



# Sistemas Multiagente

## 5º curso: Ingeniería Informática

Pedro Cuesta Morales

Área de Linguaxes e Sistemas Informáticos  
Departamento de Informática  
Escola Superior de Enxeñería Informática  
Universidade de Vigo



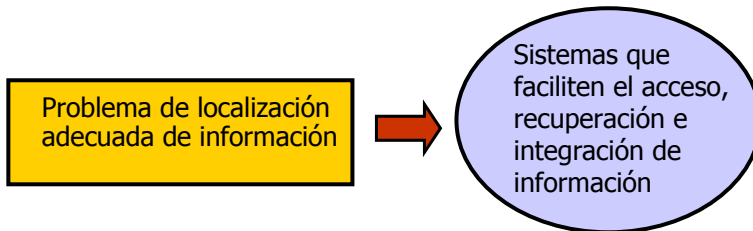
## Tema 4: Agentes de Información/Internet

1. Introducción
2. Recuperación de Información (RI)
  - 2.1. Sistemas de Recuperación de Información (SRI)
  - 2.2. Representación de Documentos de Texto
  - 2.3. Modelos de RI
    - 2.3.1. Modelo Booleano de RI
    - 2.3.2. Modelo Vectorial de RI
  - 2.4. Evaluación del Funcionamiento de un SRI
  - 2.5. Técnicas de Mejora de los SRI
3. Recuperación de Información en la Web
4. Agentes de Información
  - 4.1. No cooperativos
  - 4.2. Cooperativos
  - 4.3. Adaptativos

# 1. Introducción

## ✓ Problemas relativos a la información en Internet:

- Gran volumen (añaden cada día del orden de 7.3 millones de páginas)
- Distribuida entre multitud de computadores
- Heterogénea (tipo, formato, lenguaje, ...)
- Carácter dinámico (las páginas aparecen y desaparecen, o son modificadas con frecuencia)
- Redundancia (contenidos duplicados)
- Calidad (información falsa, incorrecta o errónea),...



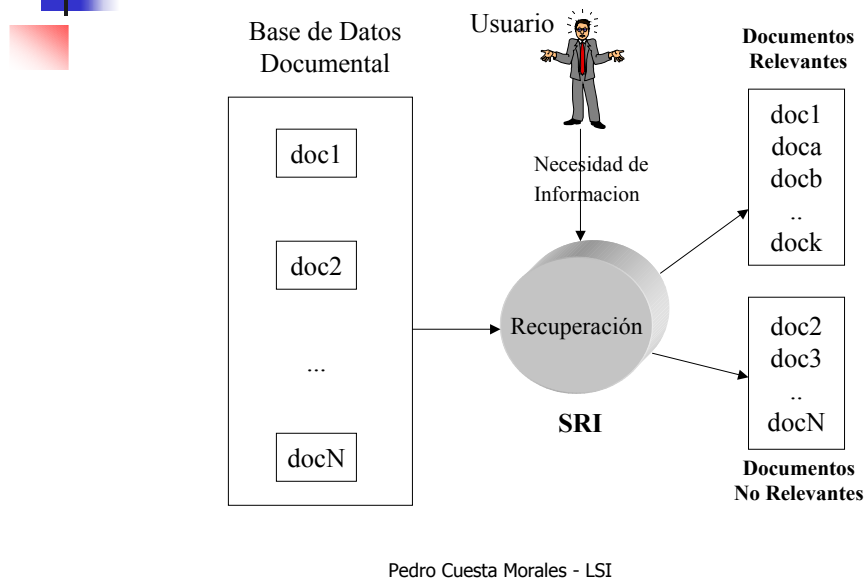
Pedro Cuesta Morales - LSI

# 2. Recuperación de Información

- ✓ La **Recuperación de Información (RI)** es el problema de la selección de información documental desde dispositivos de almacenamiento, en respuesta a consultas realizadas por un usuario
- ✓ Se desea que la información recuperada sea relevante para el usuario y que se obtenga en un intervalo de tiempo adecuado
- ✓ La RI se lleva a cabo mediante los **Sistemas de Recuperación de Información (SRI)**, que se encargan del almacenamiento y organización de un conjunto de documentos para su posterior recuperación por los usuarios
- ✓ Los SRIs manejan **bases de datos documentales**

Pedro Cuesta Morales - LSI

## 2.1. Sistemas de Recuperación de Información

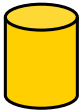


## Tareas Fundamentales en un SRI

- ✓ Cómo representar los documentos en la base documental
- ✓ Cómo representar las necesidades de información de los usuarios en forma de consultas
- ✓ Cómo evaluar la satisfacción de una necesidad de información (consulta) por un documento
- ✓ Cómo presentar los resultados de la consulta al usuario
- ✓ Cómo refinar los resultados de una consulta previa

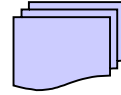
## Diferencias entre un SRI y un SGBD

### SGBD:



- ✓ Información más estructurada (tablas)
- ✓ Recuperación determinística (todo lo recuperado es relevante para el usuario)

