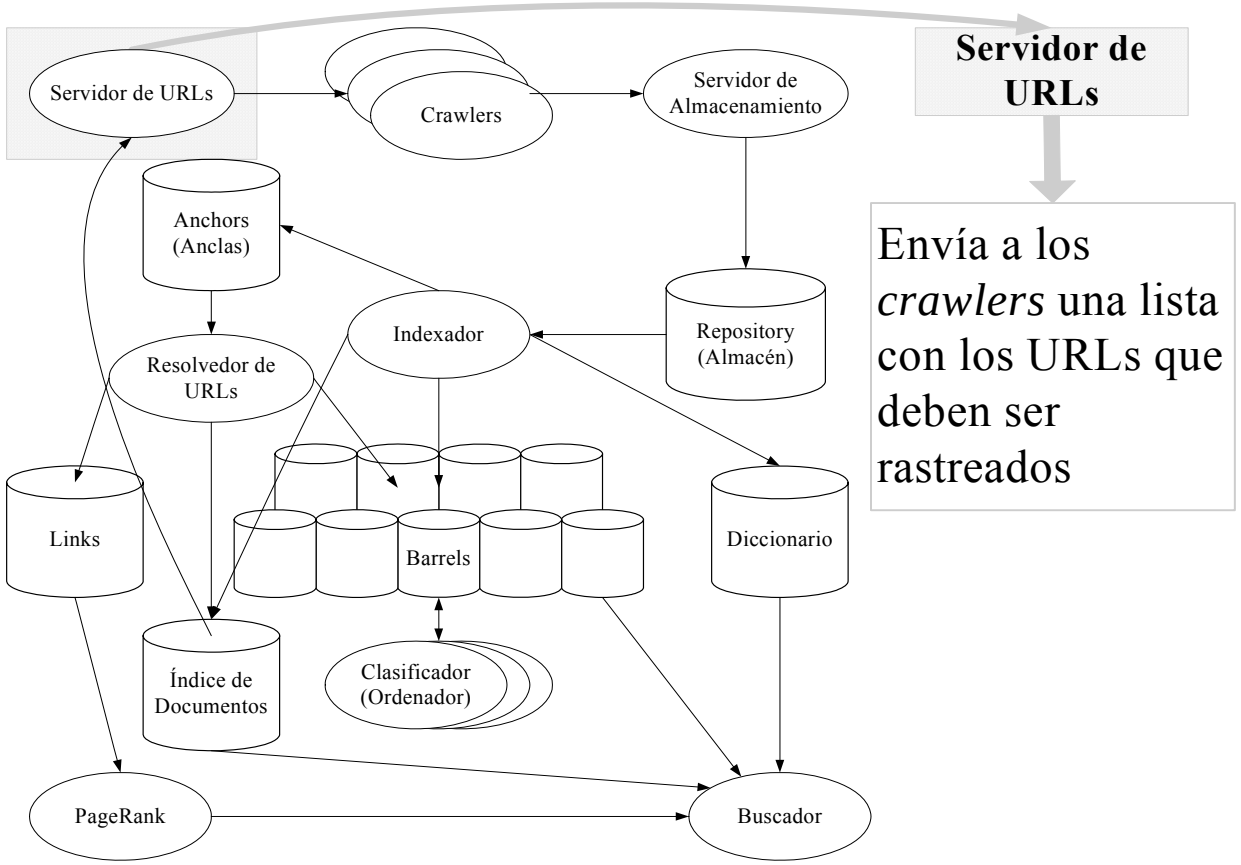


Funcionamiento de un sistema RAID, en modo *mirroring* (copia de datos).

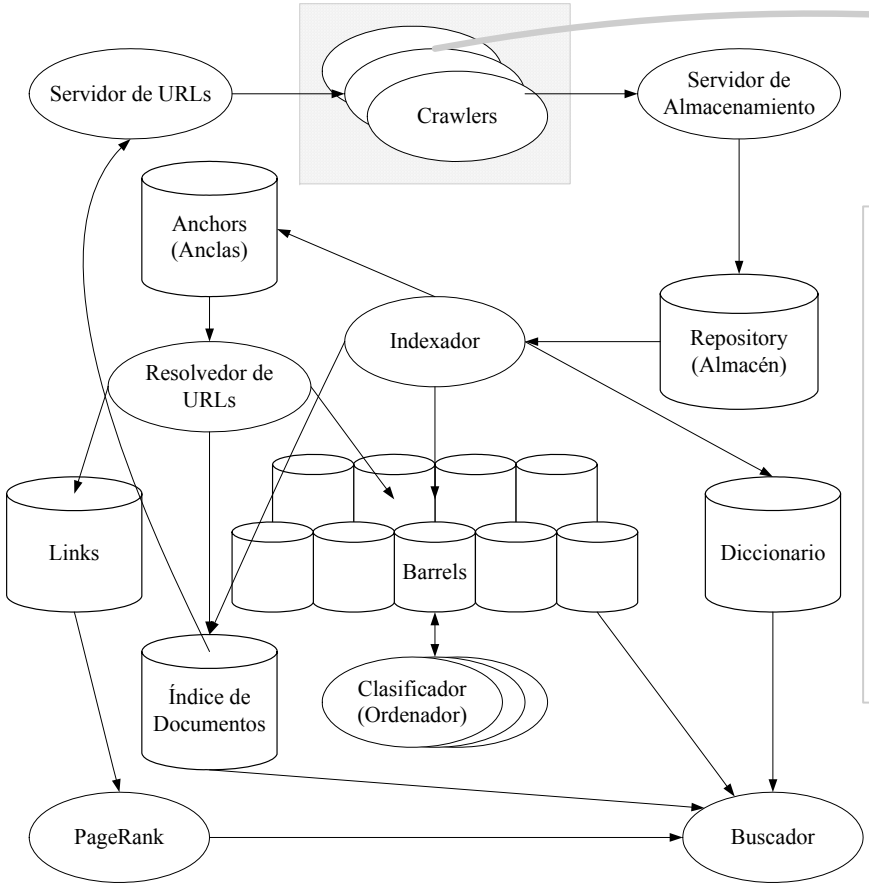
Arquitectura de Google



Arquitectura de Google (cont.)



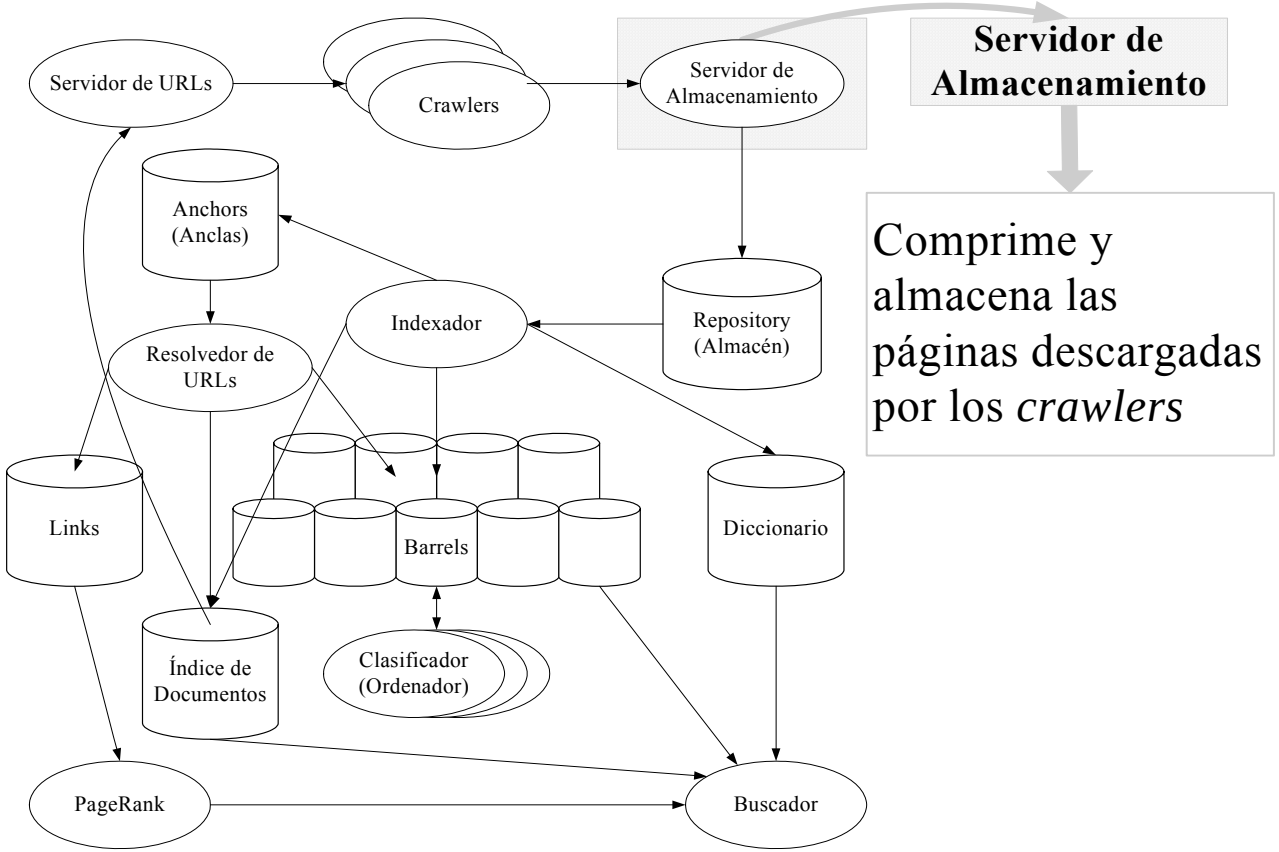
Arquitectura de Google (cont.)



