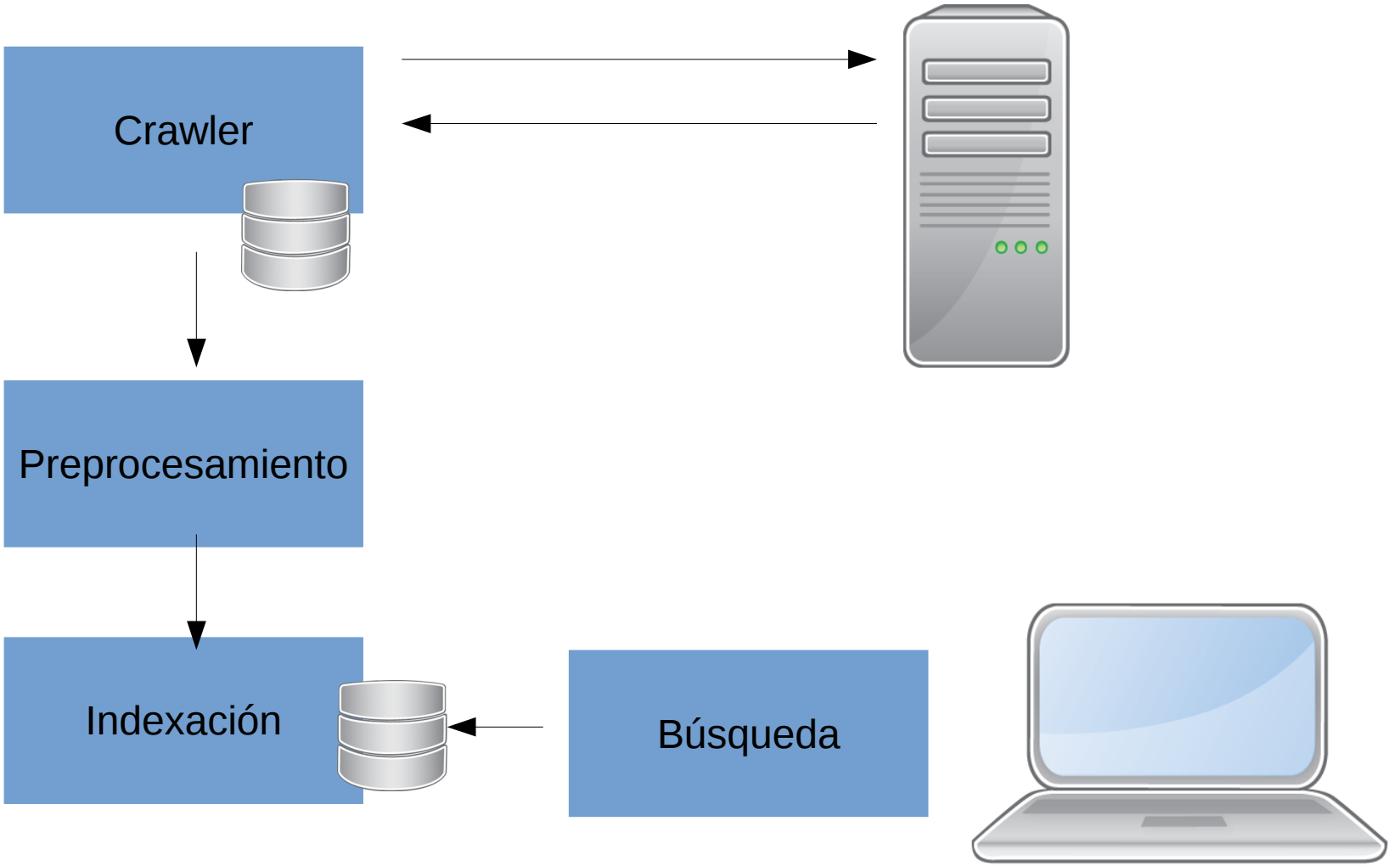
Análisis y Recuperación de Información
Cómo hacer un pequeño Google

ISISTAN
Instituto Superior de Ingenieria de Software Tandil

Arquitectura General



Bajarse Internet

- Es complejo en cantidad de recursos requeridos
  - La cantidad de links explota exponencialmente
  - Se agregan webs de forma constante
- Lentitud de ciertas webs
- Webs maliciosas
- Robots.txt
- Parsear HTML es difícil (el formato no es estricto)

Crawler

- Software que obtiene datos de la web
- Ejemplos: Scrapy, Storm Crawler, Nutch.
- El objetivo es armar una cola de URLs que tenemos que visitar, y generar múltiples threads que descarguen el texto de las webs.
- Cada HTML descargado se parsea y se obtienen los links a otras webs, que son agregados a la cola de descargas.