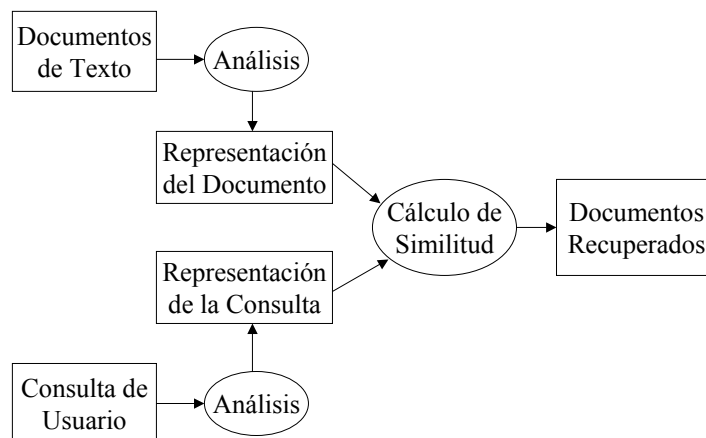
### SRI:



- ✓ Información menos estructurada (documentos)
- ✓ Recuperación probabilística (parte de la información recuperada puede no ser relevante)

Pedro Cuesta Morales - LSI

## Proceso de Recuperación de Información



Pedro Cuesta Morales - LSI



## Elementos de un SRI

### ✓ Base de datos documental:

- Almacena los documentos y una representación de sus contenidos
- La representación suele estar compuesta por términos índice
- El módulo indexador genera automáticamente estas representaciones

### ✓ Sistema de consulta:

- El usuario formula las consultas mediante un lenguaje de consulta (interfaz de consulta)
- Lleva a cabo la interfaz con el usuario mostrando los documentos recuperados (interfaz de respuesta)

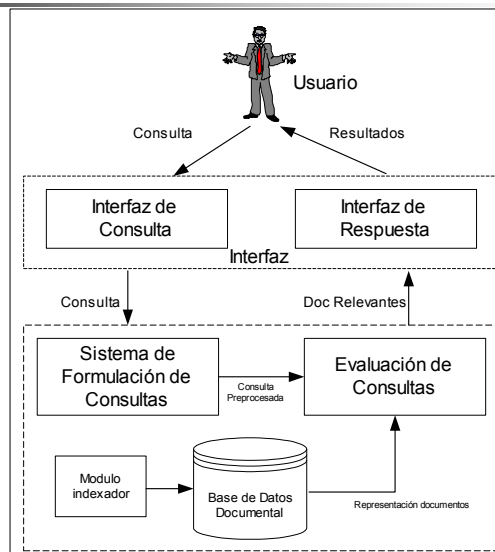
### ✓ Mecanismo de Evaluación:

- Evalúa el grado en que los documentos satisfacen la consulta y recupera los que considera relevantes mediante una técnica de RI

Pedro Cuesta Morales - LSI



## Elementos de un SRI (2)



Pedro Cuesta Morales - LSI



## 2.2. Representación de Documentos de Texto

- ✓ Este tipo de documentos se representan por:
  - una **componente estructurada** en campos: Título, autor, fuente, abstracts, palabras clave, identificador de doc., etc.
  - una **componente no estructurada**, el texto tal cual llega
- ✓ Esta representación se obtiene mediante procedimientos de indexación que asignan un conjunto de índices a cada documento en función del análisis de su contenido
- ✓ Cada índice representa uno o más términos
- ✓ La indexación puede efectuarse manual o automáticamente

Pedro Cuesta Morales - LSI



## Representación de Documentos de Texto (2)

- ✓ La indexación automática se está imponiendo cada vez más por la gran longitud de los textos procesados
- ✓ Está basada en el cálculo de la frecuencia de aparición de los términos en los documentos
- ✓ Los términos con mayor poder de resolución (medida de la validez de un término como índice) son los de las frecuencias intermedias
- ✓ Los de frecuencias muy altas o muy bajas no son significativos para representar el contenido de un documento
- ✓ Usar frases de términos como índices mejora la recuperación, al tener mayor poder de resolución que los términos individuales

Pedro Cuesta Morales - LSI



## Representación de Documentos de Texto (3)

- ✓ Para determinar los términos a usar como índices en la indexación automática, se efectúan varios pasos:
  - Extracción de los términos existentes en la base documental mediante un autómata finito
  - Eliminación de los términos muy comunes en el idioma considerado usando una lista de palabras vacías ("stoplist"). Donde estarán artículos, preposiciones, adverbios, ...
  - Reducción a la raíz (lematización o "stemming") de los términos, eliminando los sufijos (Algoritmo de Porter)
- ✓ Cada documento es un vector de índices ponderados:

$$d = \begin{matrix} t_1 & t_2 & t_3 & t_4 & & t_{n-2} & t_{n-1} & t_n \\ \boxed{w_1} & \boxed{w_2} & \boxed{w_3} & \boxed{w_4} & \dots & \boxed{w_{n-2}} & \boxed{w_{n-1}} & \boxed{w_n} \end{matrix}$$

- ✓ El peso  $w_i$  indica el grado en el que el índice  $t_i$  describe al documento  $d$