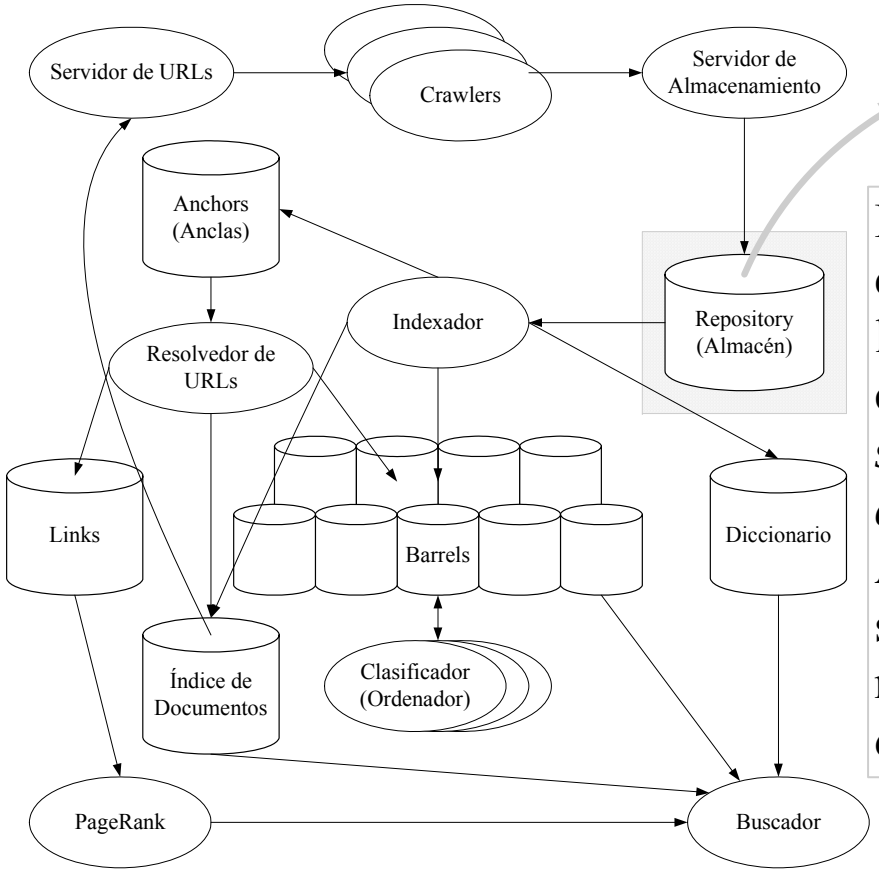
***Crawlers* Distribuidos**

Permiten realizar el *crawling* o rastreo de los servidores web, con el fin de indexar la información que contienen

Arquitectura de Google (cont.)



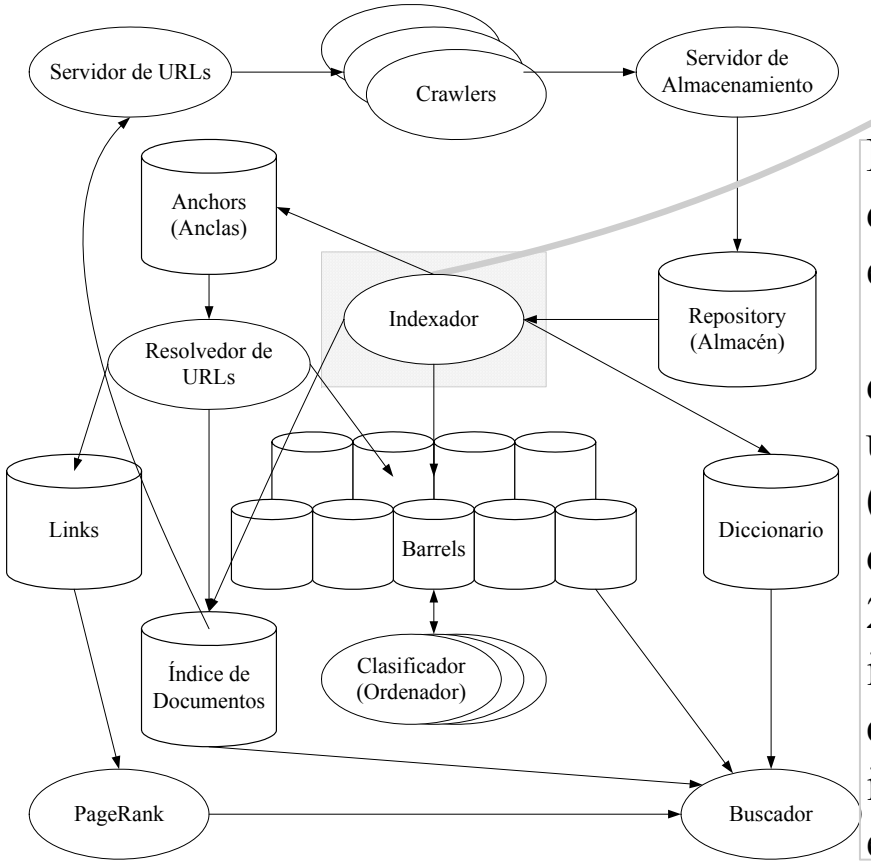
Arquitectura de Google (cont.)



Repositorio (Almacén)

Lugar físico en el que se almacenan los documentos comprimidos por el *servidor de almacenamiento*
A cada página web se le asigna un número denominado **docID**

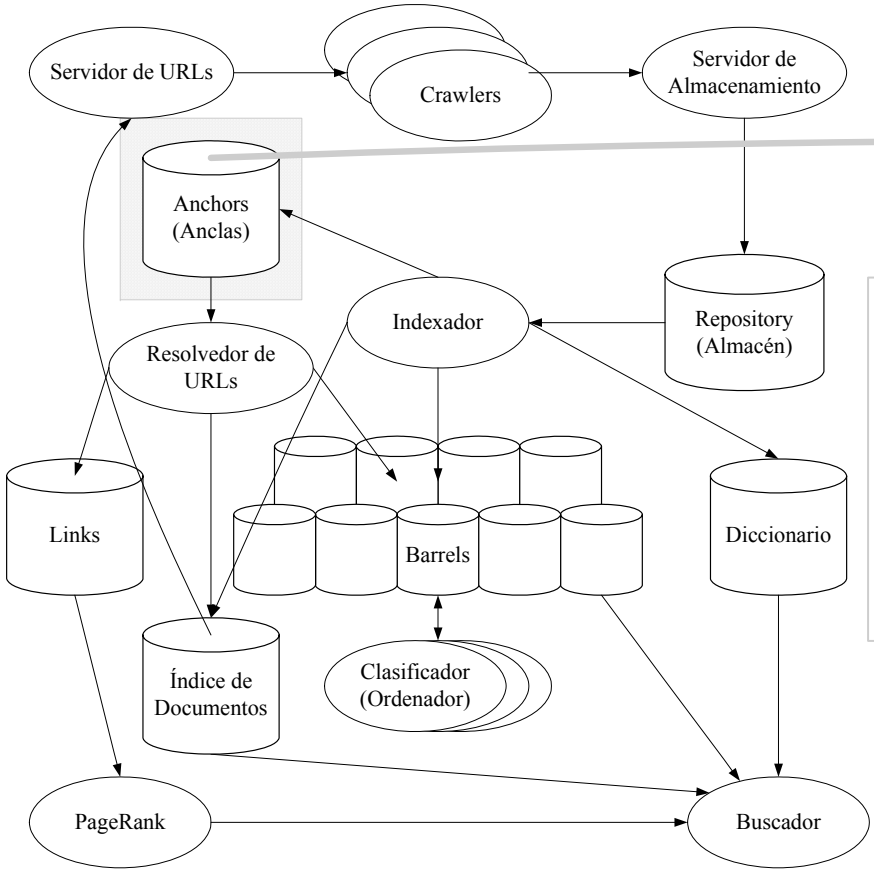
Arquitectura de Google (cont.)



