



Recuperación de Información en Internet:

---

# Estructura de Google

Martín Llamas Nistal

Nuevos Servicios Telemáticos  
Curso 2007-2008



# Contenidos

---

- Introducción
- Características de Google
- Arquitectura de Google
- Exploración de la web: “Crawling”
- Búsquedas
- Datos estadísticos y de implementación
- Conclusiones



# ¿Qué es un motor de búsqueda?

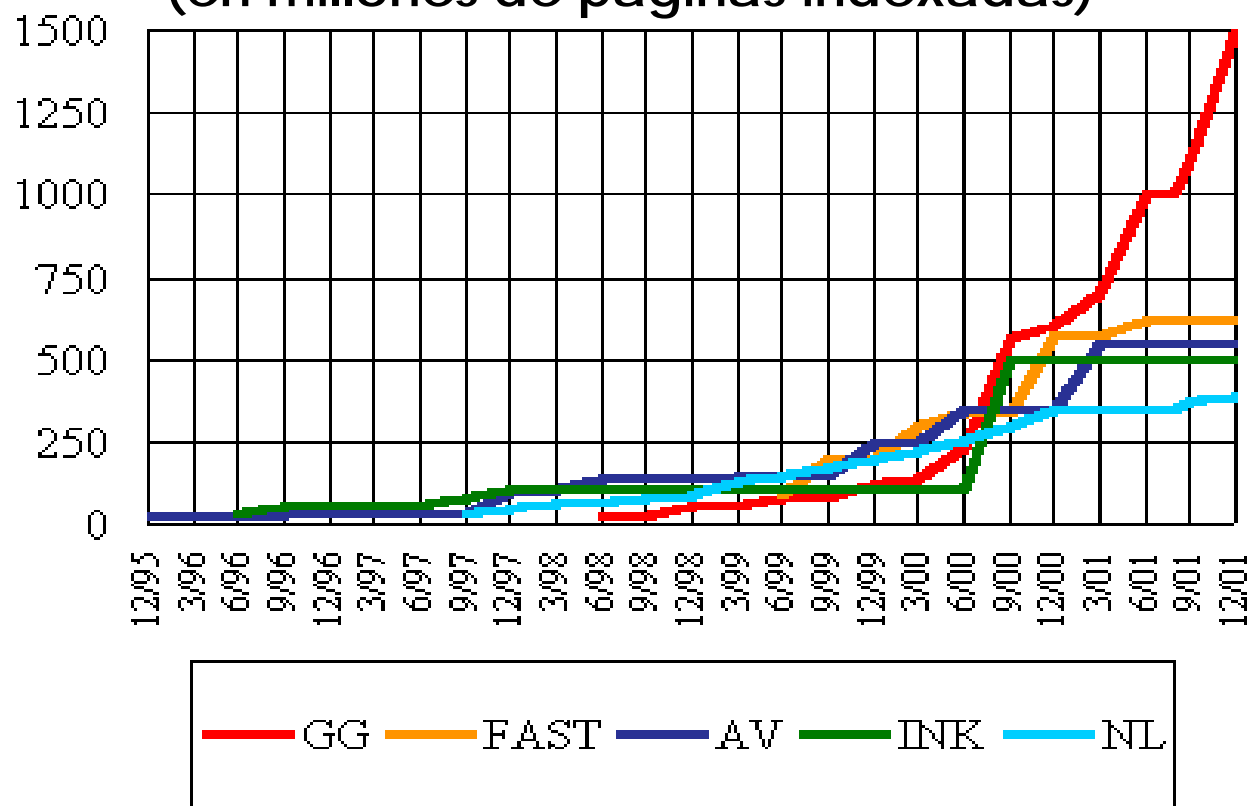
---

- **Motor de Búsqueda:** Un sistema de almacenamiento de datos (base de datos) diseñada para indexar direcciones web (url, ftp, etc.).
- **Ejemplos:** Google, Altavista, Excite, etc.
- **Servicio de Directorio:** igual que el motor de búsqueda, pero **la indexación se hace de forma manual.**
- **Ejemplo:** Yahoo



## Evolución histórica\* (I)

Tamaño de los Motores de Búsqueda Web  
(en millones de páginas indexadas)

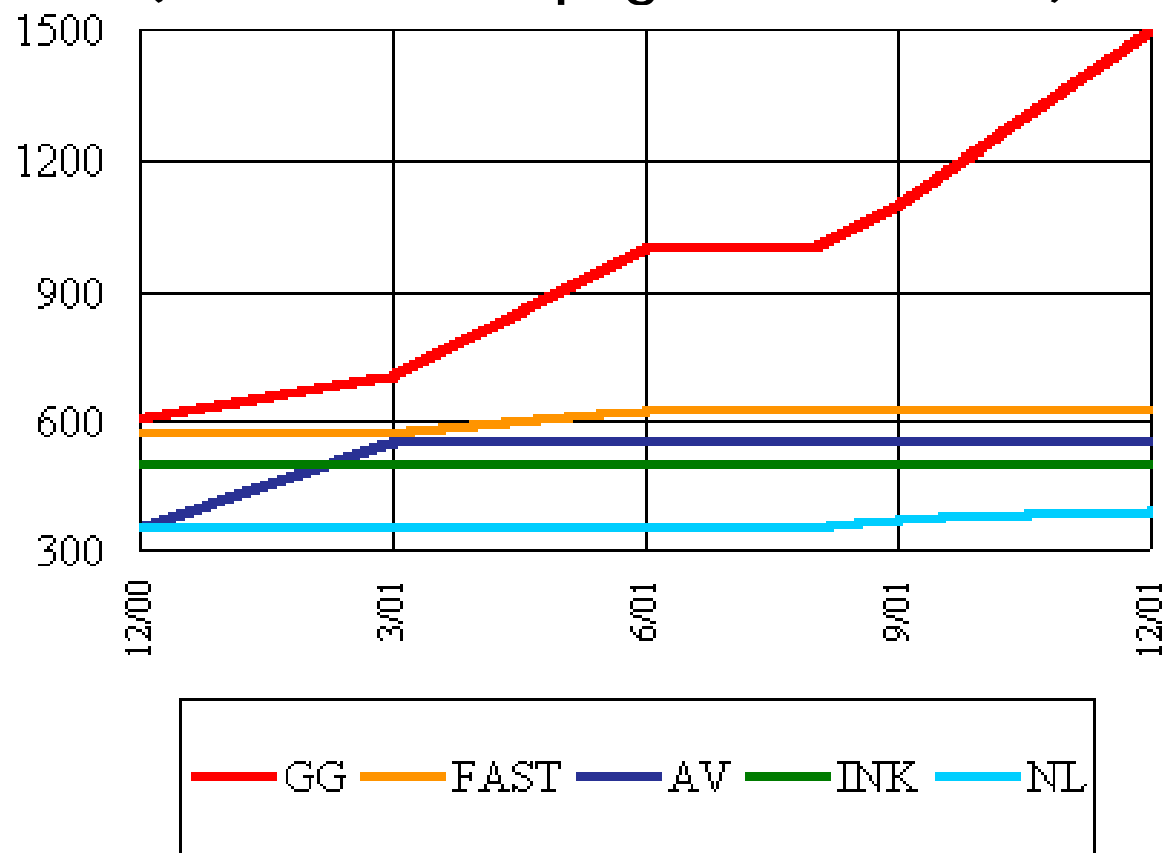


\* Fuente: *searchenginewatch.com* (Dic 2001)



## Evolución histórica\* (II)

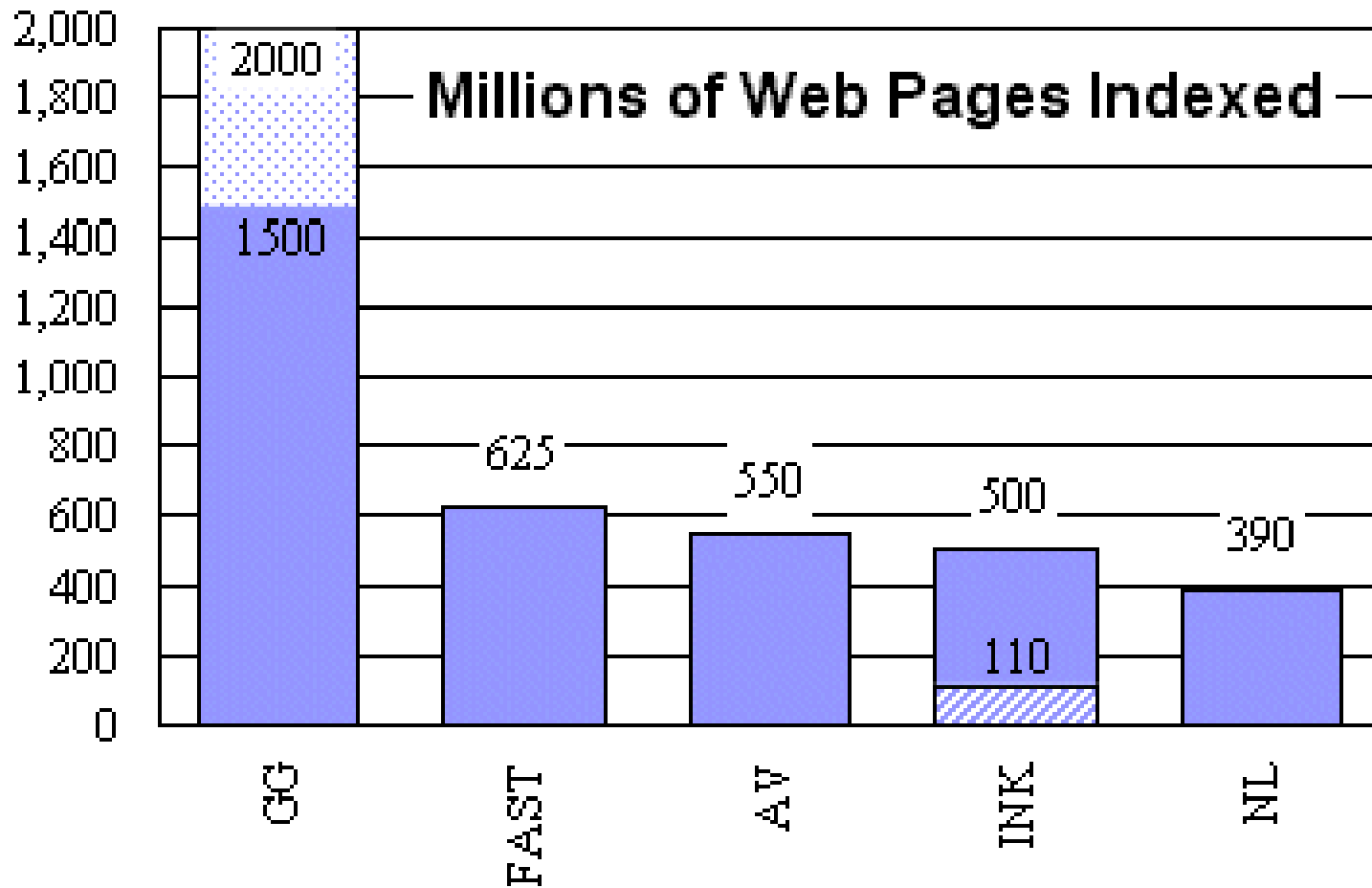
Tamaño de los Motores de Búsqueda Web  
(en millones de páginas indexadas)