Pedro Cuesta Morales - LSI



## Representación de Documentos de Texto (4)

- ✓ Hay varias formas de calcular el peso de un término índice. La más sencilla es la **indexación binaria**: asignar peso  $w_i=1$  si el término aparece en el documento y  $w_i=0$  en otro caso
- ✓ Esta indexación provoca una pérdida de información. El uso de pesos mejora sensiblemente la recuperación
- ✓ Una de las más empleadas es la **frecuencia documental inversa (IDF)** de Salton (1989):

$$w_{ij} = \text{Peso}(t_i \text{ en } d_j) = f_{ij} \cdot \log_2(N/N_i)$$

$f_{ij}$  = frecuencia de aparición de  $t_i$  en  $d_j$

$N$  = número de documentos en la colección

$N_i$  = número de documentos donde aparece  $t_i$

Pedro Cuesta Morales - LSI



## 2.3. Modelos de RI

- ✓ Existen distintos modelos de RI dependiendo de:
  - El **tipo de consulta** considerado (lenguaje de consulta)
  - El **mecanismo de evaluación** de consultas
- ✓ Belkin y Croft proponen una clasificación en función de la filosofía seguida por el mecanismo de evaluación de consultas al emparejar consultas y documentos:
  - Modelos de RI basados en **coincidencia exacta**:  
seleccionan aquellos documentos que se adecuan totalmente a la consulta. El representante de este grupo es el modelo booleano de RI, que no permite ordenar los resultados por relevancia
  - Modelos de RI basados en **coincidencia parcial**:  
Los modelos de coincidencia parcial no exigen una adecuación total y ordenan los resultados por relevancia

Pedro Cuesta Morales - LSI



## Modelos de RI (2)

Podemos distinguir dos grupos, según emparejen documentos individuales o grupos de éstos con la consulta:

- ★ **Individuales**: comparan con documentos individuales:
  - **Modelo vectorial**: basado en funciones de similitud.
  - **Modelo probabilístico**: basado en la regla de Bayes.
  - **Modelos difusos**: operadores difusos (t-normas, t-conormas)
- ★ **En red**: comparan con grupos de documentos conectados:
  - **Basadas en clusters**
  - **Basadas en técnicas de browsing**, usando redes de conexión entre los documentos

Pedro Cuesta Morales - LSI



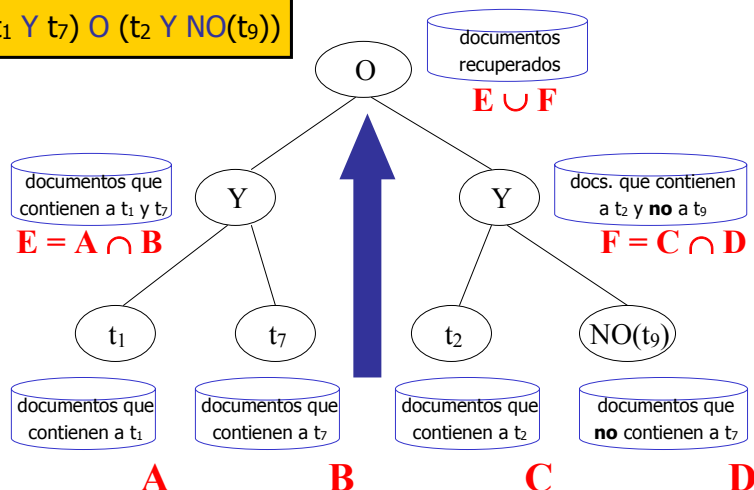
### 2.3.1. Modelo Booleano de RI

- ✓ Los SRI booleanos son los más empleados en las empresas
- ✓ Los documentos se representan mediante vectores binarios
- ✓ Las consultas son combinaciones de términos índice conectados por los operadores lógicos **Y**, **O** y **NO**
- ✓ Los documentos relevantes son los que satisfacen totalmente la consulta, el resto son irrelevantes (no hay grados)
- ✓ La evaluación usa la Teoría de Conjuntos: representa la consulta en un árbol, lo recorre desde las hojas (términos) hasta la raíz y devuelve el conjunto de documentos de ésta

Pedro Cuesta Morales - LSI

### Modelo Booleano de RI (2)

$(t_1 \text{ Y } t_7) \text{ O } (t_2 \text{ Y NO}(t_9))$



Pedro Cuesta Morales - LSI



## Modelo Booleano de RI (3)

El modelo booleano de RI presenta varios inconvenientes:

- No se puede expresar el grado de relevancia de los términos índice con los documentos ([módulo indexador](#))
- El usuario no puede especificar la importancia relativa de los términos en la consulta ([lenguaje de consulta](#))
- No existen grados de relevancia parcial de los documentos ([mecanismo de evaluación](#))

Los modelos de RI restantes solventan estos inconvenientes en mayor o menor medida

Pedro Cuesta Morales - LSI



## 2.3.2. Modelo Vectorial de RI

- ✓ Los documentos son vectores de números reales en  $[0,1]$  ([vectores en un espacio n-dimensional](#))

$$d_j = \{wd_{1j}, wd_{2j}, \dots, wd_{nj}\}$$

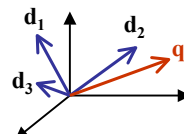
$wd_{ij}$  = peso del término  $i$  en el documento  $j$