Indexador

Lee y descomprime los documentos existentes en el *almacén*:
1. Realiza el *parsing*. Cada documento se convierte en un conjunto de *hits* (palabra, posición, tamaño de fuente, capitalización)
2. Extrae información importante de los *links* de cada documento. Esta información se almacena en un fichero de **anclas**

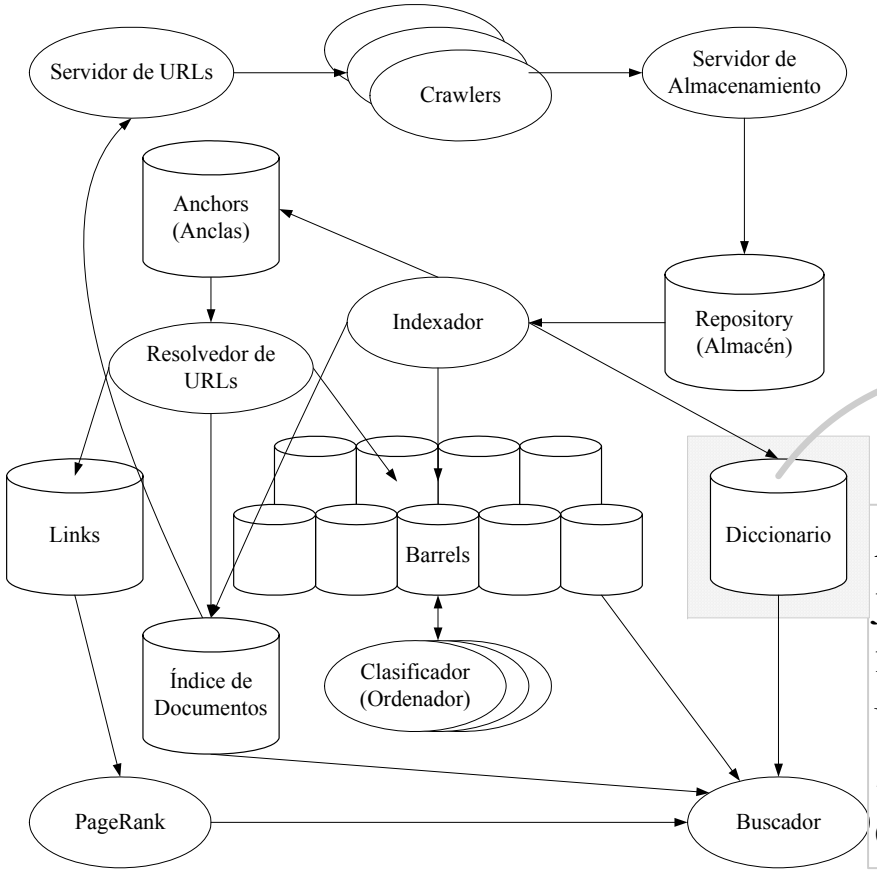
Arquitectura de Google (cont.)



***Anchors* (Anclas)**

Almacena la información generada por el indexador: desde y a donde apuntan los enlaces, y su texto

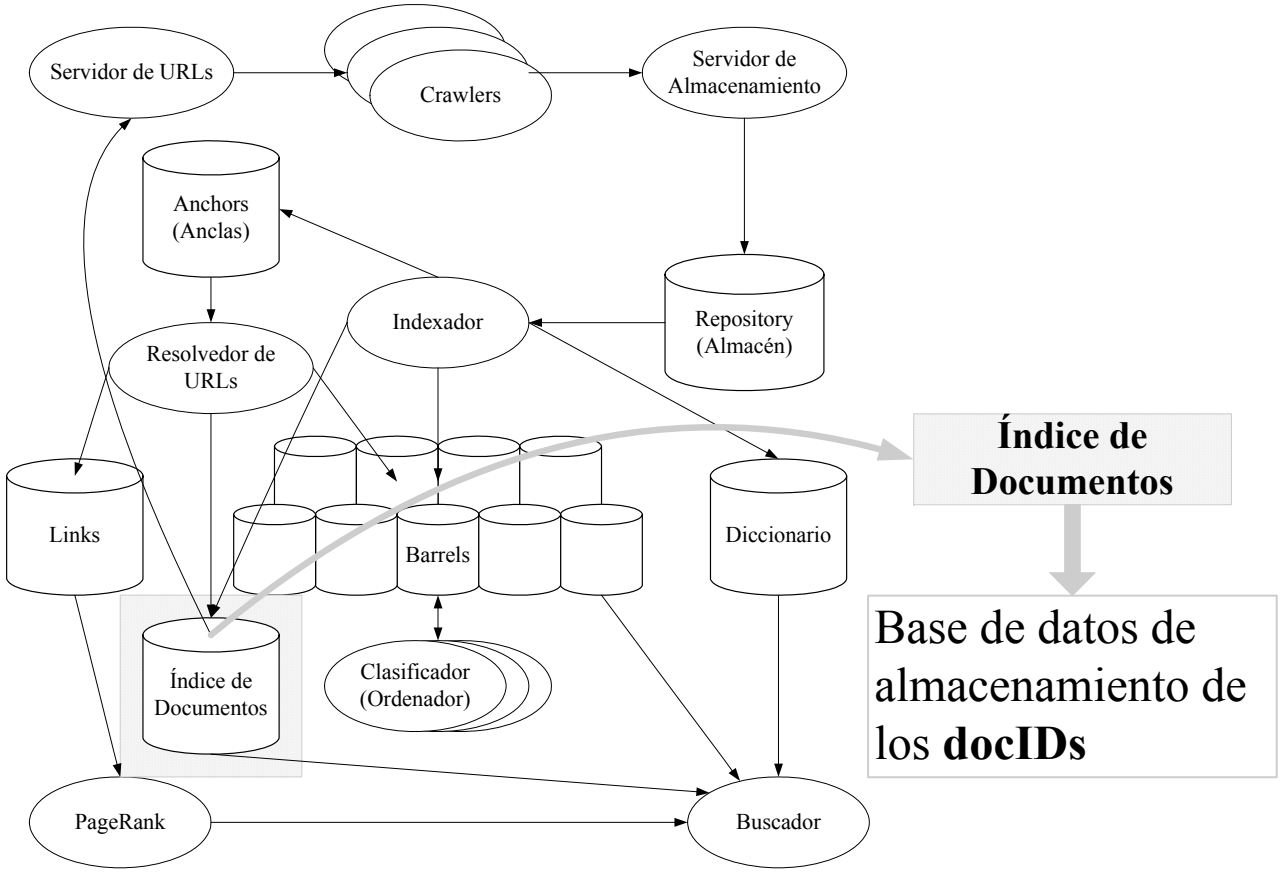
Arquitectura de Google (cont.)



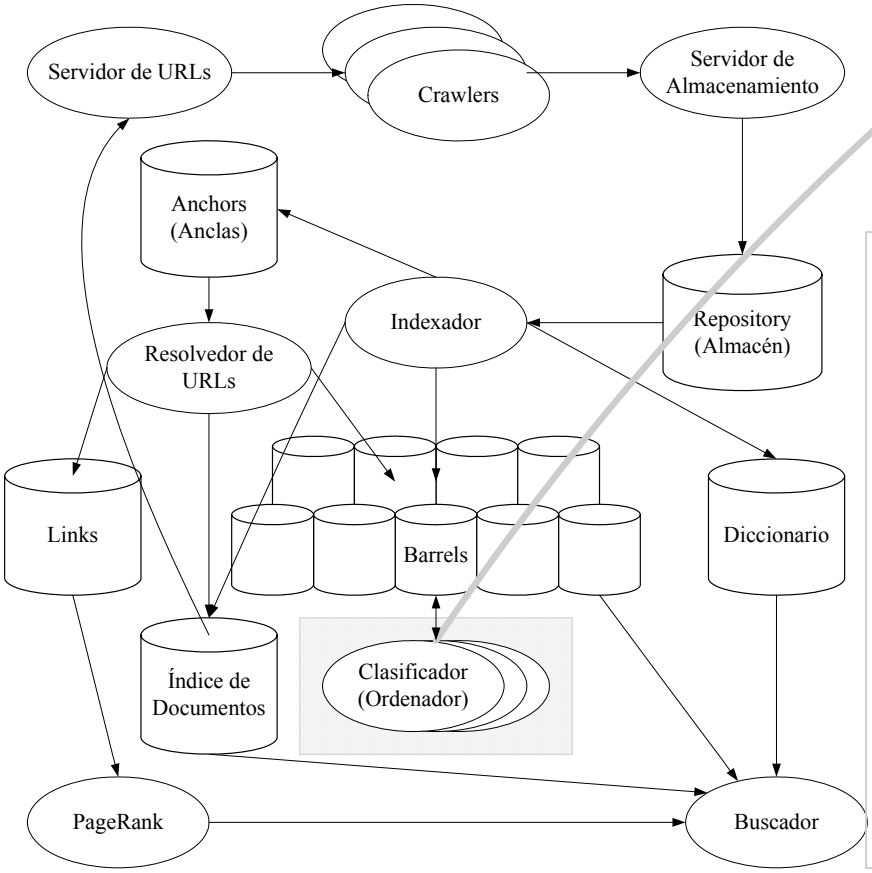
Diccionario

Almacena términos junto con sus identificadores. Es utilizado por el *indexador* para construir los *hits*

Arquitectura de Google (cont.)



Arquitectura de Google (cont.)

