# Laborator nr. 1
# Parser HTML. Extragere de cuvinte

an univ. 2018 – 2019

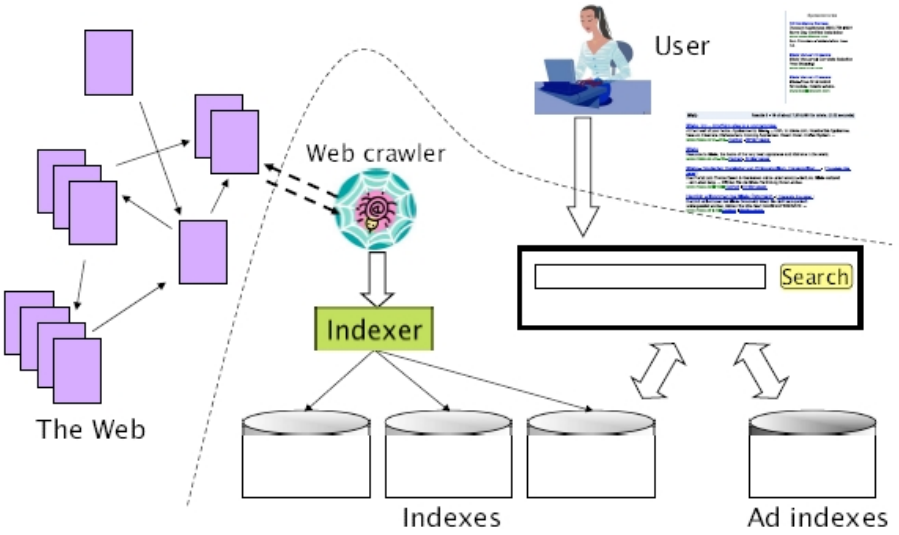


Figura 1: Structura generală a motoarelor de căutare pe WEB (preluare din [1])

## 1 Noţiuni teoretice

Caracteristici HTML:

- HTML (**H**yper**T**ext **M**arkup **L**anguage) -– reprezintă un mijloc prin intermediul căruia se pot crea documente structurate;
- Autor: **Tim Berners-Lee**
- Tip limbaj: **limbaj de tip markup** -– marcarea unui document pentru a stabili structura logică a acestuia
- Tip media: **text/html**
- Tip de aplicaţie ţintă: **interpretor**
- Extensie fişiere: htm, html
- Specificaţii: set de elemente (etichete, tag-uri), precum şi regulile de scriere ale acestora, destinate formatării documentelor.
- Sintaxa generală:
  `<nume-element [lista atribute]>continut</nume-element>`
  unde lista de atribute se defineşte ca:
  `nume-atribut="valoare-atribut" [nume-atribut="valoare-atribut"] ...`

# 2 Aplicaţie propusă

1. Implementaţi un modul de aplicaţie care să parseze un document html şi să extragă următoarele date:

   1. conţinutul elementului `<TITLE>`, dacă acesta există;
   2. conţinutul atributului `content`, tag-ul `<META>`, pentru cazul în care atributul `name` al aceluiaşi element are valorea `"keywords"`;
   3. conţinutul atributului `content`, tag-ul `<META>`, pentru cazul în care atributul `name` al aceluiaşi element are valorea `"description"`;
   4. conţinutul atributului `content`, tag-ul `<META>`, pentru cazul în care atributul `name` al aceluiaşi element are valorea `"robots"`;
   5. conţinutul elementului `<A>`, atributul `href`, dacă acesta există în cadrul documentului şi dacă referinţele indicate NU reprezintă legături interne; link-urile trebuie reconstruite sub formă de URL-uri absolute dacă este cazul;
   6. conţinutul text al documentului HTML.

   **Indicaţie**: pentru simplitate, se recomandă utilizarea librăriei JSOUP [2].

2. Realizaţi un modul de aplicaţie care primeşte ca intrare un text şi împarte acest text în cuvinte. Pentru fiecare cuvânt în parte, se va contoriza şi numărul de apariţii ale acelui cuvânt în textul de intrare. Rezultatele vor fi stocate fie într-un HashTable [4], fie într-un HashMap [3] (`key` va fi un cuvânt extras din text, `value` va stoca, sub forma unui întreg, numărul de apariţii).

   Se va considera cuvânt orice succesiune de caractere alfa-numerice cuprinsă între separatorii clasici precum: `' '` (spaţiu, succesiune de spaţii, tab), `'"'`, `','`, `'.'`, `'!'`, `'?'`, etc.. În mod ideal, implementările oferite NU ar trebui să conţină bucle îmbricate.

**Observaţie 1**:
Implementările care se bazează pe metoda `split` [5] a clasei `String` vor fi punctate cu nota maximă **8** (**opt**).

# Bibliografie

[1] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze. *Introduction to Information Retrieval*. Cambrige University Press, 2009.

[2] Jonathan Hedley. jsoup: Java HTML Parser. `http://jsoup.org`.

[3] Oracle. Class HashMap<K,V>. `http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html`.

[4] Oracle. Class Hashtable<K,V>. `http://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html`.

[5] Oracle. Class String. `http://docs.oracle.com/javase/7/docs/api/java/lang/String.html`.