$n$  (términos) = dimensión del espacio vectorial

- ✓ Las consultas tienen la misma representación que los documentos (los términos relevantes en la consulta toman un peso distinto de 0 y el resto, peso 0)

$$q_k = \{wq_{1k}, wq_{2k}, \dots, wq_{nk}\}$$

- ✓ La evaluación empareja cada documento con la consulta determinando el grado en que dicho documento la satisface mediante una [medida de similitud](#)



Pedro Cuesta Morales - LSI



## Modelo Vectorial de RI (2)

✓ Como ejemplos de medidas de similitud tenemos:

- **Producto escalar:**  $SIM(d, q) = \sum_{i=1}^n t_i q_i$

- **Coseno:**  $SIM(d, q) = \frac{\sum_{i=1}^n t_i q_i}{\sqrt{\sum_{i=1}^n (t_i^2 q_i^2)}}$

- **Distancia:**  $SIM(d, q) = -\sqrt{\sum_i |t_i - q_i|^2}$

Pedro Cuesta Morales - LSI



## Modelo Vectorial de RI (3)

- ✓ Al existir grados de relevancia, los resultados obtenidos se puedan ordenar en función de éstos  
→ documentos ordenados por relevancia
- ✓ El tamaño de la salida se puede controlar por el usuario:
  - poniendo un valor límite al número de documentos recuperados, o
  - devolviendo únicamente aquellos documentos que superan un umbral de relevancia fijado por el usuario
- ✓ Un ejemplo de SRI vectorial es SMART de Salton con medidas angulares como el coseno

Pedro Cuesta Morales - LSI



## 2.4. Evaluación del Funcionamiento de un SRI

Podemos evaluar un SRI en función de dos características:

✓ **Eficiencia:**

- *Tiempo de respuesta*
- *Espacio de almacenamiento necesario*

✓ **Eficacia:** grado en el que el SRI trabaja correctamente:

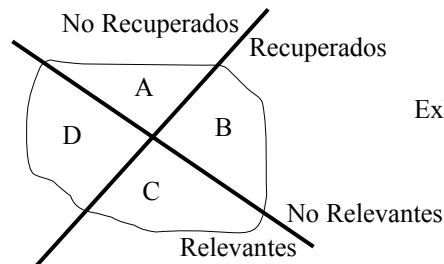
- Precisión: en qué medida los documentos recuperados por el SRI son relevantes para el usuario
- Exhaustividad ("recall"): en qué medida el SRI recupera todos los documentos relevantes existentes en la base de datos. Imposible de calcular en sistemas abiertos como la WEB.

La eficacia es la más empleada

Pedro Cuesta Morales - LSI



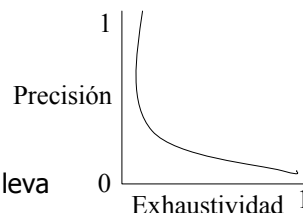
## Evaluación del Funcionamiento de un SRI (2)



$$\text{Exhaustividad} = \frac{C}{D + C}$$

$$\text{Precisión} = \frac{C}{B + C}$$

Alcanzar valores altos de exhaustividad conlleva valores bajos de precisión



Pedro Cuesta Morales - LSI



## 2.5. Técnicas de Mejora de los SRI

- ✓ Información que el SRI puede dar al usuario para orientarlo de cara a facilitar el análisis de la respuesta:
  - Cantidad de documentos recuperados: permite refinar la consulta con términos más específicos o más genéricos
  - Uso del tesauro de palabras relacionadas, sinónimas, genéricas o específicas
  - Información sobre cómo formular una consulta

Pedro Cuesta Morales - LSI



## Retroalimentación de Relevancia

- ✓ **Retroalimentación de relevancia** ("relevance feedback"): Refinamiento de la consulta por parte el usuario en función de la salida proporcionada por el SRI

Por ejemplo, en los SRI vectoriales se pueden distinguir dos variantes:

- **Redefinición de los pesos** de los términos de la consulta original, aumentando los de aquellos términos presentes en los documentos relevantes recuperados y reduciendo los de los irrelevantes, *sin añadir o eliminar ningún término*
- **Expansión de la consulta**, basada en el cambio de pesos y la adición/eliminación de términos presentes en los documentos relevantes e irrelevantes recuperados, respectivamente

Pedro Cuesta Morales - LSI



## Retroalimentación de Relevancia (2)

- ✓ La técnica de expansión de consultas que mejores resultados ha obtenido en los SRI vectoriales es la **Ide dec-hi**:

$$d' = d + \sum_{i=1}^{n_r} R_i - S_i$$

donde:

- $d$  es el vector de la consulta original
- $R_i$  es el vector del  $i$ -ésimo documento relevante recuperado
- $S_i$  es el vector del  $i$ -ésimo documento no relevante recuperado
- $n_r$  es el número de documentos relevantes recuperados