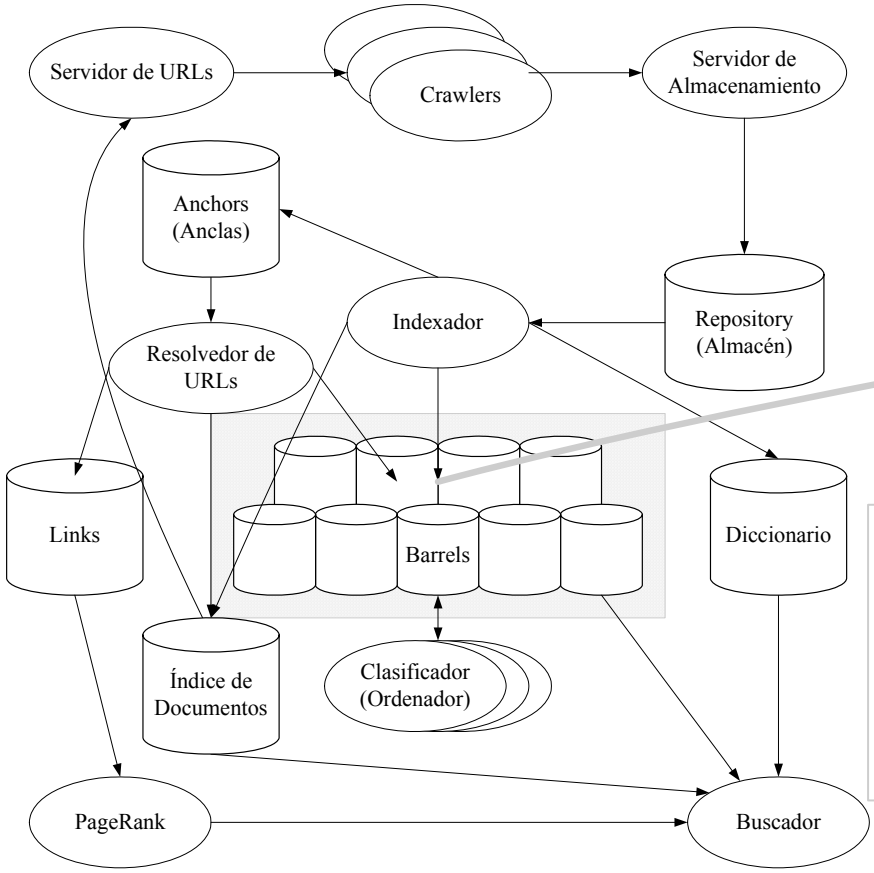


Clasificador (Ordenador)

Genera un índice invertido de los *hits* existentes en los *barrels*:

1. Asigna a cada término existente un identificador **wordID**.
2. Convierte el índice de **docIDs** en uno de **wordIDs**.

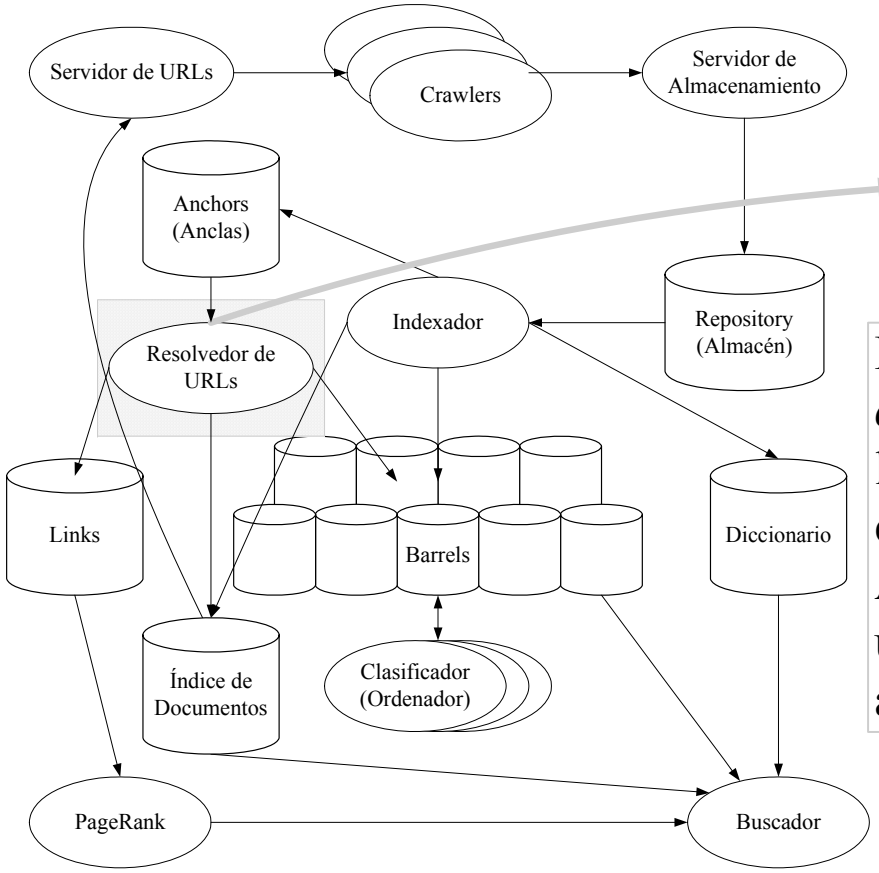
Arquitectura de Google (cont.)



Barrels

Conjunto de mini-almacenes en los que se guardan los *hits* generados por el indexador

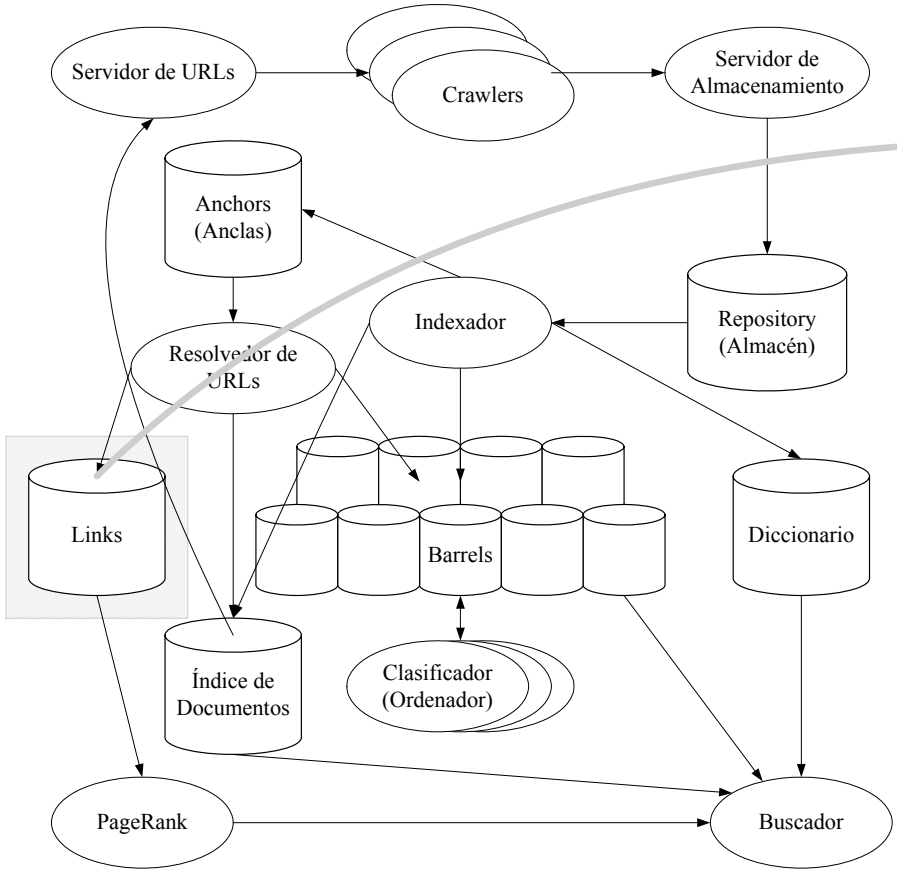
Arquitectura de Google (cont.)



Resolvedor de URLs

Lee los *ficheros de enlace* y convierte los URLs relativos en absolutos. Además les asigna un **docID** y lo almacena

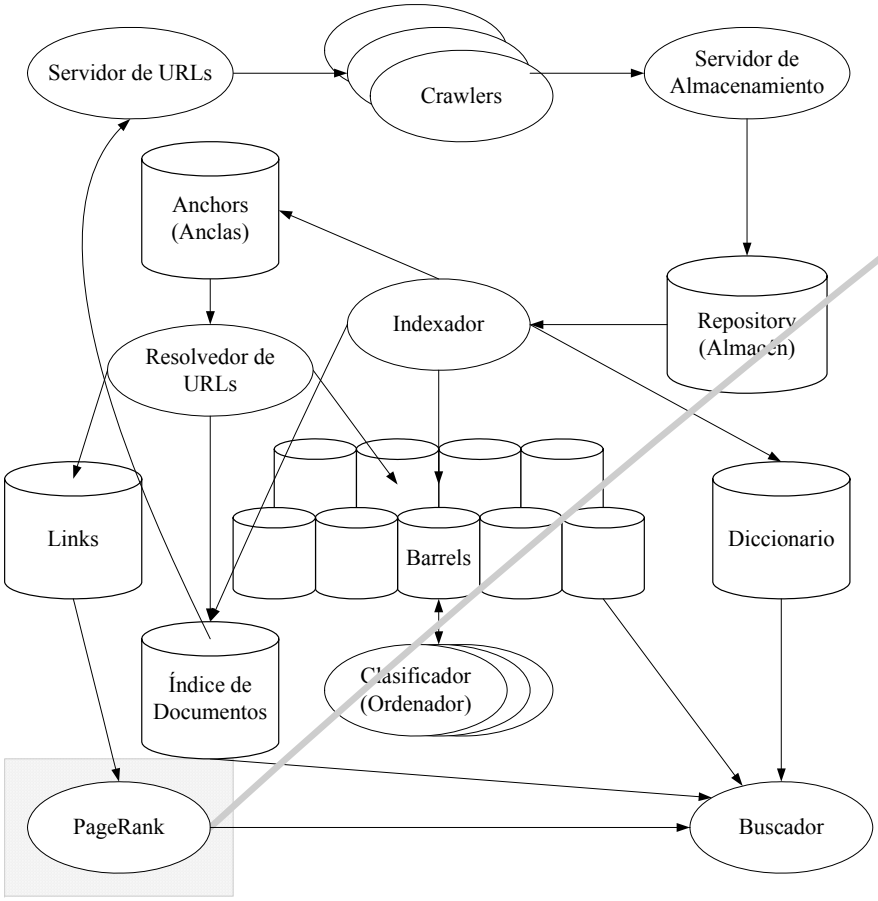
Arquitectura de Google (cont.)