\* Fuente: *searchenginewatch.com* (Dic 2001)



# Tamaño de los buscadores Web



\* Fuente: *searchenginewatch.com* (Dic 2001)



# Contenidos

---

- Introducción
- **Características de Google**
- Arquitectura de Google
- Exploración de la web: “Crawling”
- Búsquedas
- Datos estadísticos y de implementación
- Conclusiones



# Características de Google

---

- Utiliza la información hipertextual de los documentos Web para calcular la relevancia de cada página, utilizando lo que se denomina **PageRank**
- Utiliza los enlaces (*links*) y el texto de los mismos para mejorar los resultados de la búsqueda



## PageRank. *Cálculo*

---

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

- $r(i)$  es el PageRank de la página  $i$
- $N(i)$  es el número de enlaces (salientes) de la página  $i$
- $B(i)$  es el número de páginas que apuntan a la página  $i$
- $m$  es el número total de nodos en el grafo
- $d$  es el factor de decaimiento (entre 0 y 1)



# PageRank

---

Recordamos que:

- 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 PR para una página será alto:
  - **Si existen muchas páginas apuntándola**
  - O aunque la apunten pocas páginas, **éstas tienen PR alto.**



## Texto de los enlaces

---

- 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
- Ventajas/inconvenientes:
  - 👍 El texto de los enlaces, con frecuencia, proporciona descripciones sobre el contenido de las páginas
  - 👍 Pueden existir enlaces a documentos (imágenes, programas, direcciones de *e-mail*, etc.) que no pueden ser indexados por motores de búsqueda textuales
  - 👍 Permite devolver documentos en las búsquedas que no han sido rastreados
  - 👎 Pueden devolver páginas inexistentes



## Características adicionales

---

- Mantiene información de la posición de los términos que aparecen dentro de los documentos indexados, lo que permite búsquedas por proximidad (**aunque luego no la hace**)
- Mantiene información de la apariencia visual de los documentos (p.e: a las palabras marcadas en negrita o con un tamaño de letra mayor se les concede mayor peso al calcular la relevancia)
- El código HTML plano de los documentos **está disponible** en los almacenes de Google



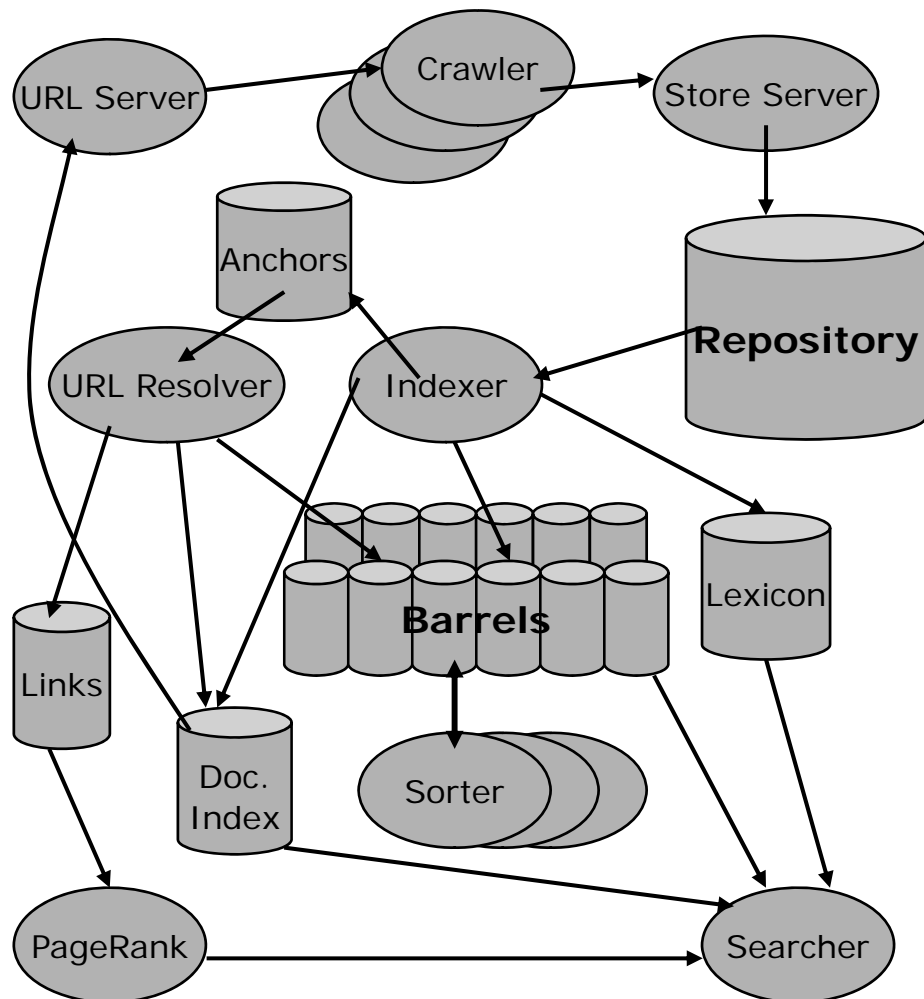
# Contenidos

---

- Introducción
- Características de Google
- **Arquitectura de Google**
- Exploración de la web: “Crawling”
- Búsquedas
- Datos estadísticos y de implementación
- Conclusiones