Pedro Cuesta Morales - LSI



## Retroalimentación de Relevancia (3)

- ✓ La técnica **Ide dec-hi** se basa en mezclar el vector de la consulta original con los de los documentos recuperados
- ✓ Se redefinen automáticamente los pesos de los términos de la consulta sumando los pesos de los términos en los documentos relevantes y restando los de los irrelevantes
- ✓ La adición de términos se efectúa incluyendo en la consulta original todos los términos procedentes de los documentos relevantes recuperados que no existieran anteriormente
- ✓ **Proactividad de un Agente de Información** → en base a la confirmación de los documentos relevantes recuperados se refina el proceso de búsqueda

Pedro Cuesta Morales - LSI



### 3. Recuperación de Información en la Web

- ✓ Facilitar la Recuperación de Información sobre la web tarea fundamental dada la trascendencia que ha tomado la web:
  - Buscadores utilizan algoritmos cada vez más potentes
  - Buscadores especializados
  - Aplicación de técnicas avanzadas, ...
- ✓ La web está formada por un conjunto dinámico de documentos, por lo que las fórmulas del espacio vectorial deben ser aproximadas
- ✓ El modelo del espacio vectorial supone que los documentos son independientes entre sí, pero la web esta formada por documentos (páginas) entrelazados, donde esta información (enlaces) puede ser utilizada, además de la frecuencia de términos
- ✓ Modelos que ofrecen mejores resultados → En red (comparan con grupos de documentos conectados)

Pedro Cuesta Morales - LSI



### Herramientas de búsqueda

- ✓ Buscadores genéricos:
  - Directorios
  - Motores de búsqueda(actualmente los buscadores combinan ambas estrategias)
- ✓ Buscadores especializados (turismo, salud, artículos, ...)
- ✓ Buscadores inteligentes
- ✓ Multibuscadores/Metabuscadores: buscan en varios buscadores
- ✓ Agentes inteligentes



Pedro Cuesta Morales - LSI



## Directorios

- ✓ Taxonomías jerárquicas que intentan clasificar los distintos temas o áreas de conocimiento (arte y cultura, ciencia y tecnología, ciencias sociales, deporte y ocio, ...)
- ✓ Ejemplo más significativo: [Yahoo!](#)
- ✓ Algunos contienen más de ***100 mil*** categorías jerarquizadas y ***millones*** de sitios web clasificados
- ✓ Ventaja: "si encontramos algo seguramente será útil"
- ✓ Inconvenientes: la clasificación muchas veces no es suficientemente especializada y no todo lo que existe en la web está clasificado → dificultad para clasificar
- ✓ Necesidad de sistemas que clasifiquen automáticamente

Pedro Cuesta Morales - LSI



## Motores de búsqueda

- ✓ Componentes de un buscador:
  - [Robot](#): realiza peticiones a sitios remotos para localizar nuevas páginas y/o cambios en las páginas. Le pasa las páginas al indexador.
  - [Indexador](#): analizar la página y actualizar el índice. Utilizan técnicas de Recuperación de Información (procesan el texto, eliminan palabras vacías, extraen raíces, ...).
  - [Motor de Búsqueda](#): es el encargado analizar la consulta del usuario y utilizando el índice buscar y ordenar por relevancia las páginas que satisfacen mejor la consulta.
  - [Interfaz de Usuario](#): capturar consulta (normal y avanzada) y mostrar resultados (de x en x).

Pedro Cuesta Morales - LSI





## Robots de búsqueda

### ✓ Estrategias para decidir qué páginas visitar:

- Partir de páginas con muchos enlaces, y/o de los sitios más populares de la web, y/o admitir solicitudes para visitar el sitio (cola de peticiones)
- Utilizar los enlaces para hacer un recorrido en anchura, en profundidad, o combinadas con medidas de popularidad (visitar las de mayor calidad, p.e. páginas que apuntan a ella)
- Visitar periódicamente páginas. Los buscadores pueden contener enlaces inválidos. Algunos buscadores aprenden la frecuencia con la que cambian las páginas.
- Mediante directivas los administradores de sitio pueden controlar el comportamiento de los robots. P.e. impedir indexar algo
- Problemas: frames, mapas de imagen, páginas dinámicas, ...

### ✓ Estrategias para decidir qué indexar → cada robot utiliza un algoritmo particular. Se tiene en cuenta:

- Título de la página o primeros párrafos
- Contenido completo de la página
- Metaetiquetas
- Imágenes, o textos alternativos de las imágenes, ...

Pedro Cuesta Morales - LSI



## Indexador

### ✓ Obtener una representación interna (índice) de las páginas que les proporciona el robot →

**Análisis de Contenido:** lista de parada (Si: Altavista, Google, ...; No: Fast) extracción de raíces (Si: Lycos; No: Google)

### ✓ Índices se organizan en archivos invertidos: lista de palabras (vocabulario) y páginas en las que aparecen dichas palabras.

### ✓ Además se puede almacenar (mayor requisitos de espacio):

- Posición de la palabra en la página
- Información del uso de mayúsculas o tipo de letra utilizado
- Fecha de creación
- Texto asociado a los enlaces, ...