Ejemplo: Nutch 1.11

- Descargar nutch: nutch.apach
- La versión 2.0 guarda por defecto los datos en HBASE a través de una interfaz de almacenamiento llamada Gora (necesita un poco más de config.)
- La versión 1.11 almacena los datos de forma local usando el formato de hdfs (más fácil de configurar).

Ejemplo: Nutch 1.11

- Archivos a tocar: nutch-site.xml

```
<configuration>
<property>
      <name>http.agent.name</name>
      <value>Analisis y Recuperacion de
Informacion</value>
   </property>
   <property>

<name>fetcher.threads.per.queue</name>
      <value>50</value>
      <description></description>
   </property>
</configuration>
```

Ejemplo: Nutch 1.11

- Seeds (por ej. urls/*.txt):
  - Un conjunto de archivos de texto de donde obtener las urls originales.
  - En el ejemplo arrancamos con 1 solo archivo seed.txt que contiene **https://es.wikipedia.org/wiki/Búsqueda_y_recuperación_de_información**
- regex-urlfilter.txt (modificado para quedarnos dentro de wikipedia):
  - +^https://es.wikipedia.org/([a-z0-9]*\.)*
- En ***windows*** hay que descargar binarios de hadoop y configurar hadoop home en bin/nutch.

Crawlear

- ./bin/crawl [directorio de seeds] [carpeta destino] [cantidad de pasadas]
- Ejemplo: ./bin/crawl urls crawl 3

Preprocesamiento

- Generalmente se trata de un pipeline de modificaciones/anotaciones sobre el contenido original. Siempre se referencia al contenido original.
- Ejemplo clásico:
  - Sentence splitter: división del texto en sentencias (muchas veces es parte del tokenizer)
  - Tokenizer: división en tokens (palabras o símbolos individuales)
  - Stopword removal: eliminación de palabras conectoras.
  - Stemmer/Lemmatiser: Obtención de la raíz de una palabra, de manera que las búsquedas se puedan realizar sobre la raíz.
  - POS tagger, NER, Semantic tagger,

Ejemplo, preprocesamiento simple integrado en Solr

- El sistema de indexación del ejemplo integra algunas herramientas simples de análisis de texto en español.

```
<fieldType name="text_es" class="solr.TextField" positionIncrementGap="100">
    <analyzer>
      <tokenizer class="solr.StandardTokenizerFactory"/>
      <filter class="solr.LowerCaseFilterFactory"/>
      <filter class="solr.StopFilterFactory" format="snowball"
words="lang/stopwords_es.txt" ignoreCase="true"/>
      <filter class="solr.SpanishLightStemFilterFactory"/>
    </analyzer>
  </fieldType>
```

Indexación

- Múltiples alternativas:
  - Bases de datos adaptadas a búsquedas: MongoDB, MySQL, etc. Generar diferentes tipos de índices.
  - Herramientas especializadas:
    - ElasticSearch
    - Solr
    - Sphinx
  - Nos vamos a basar en Solr, que a su vez, está basado en Lucene como motor de almacenamiento y búsqueda.

- Descargar de solr.apache.org
- Ejecutar: ./bin/solr.cmd start
- Crear un “core” (un índice): ./bin/solr.cmd create nutch_solr
- Entrar a localhost:8983/solr .

Nutch -> Solr

- Se pueden envíar directamente los datos del almacén de Nutch a Solr.
- Comando:

```
bin/nutch solrindex http://127.0.0.1:8983/solr/nutch_solr crawl/crawldb -linkdb crawl/linkdb crawl/segments/*
```

Búsquedas en Solr

- Básicas:
  - field1:string AND field2:otrostring OR field3:untercerstring.
  - El menos (-) indica que no debe aparecer la secuencia.
  - Se pueden definir rangos para fechas, enteros y strings. Las llaves son exclusivas, los corchetes inclusivos. {1 TO 10]

Otros parámetros

- Highlighting: retorna un fragmento de texto con una parte remarcada con la coincidencia de la búsqueda.
- Faceting: Permite retornar las diferentes tópicos/caterogorías de los documentos del resultado, separados por la cantidad de apariciones.
- Spellcheck: basado en lo que se indexó se arman sugerencia ante un posible error de tipeo/ortografía.
- More like this: Genera una nueva query en base a lo encontrado
- Collapse/Expand: se indica un campo por el cual colapsar resultados.