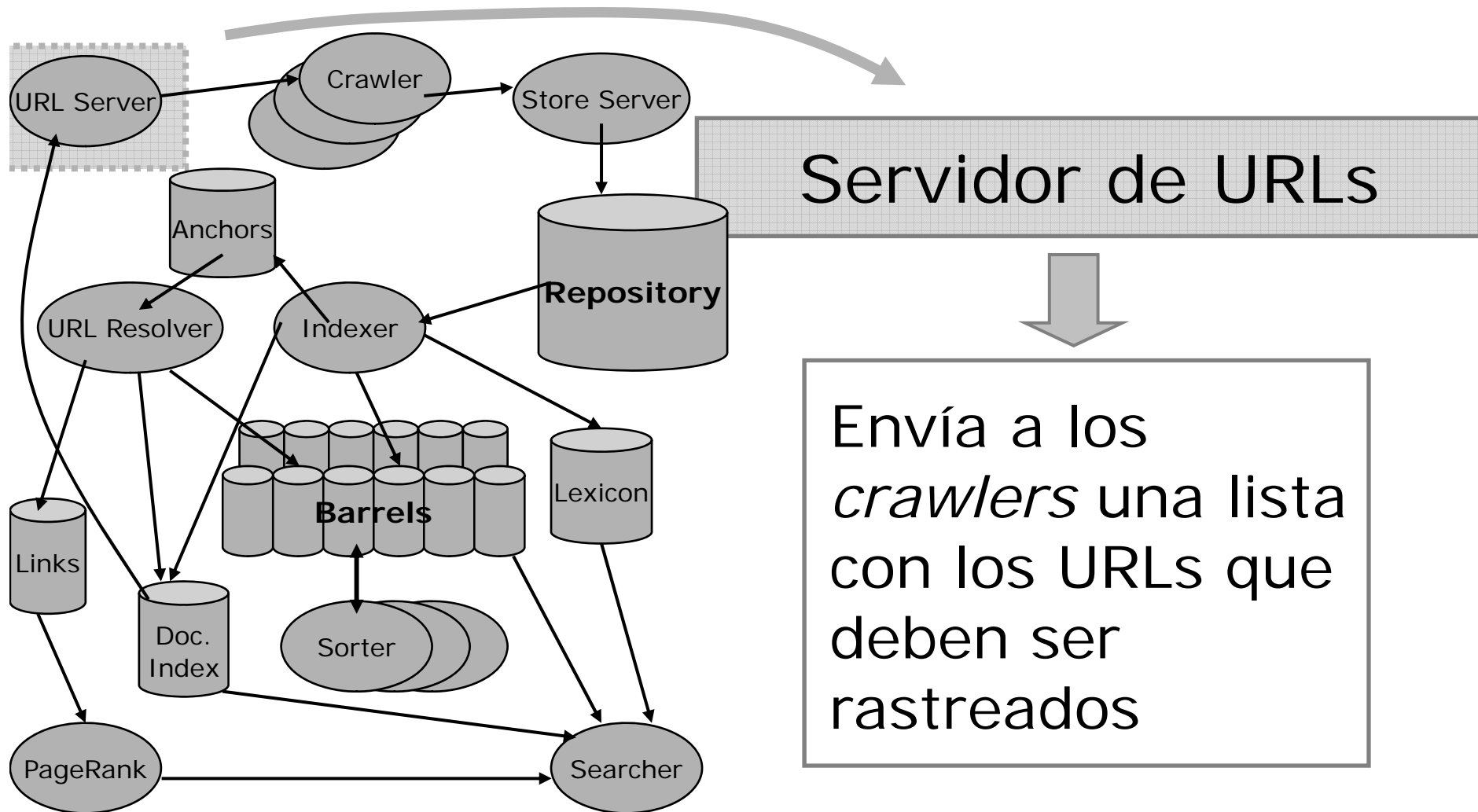
# Arquitectura





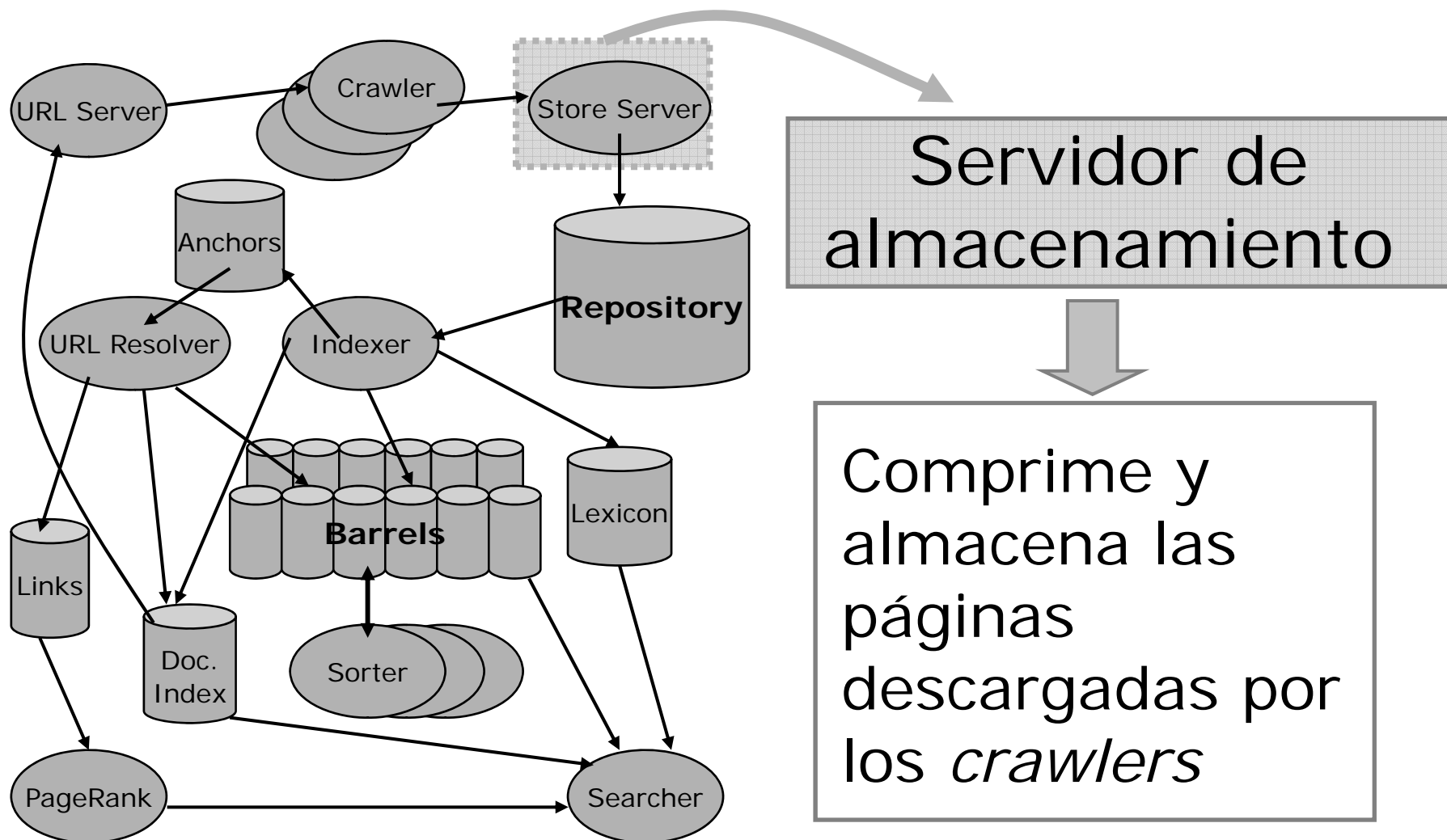


# Arquitectura: Servidor de URLs



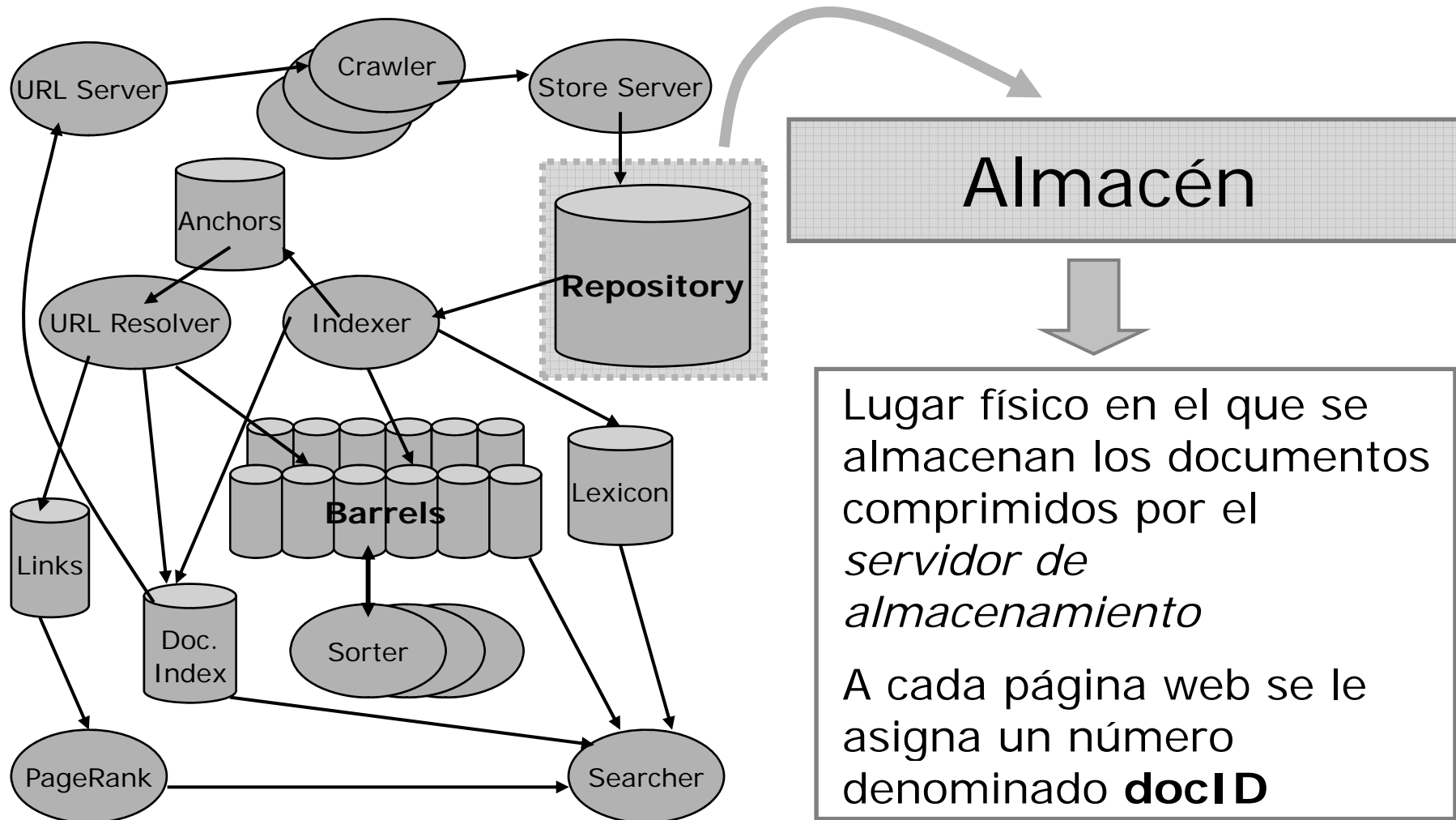


# Arquitectura: Servidor de Almacenamiento



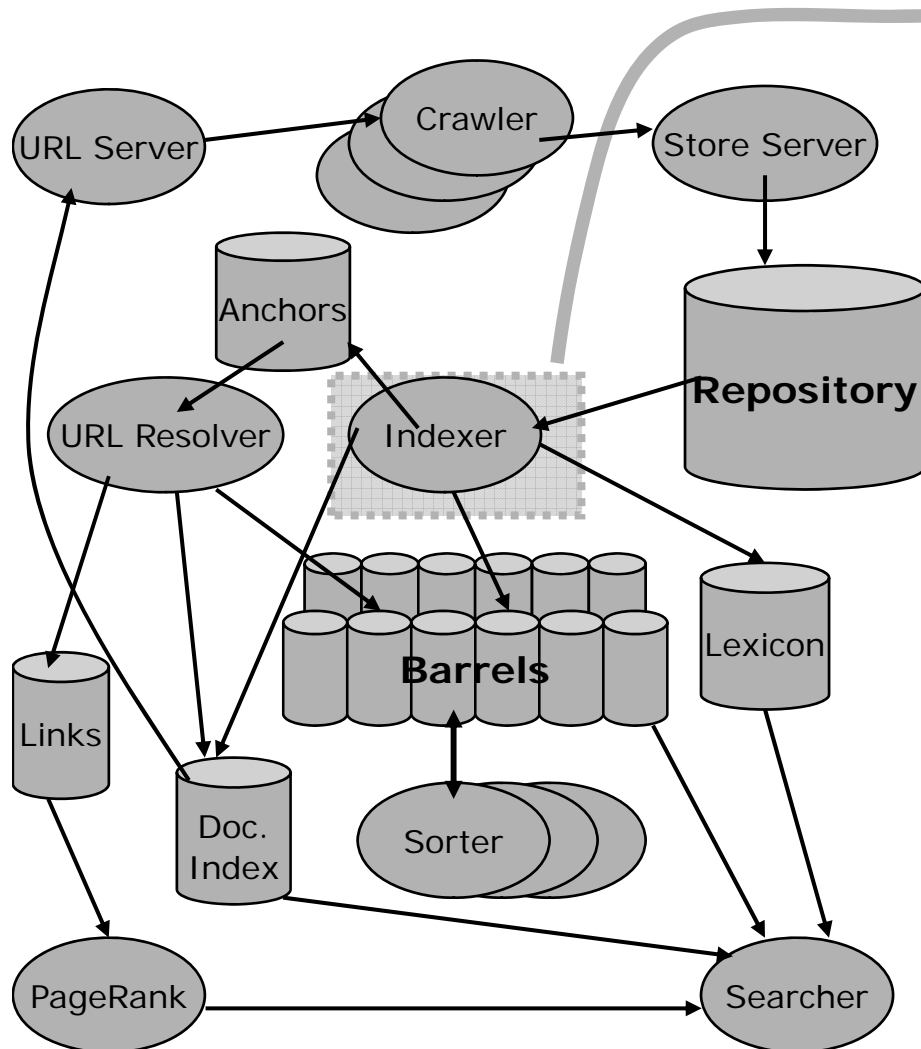


# Arquitectura: Almacén





# Arquitectura: Indexador



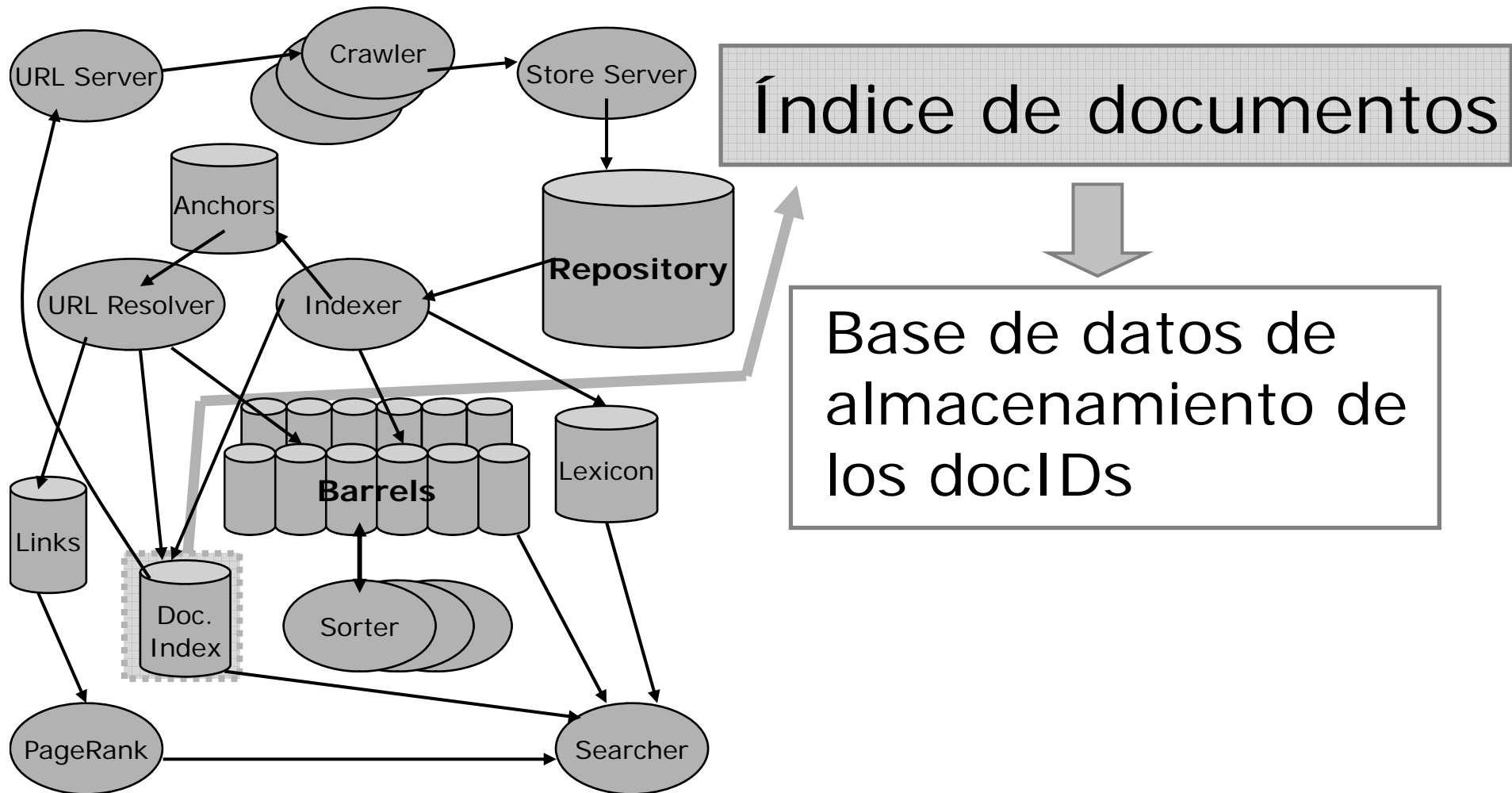
## Indexador

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**"