Pedro Cuesta Morales - LSI



## Motor de Búsqueda

- ✓ Encontrar en el índice las páginas relacionadas con la consulta y ordenarlas por relevancia
- ✓ Utilizando criterios de:
  - Localización: mayor relevancia cuando las palabras aparecen en el título, o al comienzo
  - Frecuencia: número de veces que aparecen las palabras de la consulta
  - Popularidad: una página es mejor cuantos más enlaces apuntan a ella

Problema: se podrían crear páginas que apuntan a una determinada para incrementar su popularidad

Solución: PageRank (Google)



Calcula el PageRank de cada página en función del número de enlaces que apuntan a ella, pero ponderando en función de la calidad de esas páginas que apuntan a la original y del número de enlaces que salen de dichas páginas

Pedro Cuesta Morales - LSI



## Interfaz de usuario

- ✓ Interfaz de consulta
  - Caja de texto para introducir consulta → secuencia de palabras. Significado diferente en función del buscador:
    - Páginas que contengan al menos una de las palabras
    - Páginas que incluyan todas las palabras
  - Tratamiento de mayúsculas/minúsculas, extracción de raíces si/no , ... → Obtener información del funcionamiento del buscador
- ✓ Interfaz de respuesta
  - X páginas más relevantes (título, URL, tamaño, resumen, ...), total de páginas, ...

Pedro Cuesta Morales - LSI



## Interfaz de usuario: búsquedas avanzadas

✓ Los buscadores actuales no se limitan sólo a buscar texto, sino que incorporan funciones avanzadas:

- Idioma: buscar en páginas escritas en un idioma determinado
- Formato de archivo: restringir los resultados a un determinado formato de archivo (html, pdf, ...)
- Fecha: buscar en páginas actualizadas en un periodo de tiempo determinado
- Presencia: páginas en donde los términos aparezcan en: título, contenido, url, ...
- Dominios: restringir la búsqueda a un dominio: com, es, ...
- Enlaces: páginas que contengan algún enlace a uno determinado
- Imágenes: páginas que contengan una imagen, ...

Pedro Cuesta Morales - LSI



## Buscadores especializados

✓ Estrategias y heurísticas especializadas y dependientes del un dominio de aplicación particular

- Tutoriales/cursos: <http://www.abcdatos.com/>
- Noticias:
  - <http://noticias.biwe.es/>
  - <http://www.titulares.com/>
  - <http://www.acceso.com>
- Precios de productos: <http://www.buscaproductos.com/>
- Software: <http://www.softonic.com/>
- Viajes: <http://kasbah.com/>
- ...

Pedro Cuesta Morales - LSI

## Buscadores inteligentes



### ✓ Consultas en lenguaje natural

- <http://www.ask.com>
- <http://www.pregunta.com>

### ✓ Una pregunta:

- Opción más parecida a la pregunta
- Sitios que les han parecido interesantes a personas con preguntas similares

Pedro Cuesta Morales - LSI

## Multibuscadores/Metabuscadores

- ✓ Los metabuscadores se encargan de recoger e integrar los resultados obtenidos por diferentes motores de búsqueda, presentándoselos al usuario de forma uniforme
- ✓ Cada buscador particular sólo indexa una parte de la web
- ✓ Cada buscador utiliza algoritmos propios de indexación
- ✓ Hipótesis:
  1. una búsqueda ofrecerá mejores resultados si se conoce la respuesta de varios buscadores
  2. Además: se traduce la consulta del usuario al lenguaje de consulta de cada buscador particular
- ✓ Importante → como se realiza la integración de resultados
- ✓ Ejemplos:
  - Ixquick: <http://www.ixquick.com/>
  - Todalenet: <http://es.todalanet.com/>
  - Vivisimo: <http://vivisimo.com/>
  - Biwe: <http://multibuscador.biwe.com/>

Pedro Cuesta Morales - LSI



## 4. Agentes de Información

- ✓ Los agentes de información son agentes software que tienen acceso a múltiples fuentes de información heterogéneas geográficamente distribuidas. Intentan resolver los problemas asociados al manejo de información en Internet.
- ✓ Los agentes pueden asistir al usuario en la búsqueda y filtrado de información relevante, informar cuando nuevos datos de interés estén disponibles, negociar la compra o venta de productos, participar en subastas electrónicas, etc.
- ✓ Los agentes de información ayudan al usuario en la ejecución de tareas. Para llevar a cabo este objetivo tienen que ser capaces de capturar y almacenar las preferencias del usuario.
- ✓ Además, deberían de ser capaces de actuar adecuadamente ante nuevas situaciones no previstas, es decir, deberían tener capacidad de aprendizaje (esta es una de las características más difíciles de conseguir).
- ✓ Los agentes pueden llevar a cabo sus tareas de manera independiente o trabajar de manera coordinada con otros agentes.