Enlaces

Base de datos en que se almacenan pares (enlace,docID). Esta información será utilizada para calcular el *PageRank* de cada documento en las búsquedas.

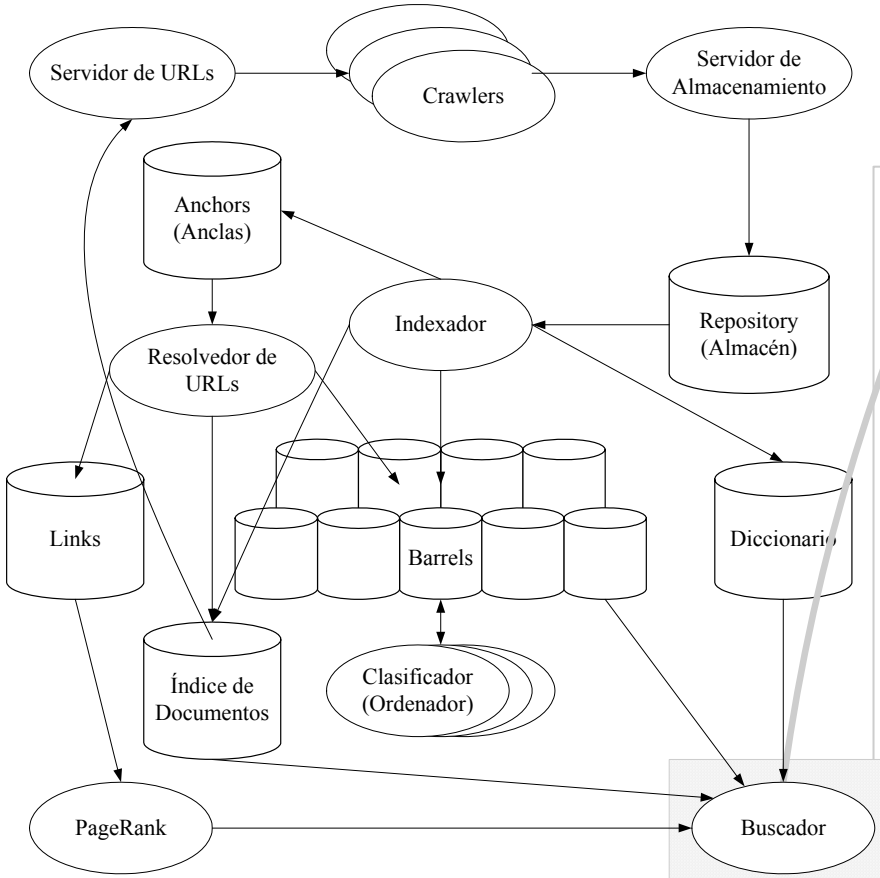
Arquitectura de Google (cont.)



PageRank

Encargado de calcular el *PageRank* de cada documento que cumple las restricciones impuestas en una búsqueda. Este cálculo se realiza basándose en la información almacenada en la base de datos de *Enlaces*.

Arquitectura de Google (cont.)



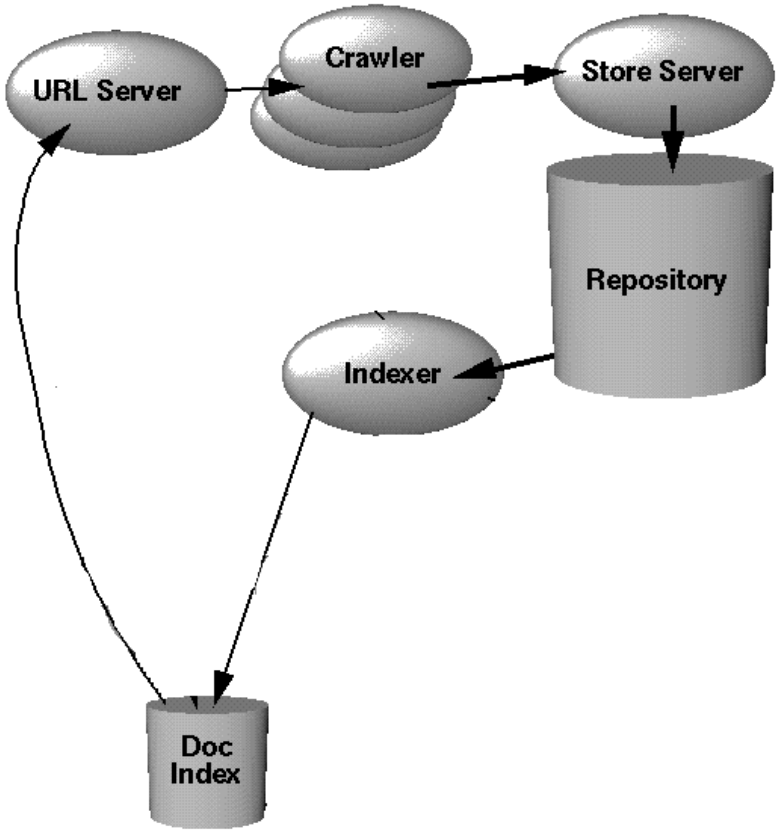
Buscador

Se ejecuta en un servidor Web y es el encargado de realizar las búsquedas para responder a las peticiones de los clientes. Utiliza el *vocabulario* generado por el *indexador*, el *índice invertido* generado por el *clasificador* y la información sobre el *PageRank* para calcular la relevancia de cada documento.

Crawling

- El proceso comienza por la captura las primeras URL por parte de los *crawlers*, normalmente se utilizan tres *crawlers* que mantiene abiertas más de 300 conexiones.
- Estos *crawlers* llegan hasta los servidores y obtiene las páginas web en HTML. Luego, mandan estas páginas al Servidor de Almacenamiento, el cual las comprime y las guarda en el repositorio.
- Una vez que el indexador ha analizado estás páginas y generado el docID del documento, el Servidor de URLs se encarga de obtener las nuevas URL del Índice de Documento. Estas URL son adheridas a los *crawlers* para ser rastreadas.

Crawling (cont.)



Crawling (cont.)

- Sistema de *crawlers* distribuidos.
- El servidor de URLs proporciona una lista de URLs a cada uno de los *crawlers*.
- Cada *crawler* mantiene abiertas 300 conexiones al mismo tiempo.
- Hasta 100 págs.Web/seg, utilizando 4 *crawlers*.

Crawling (cont.)

- Tasa de datos alrededor de 600k/seg.
- Cada *crawler* mantiene una caché DNS propia.
- Mejora de rendimiento, ya que se reduce considerablemente el número de veces que el *crawler* tiene que acceder a un servidor de nombres (DNS) externo.

Indexación

- El Indexador es el proceso más pesado en el sistema.
- Empieza por leer y descomprimir los archivos dentro del Repositorio.
- Luego parsea el HTML y comienza a convertir cada palabra en *Hits*. Estos Hits guardan toda la información que necesita la estructura, y las palabras son mandadas al Diccionario.

Indexación (cont.)

- Para hacer el conteo de las palabras, el Indexador se vale de un conjunto de funciones que llevan por nombre *MapReduce*.
  - La función *Map* se encarga de tomar las palabras y emitirlas junto con una inicialización de una ocurrencia.
  - La función *Reduce* toma cada aparición de una palabra y la reduce sumando todas sus apariciones en el mapa.
- Lo bueno de estas funciones es que se pueden ejecutar en paralelo, así que el tiempo se recorta a un poco más de la mitad.

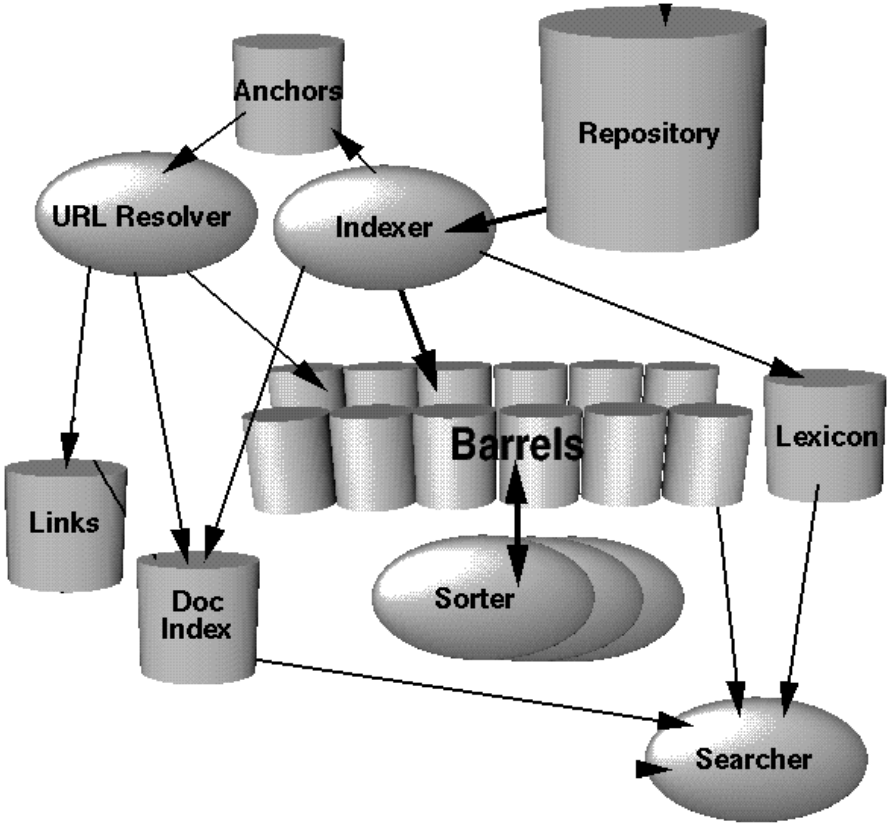
Indexación (cont.)