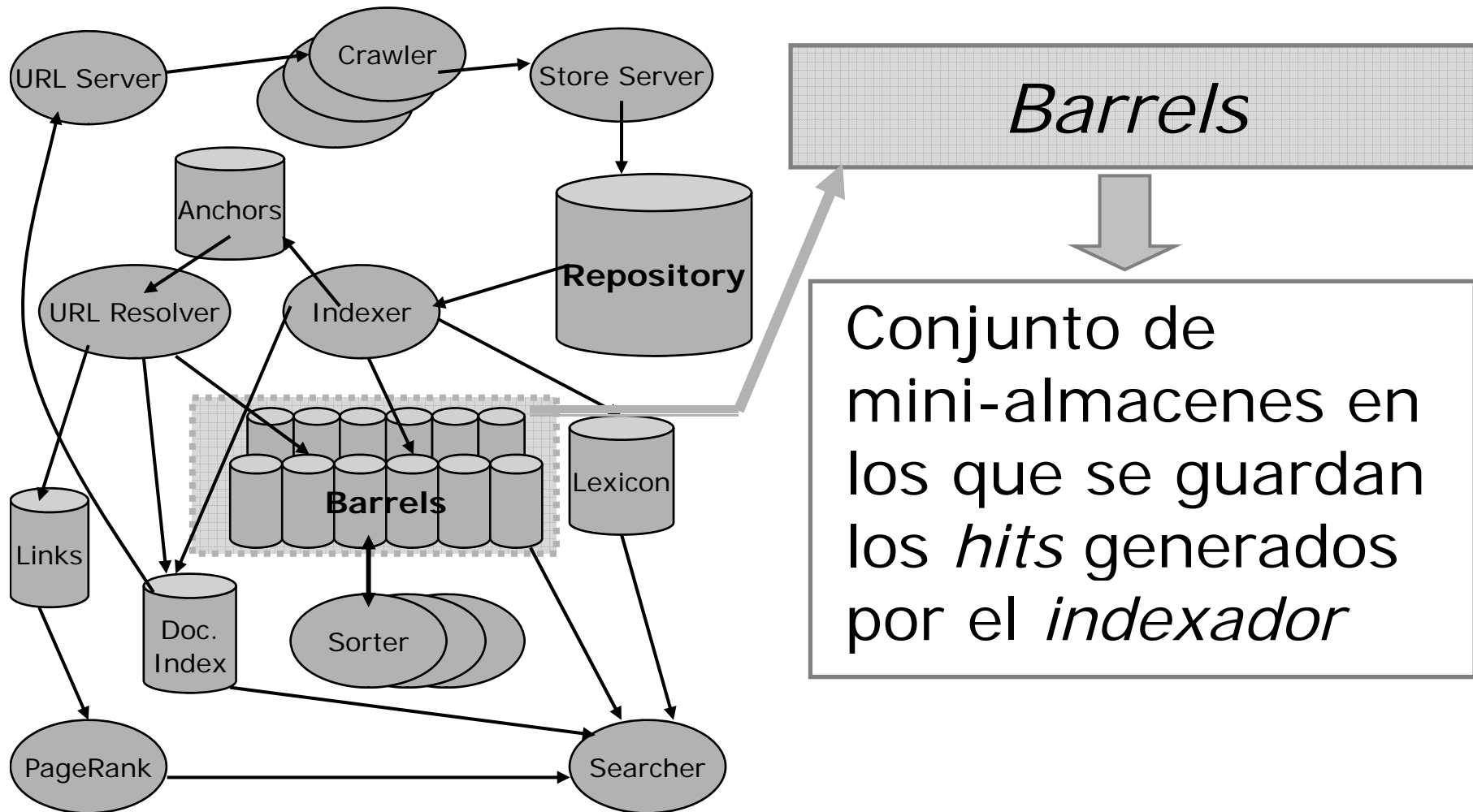


# Arquitectura: Índice de documentos



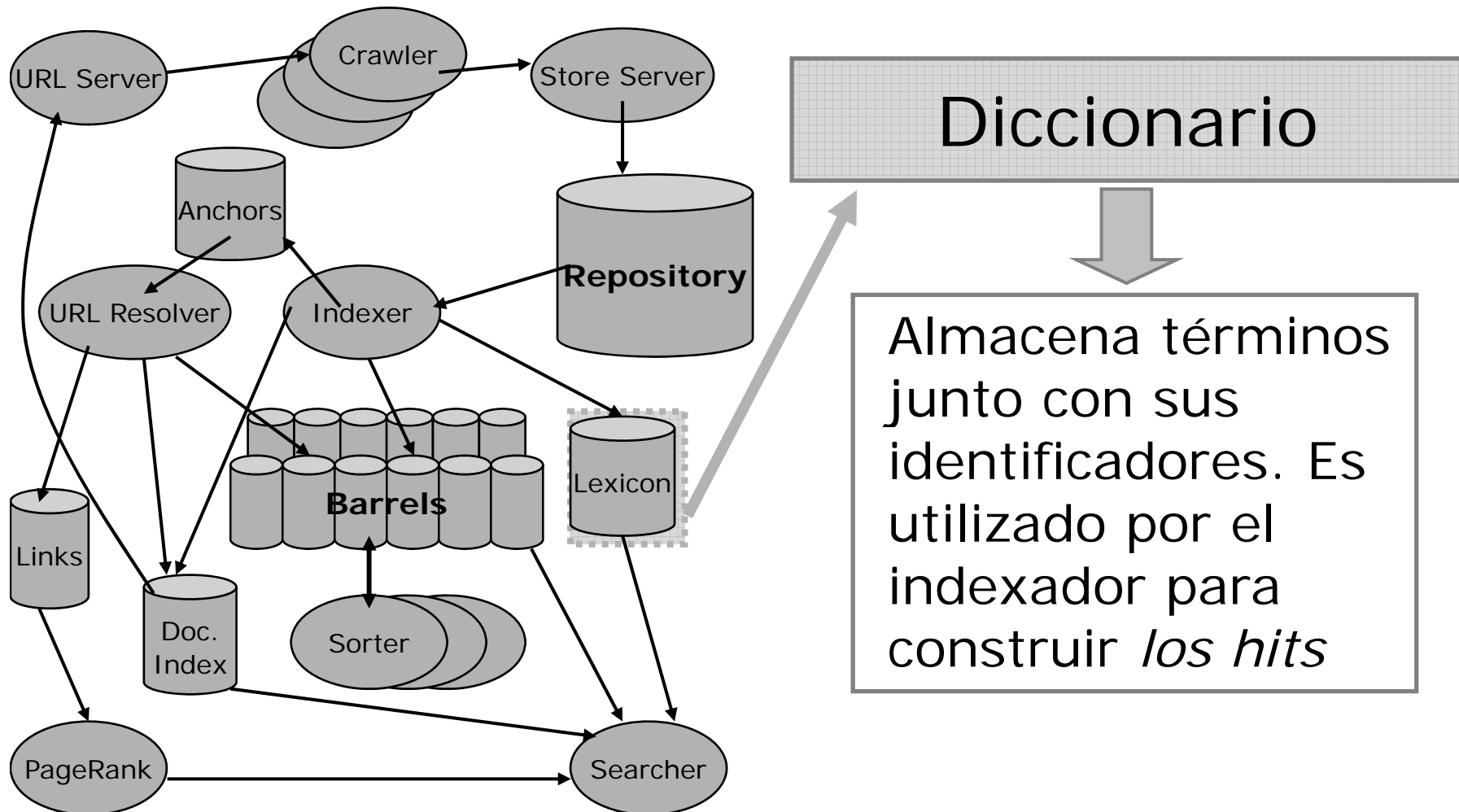


# Arquitectura: Barrels



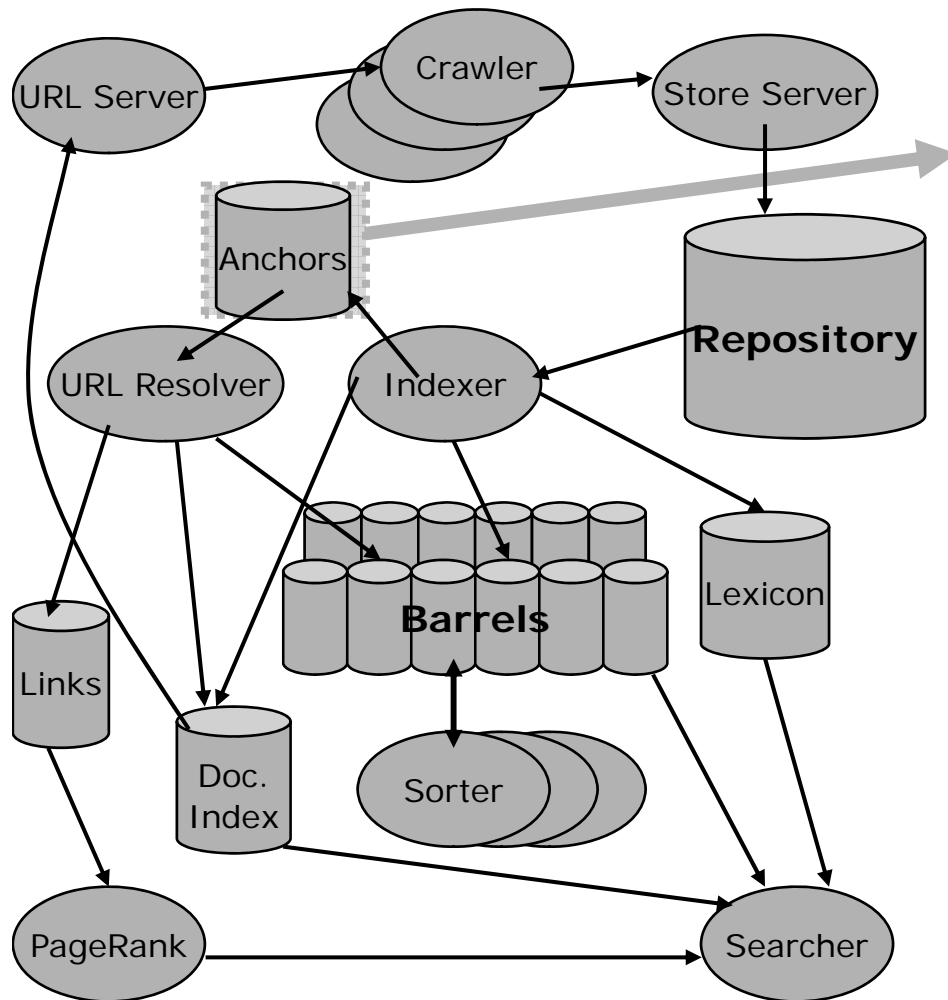


# Arquitectura: Diccionario





# Arquitectura: Anclas

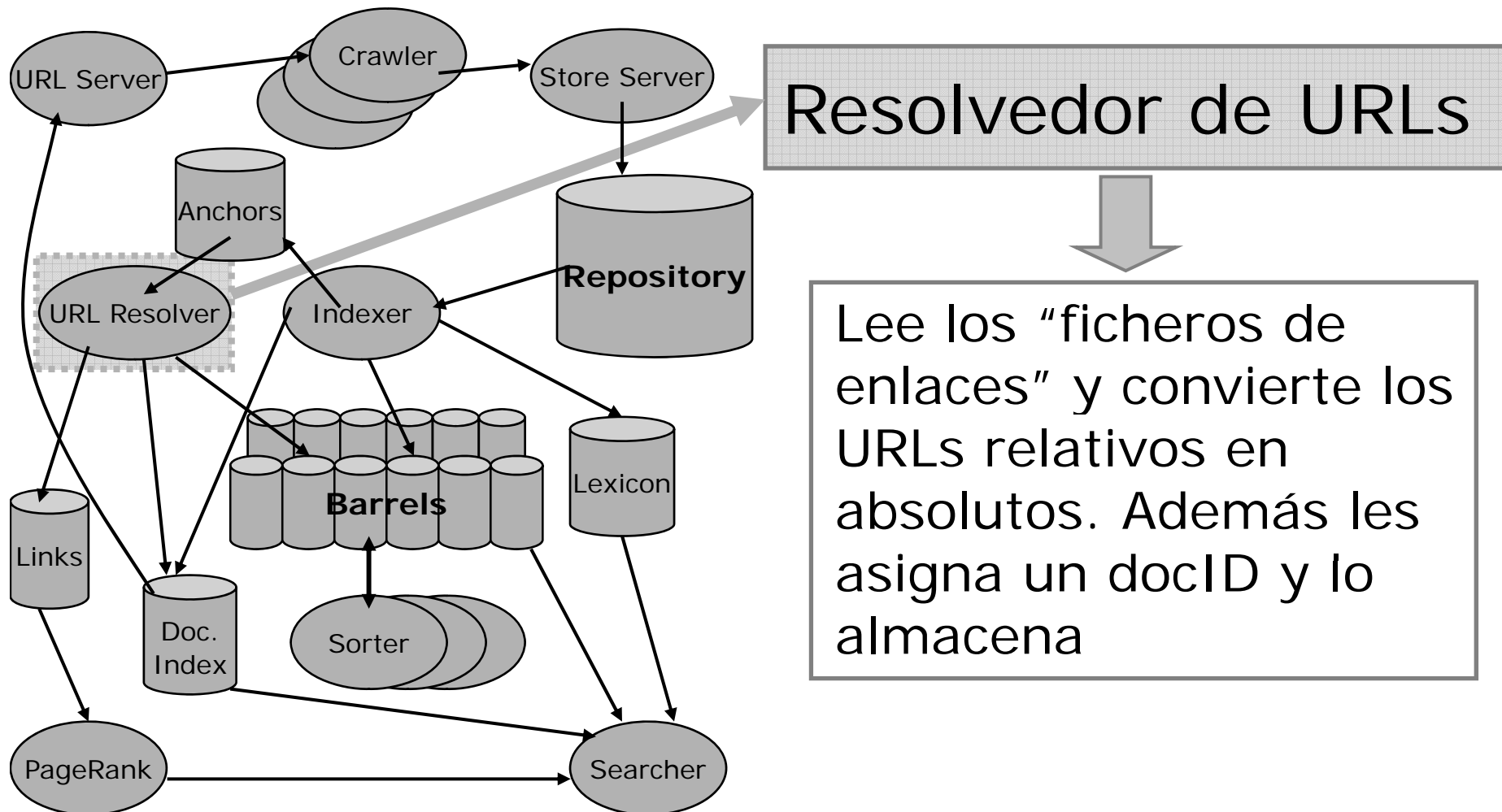


Anclas