Pedro Cuesta Morales - LSI



## Clasificación de AI (Klush, 1999)

- ✓ En función de la habilidad para cooperar con otros en la ejecución de tareas los agentes se pueden clasificar en Agentes no cooperativos (agentes individuales) y Agentes cooperativos (sistemas multiagente)
- ✓ Agentes racionales: agentes que se utilizan en el comercio electrónico y que median por su usuario en compras o subastas. P.e. agentes que buscan el mejor precio de un producto ("shopbots"), agentes que participan automáticamente en mercados y subastas, ...
- ✓ Agentes adaptativos: son capaces de adaptarse por sí mismos a cambios en su entorno. Dos aspectos importantes:
  - En Internet los agentes tienen que construirse para tratar con información incierta e incompleta, de una manera fiable y segura
  - Personalización, interesa sistemas que se adapten a cada usuario y aprendan de su comportamiento
- ✓ Agentes de información móviles: capaces de viajar autónomamente a través de Internet de un sitio a otro para la ejecución de sus tareas en diferentes servidores (p.e. obtener información)

Pedro Cuesta Morales - LSI



## 4.1. AI no cooperativos

### Clasificación en función de tareas (Cuesta et al., 2002)

- ✓ Búsqueda: ayudan al usuario en la tarea de recuperación de información sobre la web.
  - *Bullseye*: búsqueda sobre más de 800 buscadores. Permite almacenar búsquedas, añadir comentarios a las páginas, ...
  - *CiteSeer*: especializado en documentos científicos (publicaciones de investigación). Citas a cada artículo, documentos similares, bibliografía, usuarios que han visto el artículo también han visto,...
- ✓ Monitorización: vigilar cambios en una página indicada por el usuario, aparición de páginas en buscadores, etc. Se comunica con el usuario a través de e-mail.
  - *TracerLock*
- ✓ Filtrado: seleccionar información en función de las preferencias del usuario.
  - *BotBox*
- ✓ Navegación: agentes de interfaz que ayudan al usuario a navegar por la Web.
  - *Leticia*
- ✓ Comercio Electrónico: recomendar productos, comparar precios, ...
  - *MySimon*

Pedro Cuesta Morales - LSI



## 4.2. Agentes cooperativos

- ✓ Para resolver problemas complejos los agentes deben cooperar con otros agentes
  - Sistema Multiagente: conjunto de agentes que se comunican y cooperan en la ejecución de tareas.
  - Comunicación: lenguaje de comunicación de agentes (ACL)
- ✓ Ventajas: simplicidad, flexibilidad, robustez, escalabilidad, integración, ...
- ✓ Dos aspectos clave:
  - Mecanismo de intermediación entre los agentes
  - Forma de resolver las heterogeneidades de información: ontologías
- ✓ Ejemplos: RETSINA, InfoSleuth, Impact, SIMS, MASIR, ...

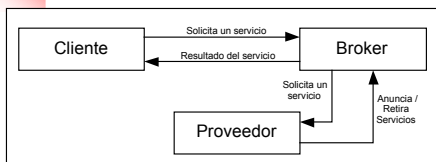
Pedro Cuesta Morales - LSI

## Clasificación de los agentes en un sistema de agentes cooperativos

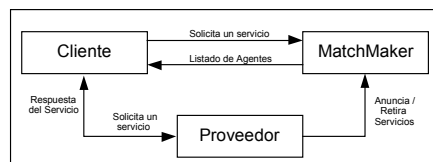
- ✓ Agentes proveedores ("provider"), constituyen la base de la cadena de "consumo" de información y servicios. Son agentes productores, que proporcionan capacidades, como por ejemplo servicios de búsqueda de información o venta de productos, a sus usuarios y a otros agentes.
- ✓ Agentes solicitantes ("requester"), consumen información y servicios ofrecidos por agentes proveedores en el sistema. Serían equivalentes a los consumidores en el mundo real.
- ✓ Agentes intermediarios ("middle"), cuya misión es mediar para que pueda tener lugar una correcta comunicación entre solicitantes y proveedores. Para que pueda llevarse a cabo una correcta mediación, los proveedores tienen que registrar sus capacidades ante uno, o varios agentes mediadores. Los solicitantes o consumidores pueden:
  - Solicitar a un agente intermediario quién de los posibles proveedores puede llevar a cabo un determinado servicio, o
  - La intermediación del mediador para la realización del servicio

Pedro Cuesta Morales - LSI

## Modelos de intermediación



- ✓ No hay comunicación directa entre el proveedor y el solicitante
- ✓ Broker: contactar con el proveedor, negociar, controlar la transacción y devolver los resultados al solicitante



- ✓ El resultado es una lista de proveedores que pueden proporcionar el servicio.
- ✓ Es el propio solicitante el encargado de contactar, negociar con el proveedor del servicio

Pedro Cuesta Morales - LSI

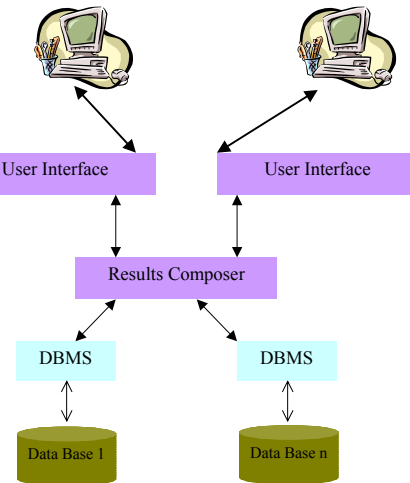
# MASIR

## ✓ Problema de partida:

La construcción de un sistema escalable que permita el acceso uniforme a través de Internet a un conjunto de bases de datos documentales independientes y heterogéneas