- Luego de haber barrido todo el documento, el Indexador distribuye todos los hits en los Barriles (*Barrels*).
- En paralelo, el Indexador parsea todos los links en la página que procesó y guarda toda la información necesaria en el archivo de enlaces.
- El Resolvedor de URL lee el archivo de enlaces y convierte cada link en un link absoluto (el link definitivo que apunta a la página HTML).
  - A estos URL absolutos, les aplica una técnica de hash para encontrar su *docID*.

Indexación (cont.)

- A continuación, se encarga de distribuir la información recolectada en los sitios pertinentes: pone el texto anclado en el Índice asociándolo con el *docID* al que éste apunta, y genera una base de datos de links la cual será usada para calcular el *PageRank* para cada documento.

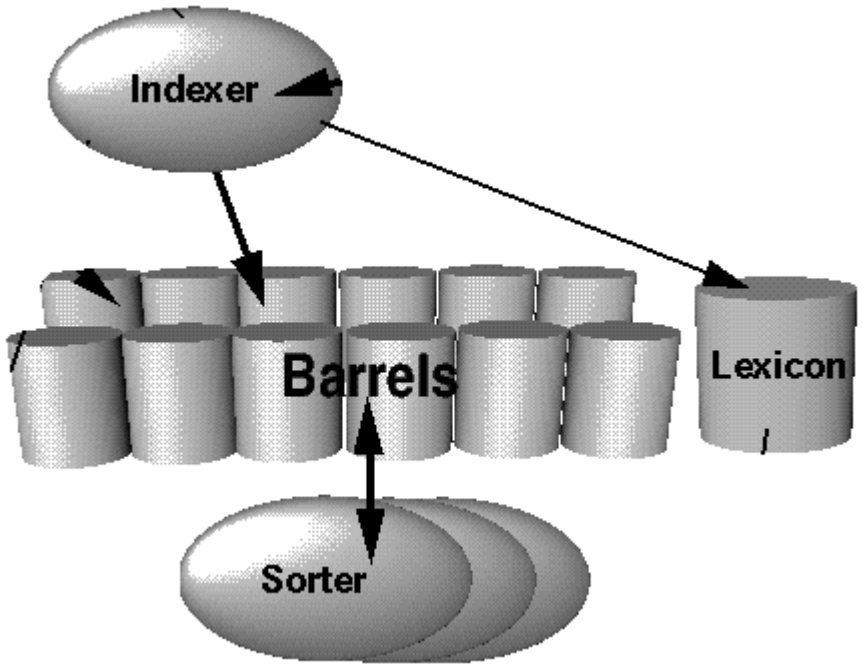
Indexación (cont.)



Clasificación

- La Clasificación es la parte donde se genera el Índice Invertido.
- El Ordenador (*sorter*) ordena lo barriles por *wordID* (originalmente semiordenados por *docID*).
- Luego, se produce una lista de *wordID* y *offsets* en la lista invertida.
- Un programa llamado *DumpLexicon* toma el léxico (diccionario) generado por el Indexador, y las cruza con estas listas, generando un nuevo léxico que podrá usar el Buscador.

Clasificación (cont.)



Búsqueda

- Se proponen los siguientes pasos para la búsqueda:
  - Se parsea la consulta.
  - Se convierten cada palabra en un *wordID*.
  - Se busca en el inicio de cada *docList* en los barriles pequeños cada una de las palabras.
  - Se sigue buscando por toda la lista hasta que se encuentre un documento que contenga todas las palabras.
  - Se computa la prioridad para ese documento según la consulta.
  - Si se está en el barril pequeño y se está al final de la *docList*, se busca en los barriles grandes por cada palabra y se retorna al paso 4.
  - Si no se ha llegado al final de ninguna *docList*, regresar al paso 4.
  - Se ordenan los resultados por prioridad y se muestran los primeros *k* elementos.

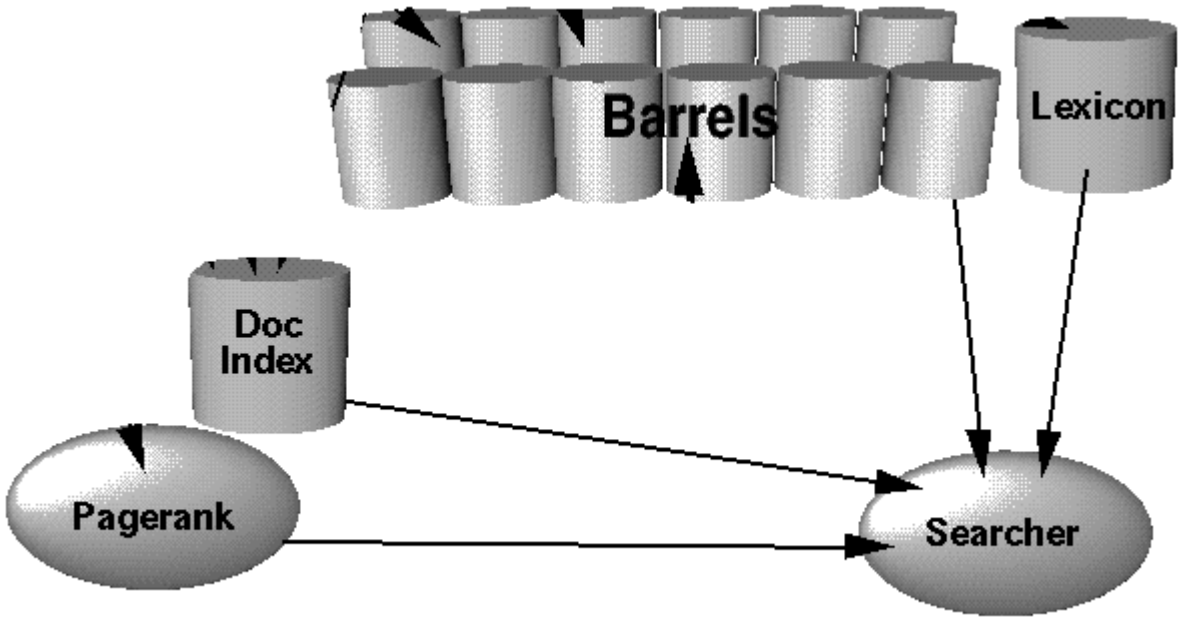
Búsqueda (cont.)

- El cálculo de la prioridad se vuelve un problema matemático bastante complejo. Se pueden dar dos casos con diferente complejidad al momento de parsear la consulta:
  - Si se considera el caso más sencillo, una sola palabra en la consulta, se busca en la *Hit List* de cada documento por esa palabra; cada hit contiene su propio tipo, y cada tipo tiene su valor. El Tipo-Peso hace un vector indexado por tipo; se cuenta el número de hits del tipo en la *Hit List* y así se obtiene otro vector Cantidad-Peso. Si se hace el producto punto de ambos vectores, se habrá computado un puntaje llamado IR para este documento. Finalmente este IR se combina con el *PageRank* para dar la prioridad final al documento.