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

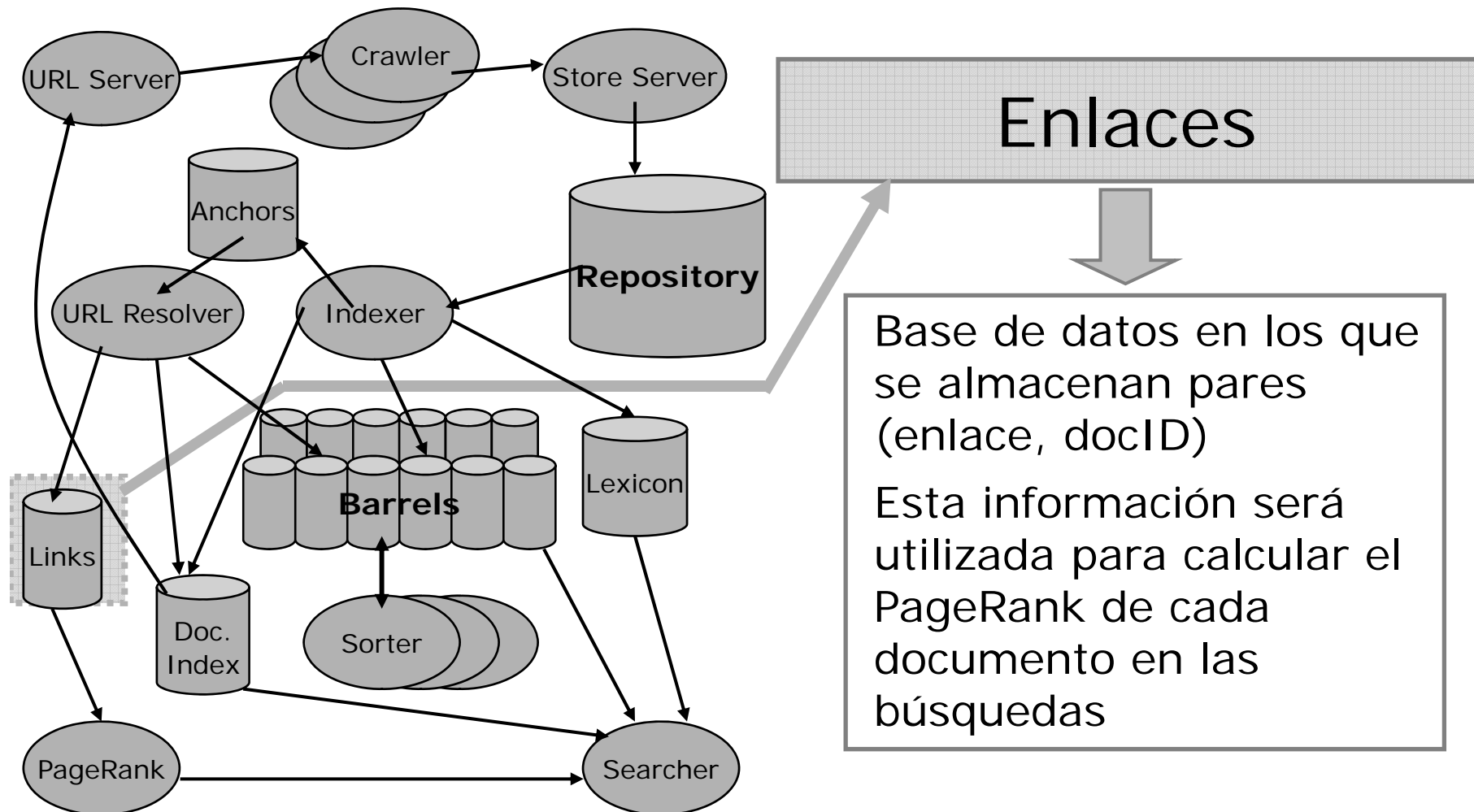


# Arquitectura: Resolvedor de URLs



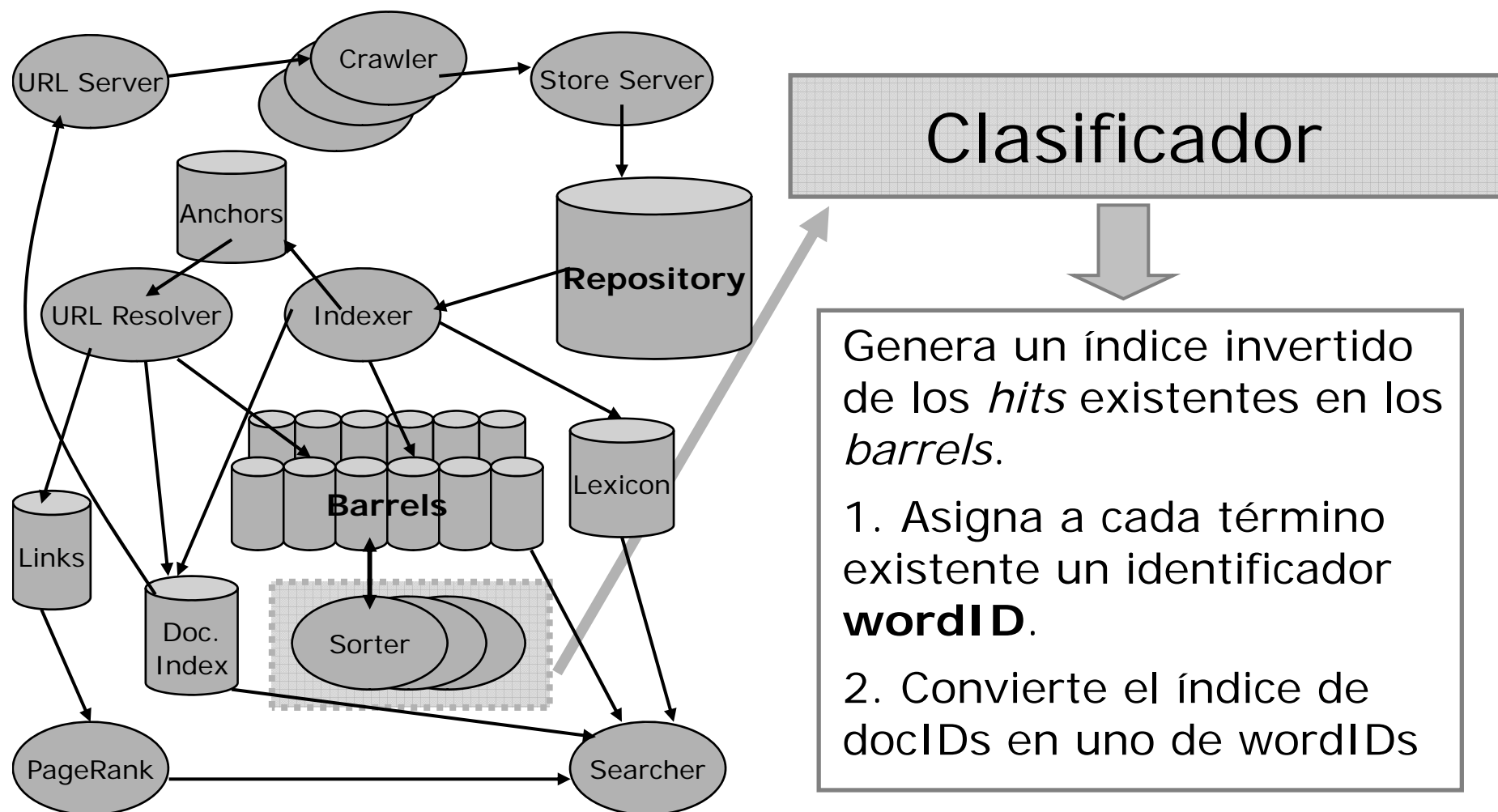


# Arquitectura: Enlaces



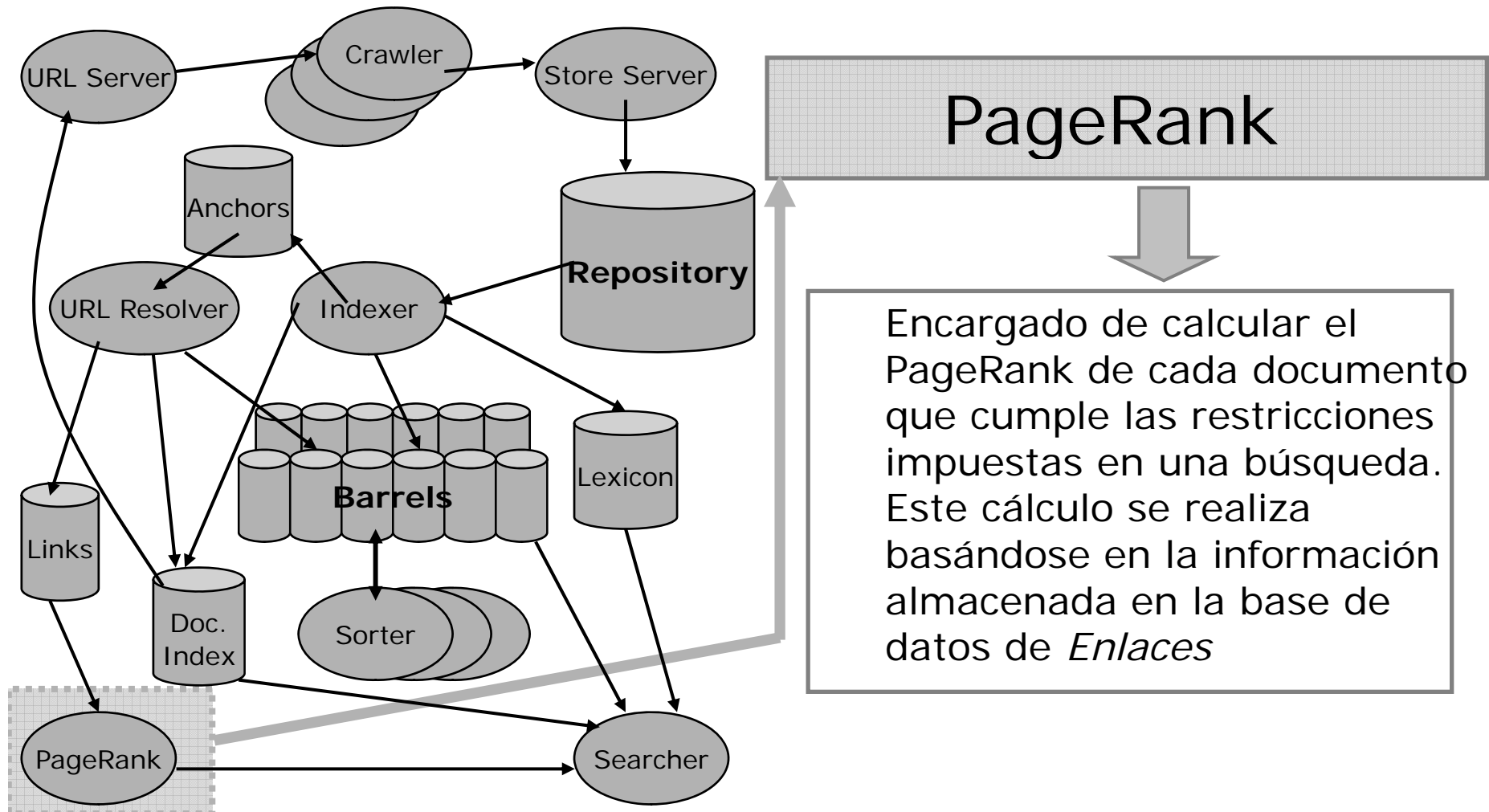


# Arquitectura: Clasificador



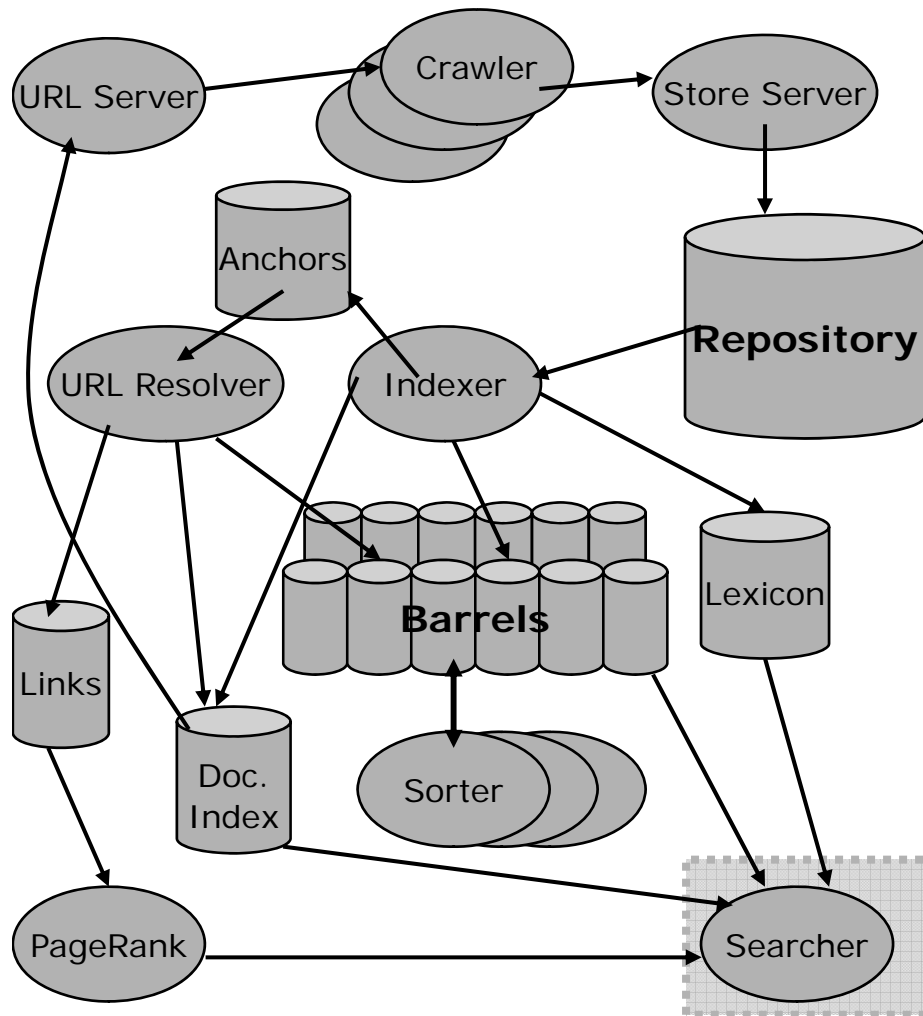


# Arquitectura: PageRank





# Arquitectura: Buscador



**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 *léxico* generado por el *indexador*, el índice invertido generado por el clasificador y la información sobre el PageRank para calcular la relevancia de cada documento