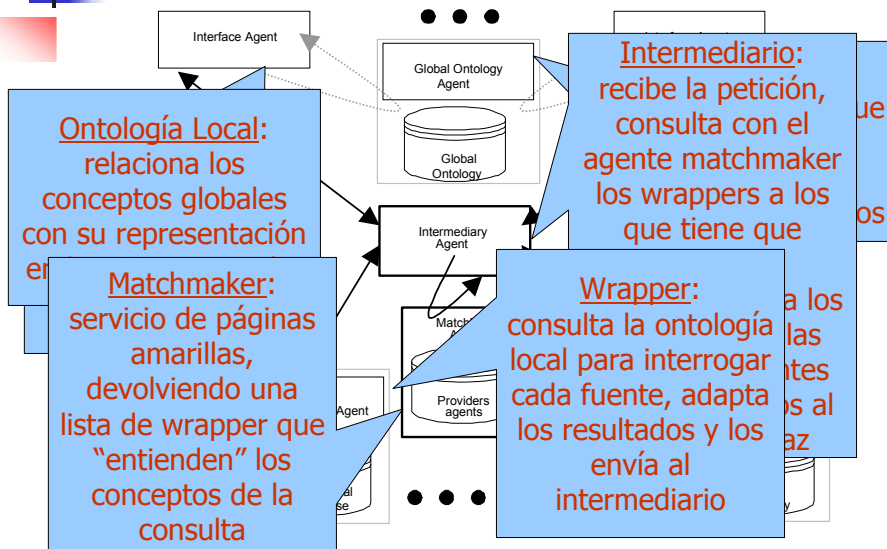
## MASIR

"Sistema multiagente para Recuperación de Información"



Pedro Cuesta Morales - LSI

# MASIR: Arquitectura



Pedro Cuesta Morales - LSI





## MASIR: Intermediación

### ✓ Intermediario (broker):

- Descomponer consulta, trasladarla a las fuentes (wrapper) e integrar resultados

### ✓ Matchmaker:

- Saber que se puede consultar en cada fuente de datos
- Ante una petición (concepto particular) devuelve las fuentes (wrapper) en las que ese concepto tiene sentido

Pedro Cuesta Morales - LSI



## MASIR: ontologías

### ✓ Agente ontología global:

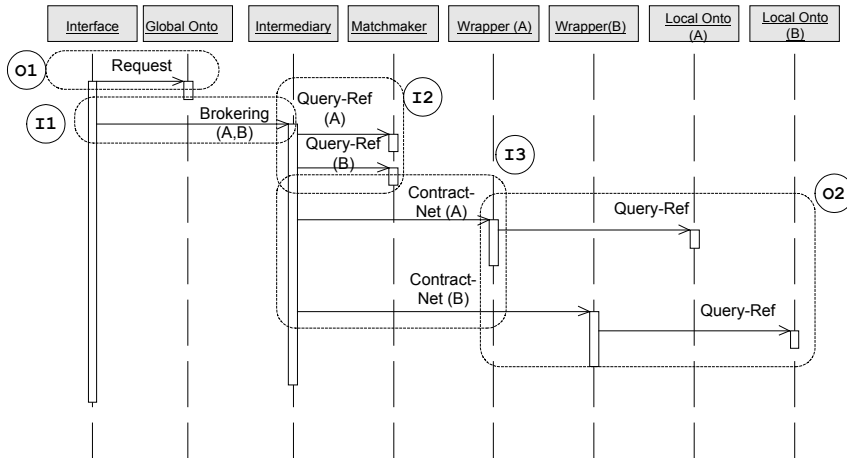
- Representación de alto nivel de los modelos de datos de las BBDD documentales

### ✓ Agente ontología local:

- Relaciona cada concepto definido en la ontología global a la representación particular en cada BBDD específica

Pedro Cuesta Morales - LSI

# MASIR: Protocolos interacción



Pedro Cuesta Morales - LSI

## 4.3. Agentes adaptativos

✓ Selección de motores de búsqueda: [SavvySearch](http://savvy.search.com/) (<http://savvy.search.com/>)

- Aprende a seleccionar los buscadores más apropiados para cada tipo de consulta en cada momento.
- Supervisa al usuario comprobando que enlaces visita. Si un usuario para una consulta utiliza un cierto buscador la "confianza" en ese motor para ese tema aumenta. Si un buscador no devuelve ningún resultado para una consulta, la confianza disminuye.
- Además tiene en cuenta otros aspectos como la rapidez en ofrecer resultados o la disponibilidad.

Pedro Cuesta Morales - LSI



# Bibliografía

- ✓ Klusch, M. Intelligent Information Agents -Agent-Based Information Discovery and Management on the Internet. Springer, 1999.
- ✓ Baeza-Yates, R. and Ribeiro-Neto, B. Modern Information Retrieval. Addison-Wesley, 1999.
- ✓ Salton, G., Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Reading, MA: Addison-Wesley, 1989.
- ✓ P. Cuesta, A. Gómez, M.E. González, R. Laza, F.J. Rodríguez. Agentes de Información. Departamento de Informática IT1/2001 (Universidade de Vigo)

---

\* Gracias a Oscar Cordón García, Departamento de Ciencias de la Computación e Inteligencia Artificial (Universidad de Granada), por sus transparencias de "Introducción a los Sistemas de Recuperación de Información".