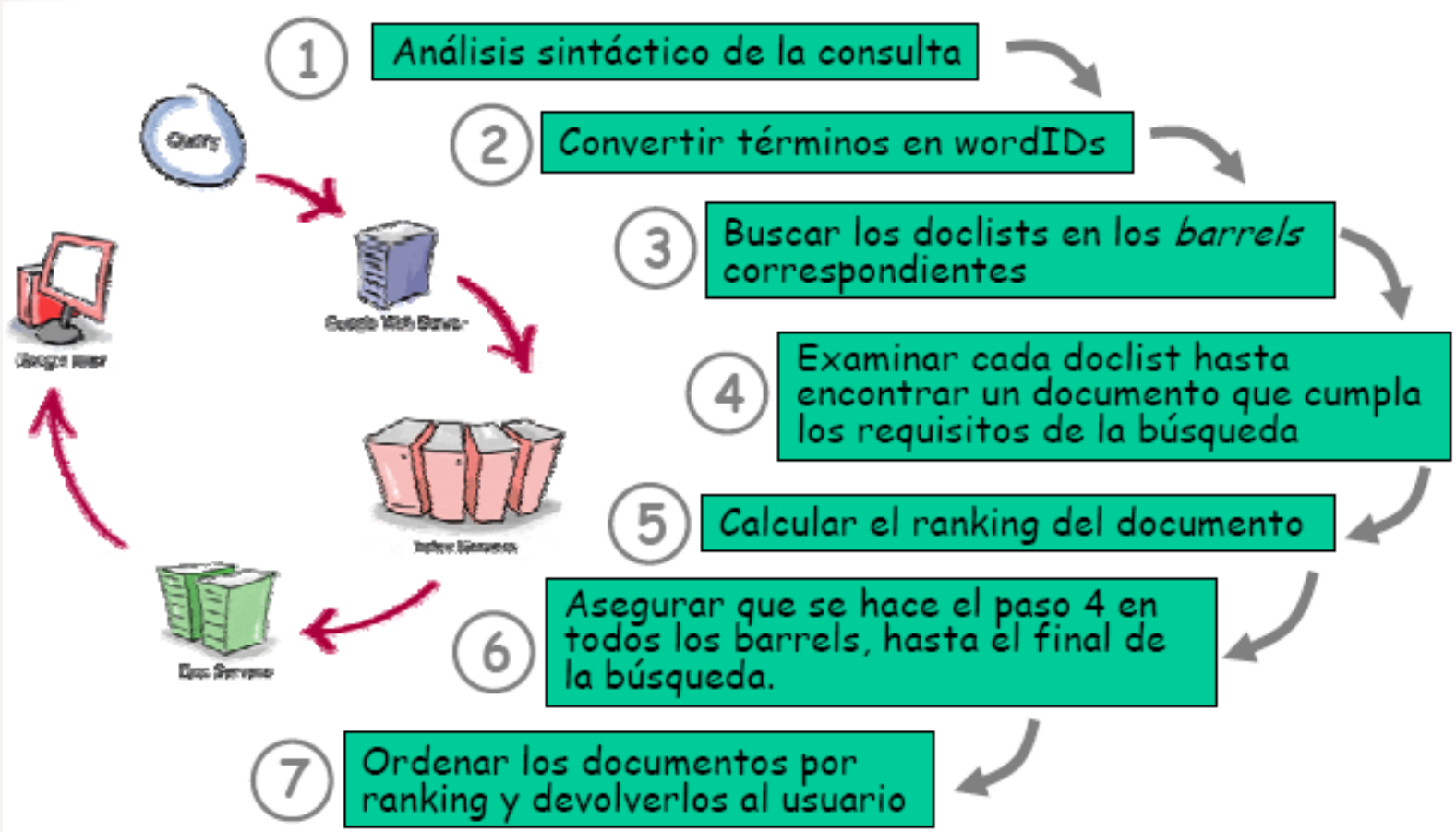
Búsqueda (cont.)

- Se pueden dar dos casos con diferente complejidad al momento de parsear la consulta (cont.):
  - Cuando son dos o más palabras, la cosa se complica aún más. Ahora se deben revisar múltiples *Hit List* al mismo tiempo para que cuando se encuentren dos hits cercanos, se les pueda dar prioridad mayor. Una vez que se han encontrado incidencias de palabras cercanas, se empieza a computar una aproximación. Ahora el conteo no se hace solo por cada tipo de *hit*, sino que también por cada tipo de proximidad. Con esto se obtiene los vectores Tipo-Proximidad-Peso y Conteo-Peso. Y luego se computa el IR haciendo el producto cruz entre ambos vectores.

Búsqueda (cont.)



Búsqueda (cont.)



Búsqueda (cont.)

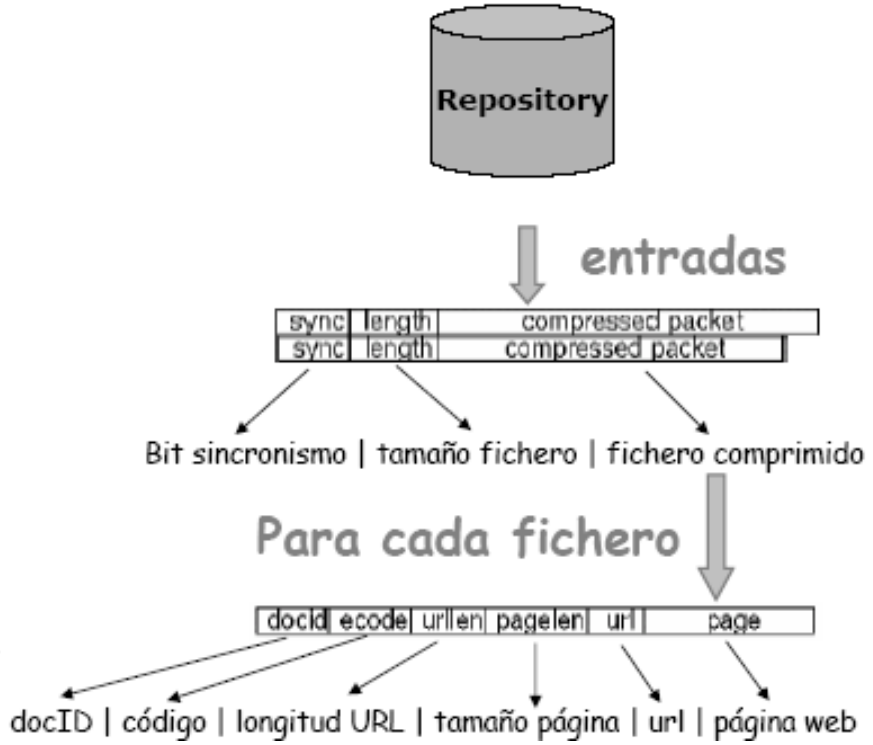
- *Ranking* ordenado y ponderado de acuerdo al *PageRank* de cada página.
- Prioridad de la calidad de las búsquedas sobre la eficiencia (en tiempo) de las mismas.
- Límite del tiempo de respuesta: una vez que se ha encontrado un número determinado de documentos (40.000, 1998) se devuelven resultados parciales.

BigFiles

- Paquete software para gestión de ficheros.
- Ficheros virtuales que abarcan varios tipos de sistemas de ficheros.
- Direccionables por enteros de 64 bits.
- Gestiona de forma automática la asignación en múltiples sistemas de ficheros.
- Gestiona también la asignación y desasignación de los descriptores de los ficheros (SO no lo hace de manera adecuada).
- Proporciona opciones básicas de compresión.

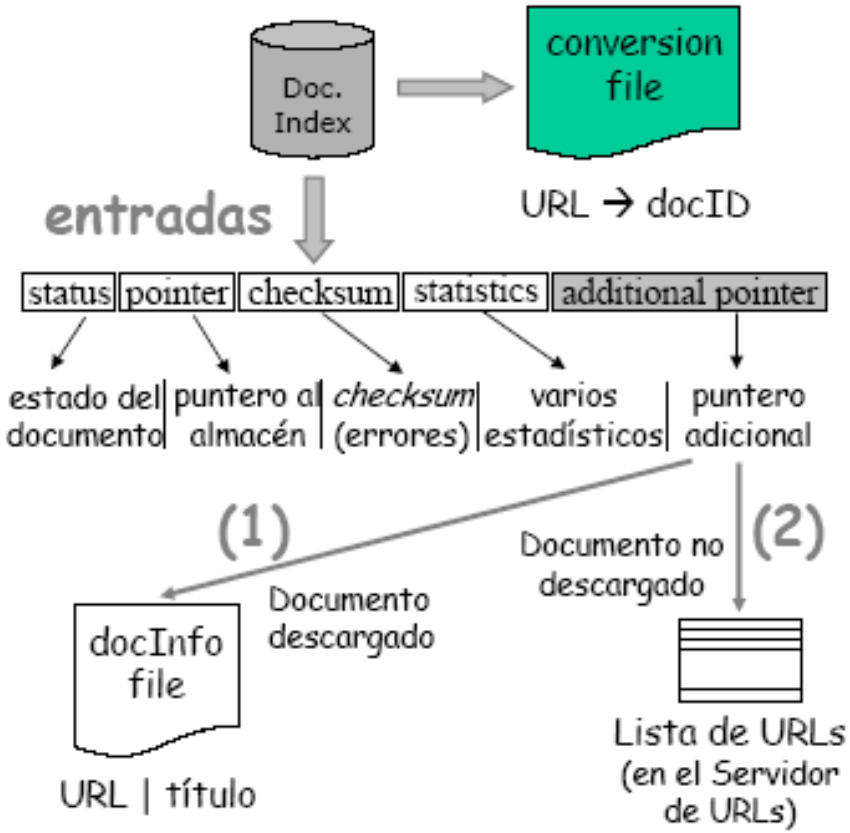
Almacén

- Contiene el código HTML de cada página descargada.
- Se almacena la información comprimida con formato zlib (RFC 1950).
- Estructura de datos simple: ayuda a consistencia de datos y facilita el desarrollo.