# BigFiles

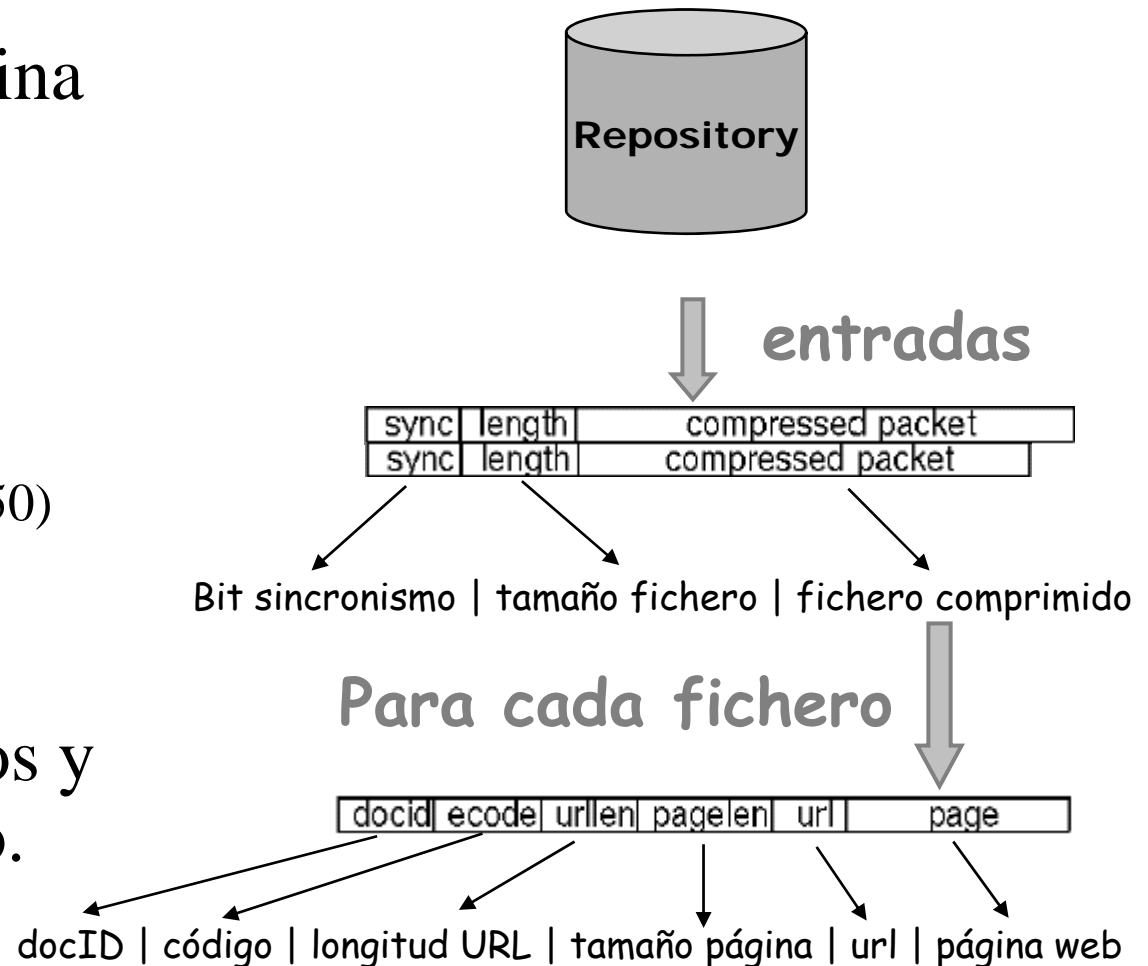
---

- 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 descriptors de los ficheros. (SO no lo hace de manera adecuada)
- Proporciona opciones básicas de compresión.



# Almacén

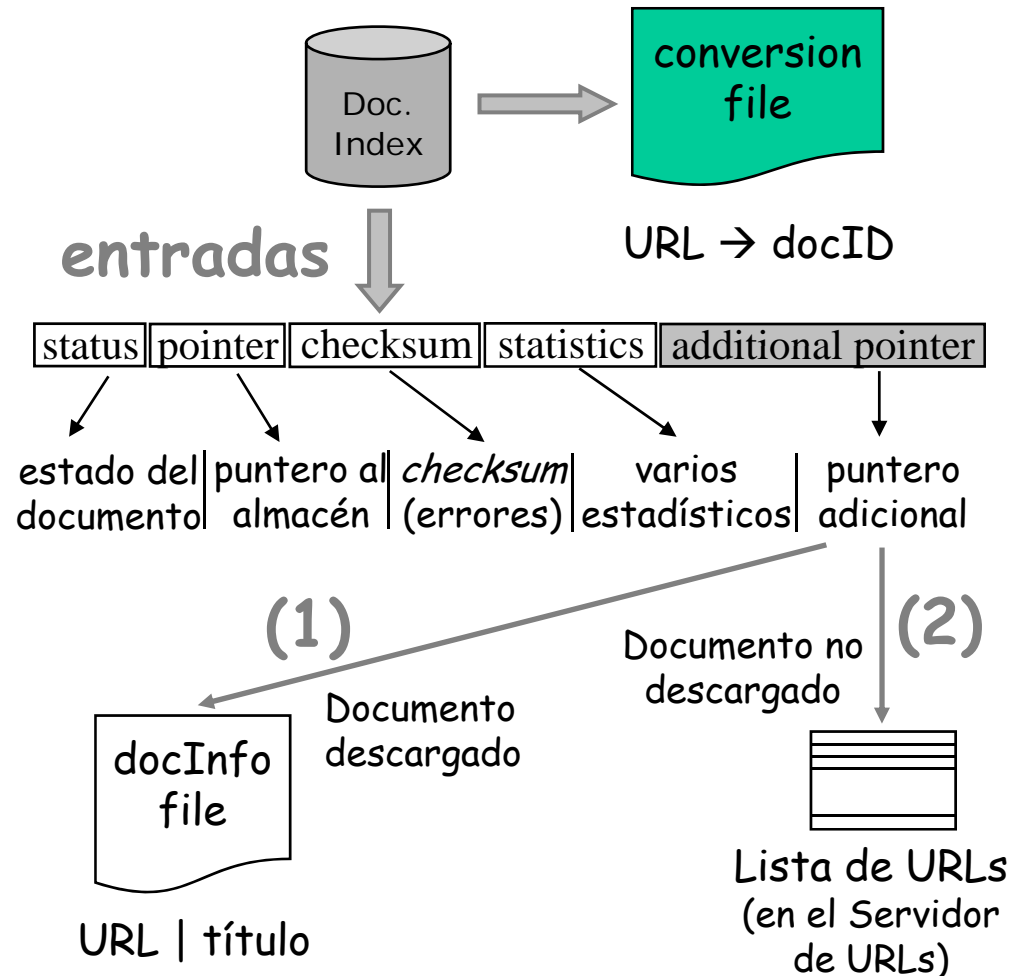
- Contiene el código HTML de cada página descargada
- Se almacena la información comprimida con formato zlib (RFC 1950)
- Estructura de datos simple: ayuda a consistencia de datos y facilita el desarrollo.