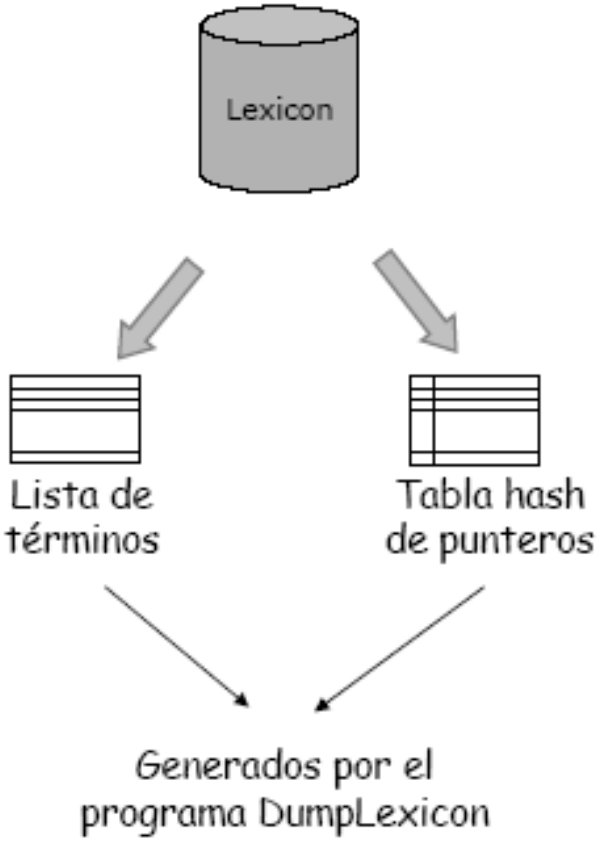
Índice de Documentos

- Almacena información sobre cada documento.
- Índice ISAM (*Index Sequential Access Mode)* ordenado por el **idDoc**.



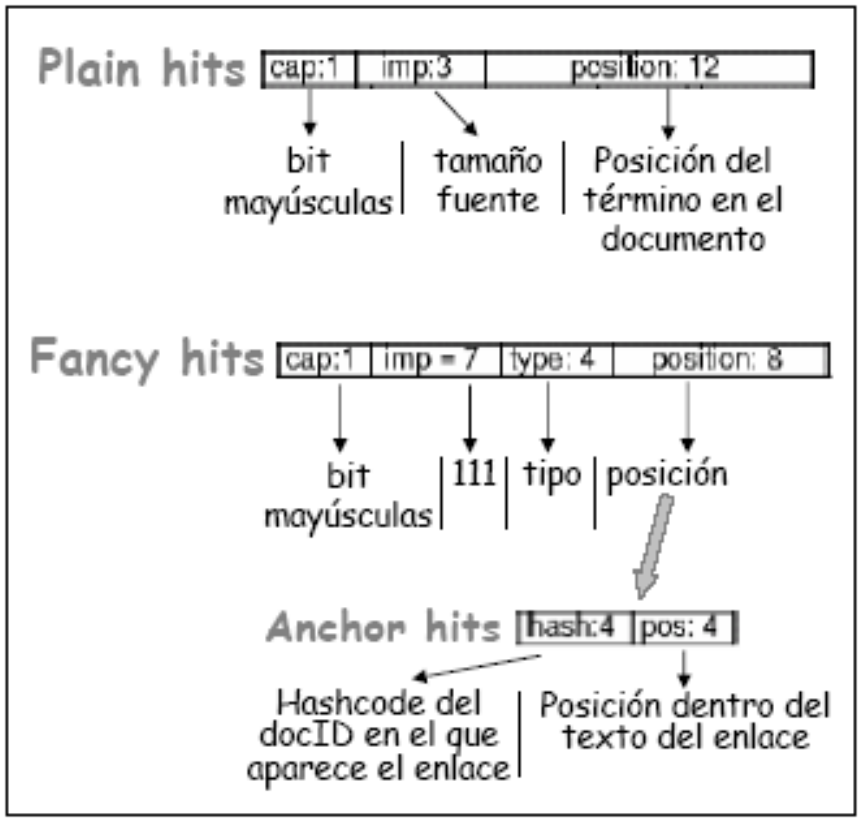
Diccionario

- Base de datos de los términos existentes en los documentos.
- En 1998 contenía 14 millones de entradas.



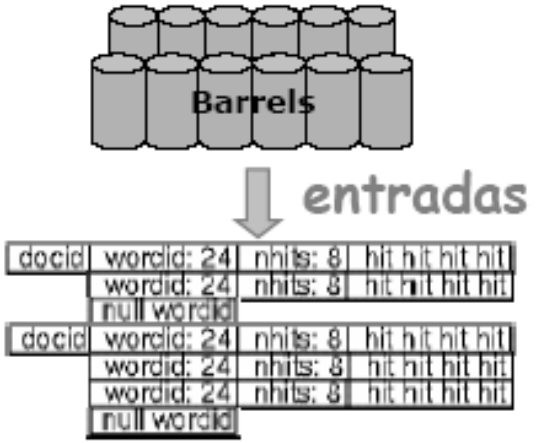
Hits

- Cada *hit* 2 bytes.
- 2 tipos:
  - *Fancy hits* (URLs, texto enlaces, etiquetas meta, título).
  - *Plain hits* (resto).



Índice Directo

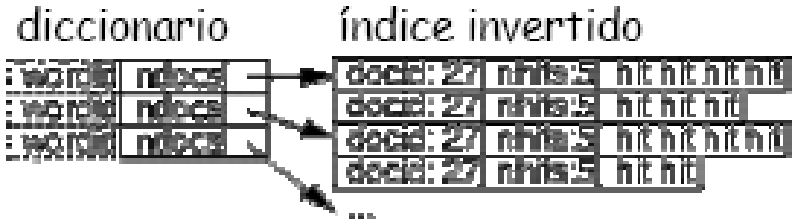
- Índice ordenado almacenado en los *barrels* (actualmente 64).
- Cada *barrel* un rango de **wordIDs** (identificadores numéricos).



docid	docID del documento
wordid: 24	Lista de wordIDs correspondientes a los términos de los *hits* de cada entrada
nhits: 8	Longitud total de la lista de hits almacenados a continuación
hit hit hit hit	Lista de hits correspondientes a los wordIDs indicados
null wordid	Separador de entradas

Índice Invertido

- Índice directo procesador por el clasificador.
- Se crea un **doclist** de **docIDs** que representa las ocurrencias de un término en todos los documentos en los que aparece.
- Se crean los punteros entre las entradas del diccionario y las entradas correspondientes en el **doclist**.



wordid	wordID del término
ndocs	Número de documentos en los que aparece
→	Punteros a las entradas correspondientes a dichos documentos en el doclist
docid: 27	docID del documento
nhits:5	longitud de la lista de hits que se almacenan a continuación
hit hit hit hit	lista de *hits* del documento

Datos Estadísticos 1998

Estadísticas de almacenamiento	
Tamaño total de páginas descargadas	147,8 GB
Almacén de páginas comprimidas	53,5 GB
Índice invertido (pequeño)	4,1 GB
Índice invertido (total)	37,2 GB
Diccionario	293 MB
Datos de enlaces (*anchors*) temporal	6,6 GB
Índice de documentos	9,7 GB
Base de datos de enlaces	3,9 GB
Tamaño total sin el Almacén	**55,2 GB**
Tamaño total con el Almacén	**108,7 GB**

Datos Estadísticos 1998 (cont.)

Estadísticos de páginas Web	
Número de páginas descargadas	24 millones
Número de URLs visitados	76,5 millones
Número de direcciones de *e-mail*	1,7 millones
Número de páginas con error 404 (404: Page not found on this server)	1,6 millones

Implementación

- Los lenguajes de programación utilizados son:
  - La amplia mayoría de los módulos que componen la arquitectura están implementados en C y C++.
  - Ejecución sobre Solaris y Linux.
  - Los *Crawlers* y el Servidor de URLs están implementados en Perl.

Google Inc.

- Historia de Google y más:
  - http://es.wikipedia.org/wiki/Google.
  - The Anatomy of a Large-Scale Hypertextual Web Search Engine.
- El paraíso de Google:
  - http://www.elmundo.es/albumes/2008/03/07/google_zurich/.
- Nuevas herramientas:
  - http://www.desarrolloweb.com/actualidad/google-nuevas-herramientas-1785.html.

Google y otros SRI

- Comparación Google y Yahoo:
  - http://www.langreiter.com/exec/yahoo-vs-google.html.
- Comparación Google y Bing (Kumo):
  - http://www.bingandgoogle.com/.

Referencias Bibliográficas

- La información fue tomada de:
  - http://es.wikipedia.org/wiki/Google.
  - http://en.wikipedia.org/wiki/Google_platform.
  - https://en.wikipedia.org/wiki/PageRank.
  - http://www.maxglaser.net/arquitectura-original-de-google/.
  - https://volkanrivera.com/esp/2009/04/google-revela-la-arquitectura-de-sus-data-center/.
  - https://www.link-assistant.com/news/page-rank-2018.html.
  - https://www.humanlevel.com/diccionario-marketing-online/pagerank-google-.
  - SEO y Google.

Referencias Bibliográficas

- La información fue tomada de:
  - http://www.seomoz.org/blog/google-says-yes-you-can-still-sculpt-pagerank-no-you-cant-do-it-with-nofollow.
  - http://www.mattcutts.com/blog/pagerank-sculpting/.
  - http://www.desarrolloweb.com/actualidad/google-nuevas-herramientas-1785.html.
  - http://www.desarrolloweb.com/actualidad/google-squared-nueva-herramienta-categorizacion-datos-1905.html.
  - http://royal.pingdom.com/2009/03/02/original-google-setup-at-stanford-university/.
  - Google Stanford.