# Índice de documentos

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

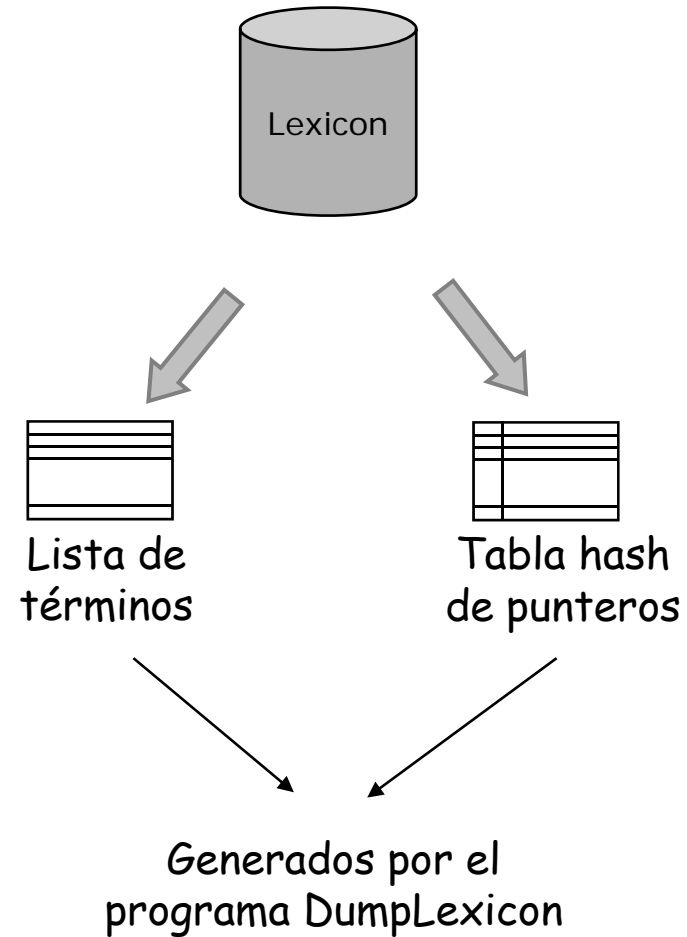




# Diccionario

---

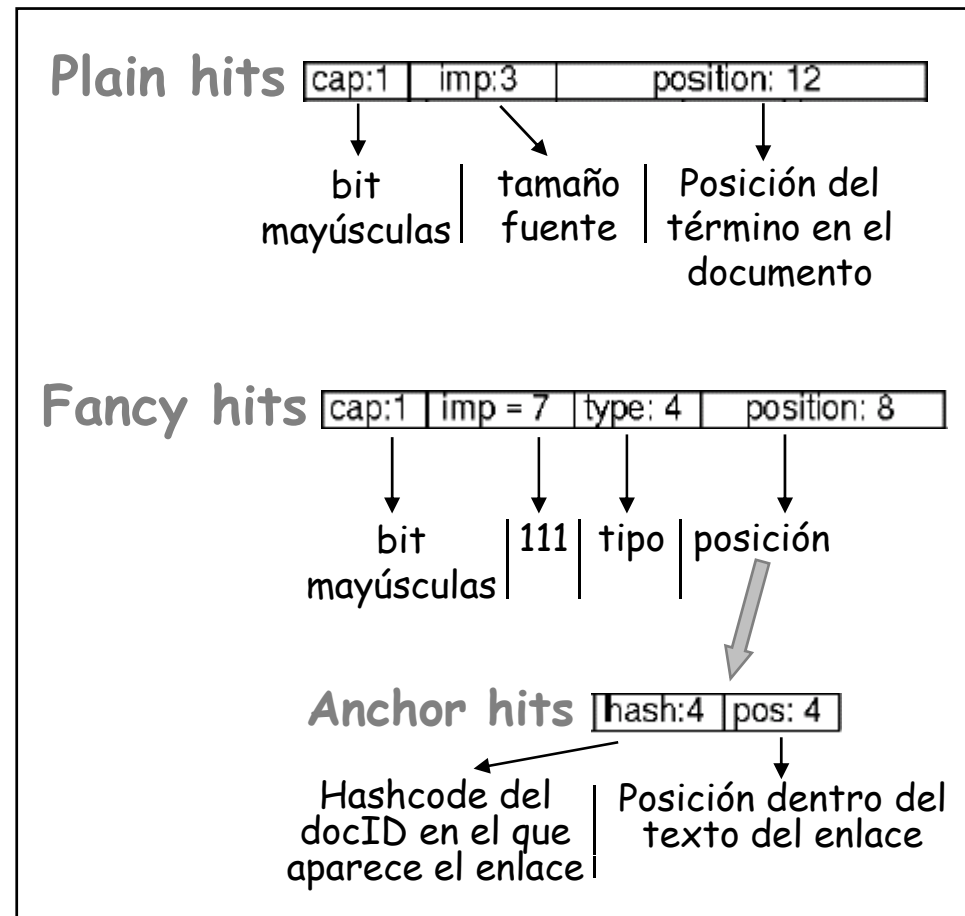
- Base de datos de los términos existentes en los documentos
- Actualmente (1998) contiene 14 millones de entradas





# Hits

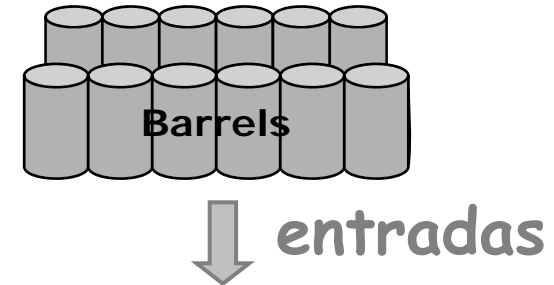
- Cada hit 2 bytes
- 2 tipos:
  - **Fancy hits** (URLs, texto enlaces, etiquetas meta, título)
  - **Plain hits** (resto)





# Índice directo

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



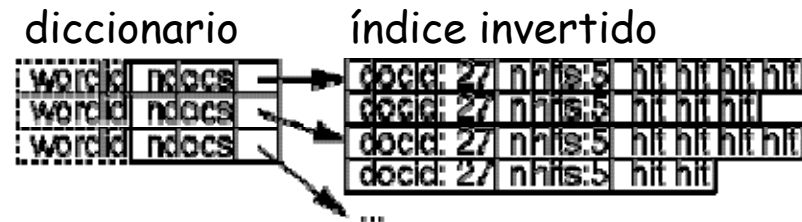
docid	wordid: 24	nhits: 8	hit hit hit hit
	wordid: 24	nhits: 8	hit hit hit hit
	null wordid		
docid	wordid: 24	nhits: 8	hit hit hit hit
	wordid: 24	nhits: 8	hit hit hit hit
	wordid: 24	nhits: 8	hit hit hit hit
	null wordid		

<code>docid</code>	docID del documento
<code>wordid: 24</code>	Lista de wordIDs correspondientes a los términos de los <i>hits</i> de cada entrada
<code>nhits: 8</code>	Longitud total de la lista de hits almacenados a continuación
<code>hit hit hit hit</code>	Lista de hits correspondientes a los wordIDs indicados
<code>null wordid</code>	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



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



# Contenidos

---

- Introducción
- Características de Google
- Arquitectura de Google
- **Exploración de la web: “Crawling”**
- Búsquedas
- Datos estadísticos y de implementación
- Conclusiones



# Conceptos

---

- Crawling: Rastrear servidores Web con el fin de indexar la información que almacenan
- Tarea complicada. Motivos:
  - Es necesario tener en cuenta cuestiones de rendimiento y fiabilidad
  - Supone interactuar con miles de servidores Web y servidores de nombres que están fuera del control del sistema



## Crawling en Google (i)

---

- 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 pags. Web/seg. Utilizando 4 crawlers
- Tasa de datos alrededor de 600k/seg



## Crawling en Google (ii)

---

- 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



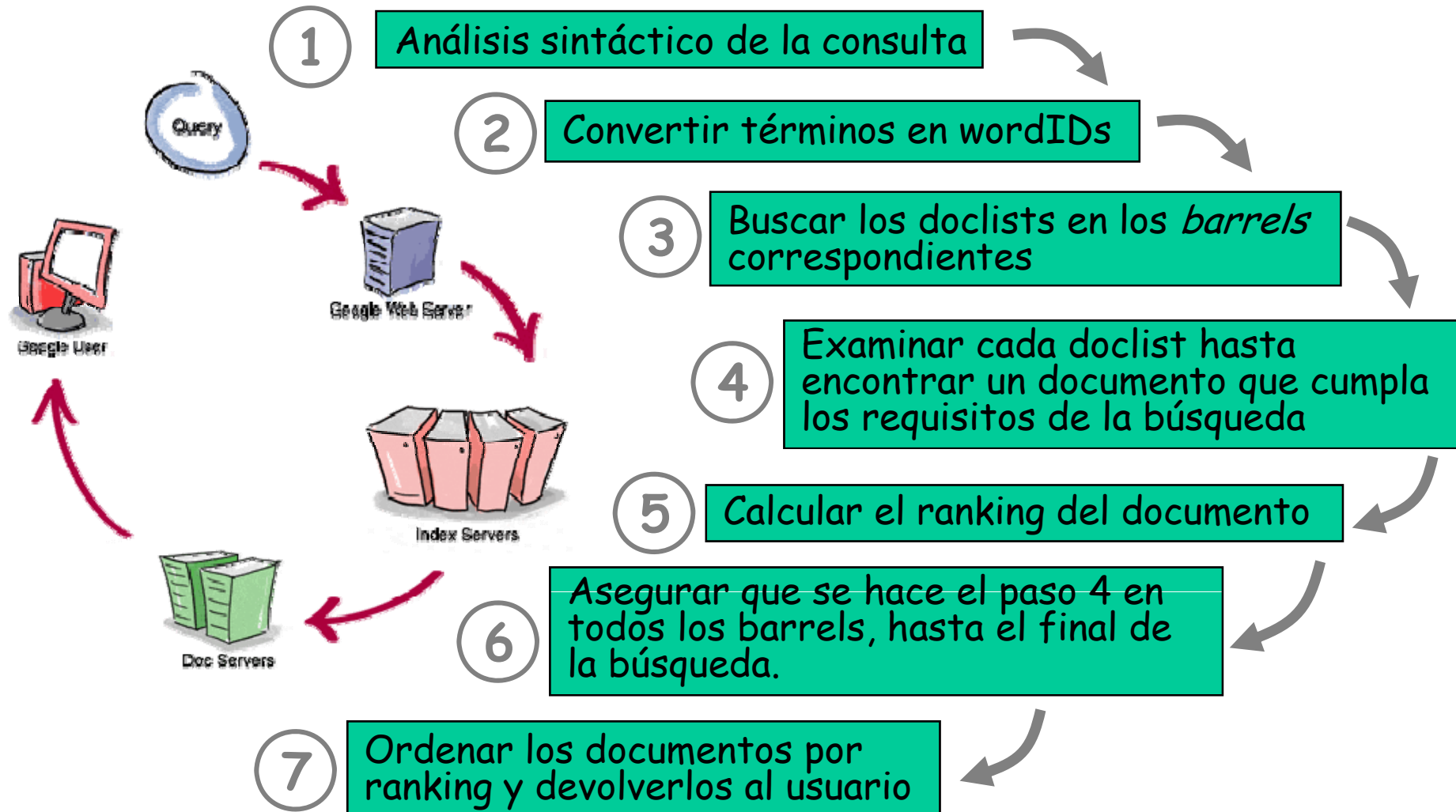
# Contenidos

---

- Introducción
- Características de Google
- Arquitectura de Google
- Exploración de la web: “Crawling”
- **Búsquedas**
- Datos estadísticos y de implementación
- Conclusiones



# Proceso





# Prestaciones

---

- 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, actualmente) se devuelven resultados parciales



# Contenidos

---

- Introducción
- Características de Google
- Arquitectura de Google
- Exploración de la web: “Crawling”
- Búsquedas
- **Datos estadísticos y de implementación**
- Conclusiones



## Datos estadísticos en 1998 (i)

---

<b>Estadísticas de almacenamiento</b>	
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 ( <i>anchors</i> ) temporal	6,6 GB
Índice de documentos	9,7 GB
Base de datos de enlaces	3,9 GB
<b>Tamaño total sin el Almacén</b>	<b>55,2 GB</b>
<b>Tamaño total con el Almacén</b>	<b>108,7 GB</b>



## Datos estadísticos en 1998 (ii)

---

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



## Lenguajes de programación

---

- 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



# Contenidos

---

- Introducción
- Características de Google
- Arquitectura de Google
- Exploración de la web: “Crawling”
- Búsquedas
- Datos estadísticos y de implementación
- **Conclusiones**



# Conclusiones

---

- Google proporciona una arquitectura modular para un motor de búsqueda
- El énfasis del diseño se ha puesto más en mejorar la calidad de los resultados de las búsquedas que en lograr un elevado rendimiento temporal (aunque el tiempo de devolución de resultados está limitado)
- Para mejorar los resultados devueltos se aprovechan datos como el texto de los enlaces, el tamaño la fuente, etc.



## Referencia

---

- S. Brin y L. Page.

The anatomy of a large-scale  
hypertextual Web search engine.

*7th International World Wide Web  
Conference,*

Brisbane, Australia, April 1998.