

Grupo 314

Integrantes:

- Brandy, Tobias
- Colonnello, Joaquin
- Vazquez, Ignacio



PERSISTENCIA POLÍGLOTA

Elasticsearch + Neo4J

Objetivos

Importación

Importar a Neo y Elastic a partir de un artículo de wikipedia, los artículos a los que linkee, filtrando por categorías y distancia máxima

Búsqueda

Desarrollar un pequeño motor de búsquedas que permita buscar articulos aprovechando las fortalezas de Neo y Elastic

Modelado Bases

Esquema Elasticsearch

```
"mappings" : {  
  "properties" : {  
    "article_id" : {  
      "type" : "keyword"  
    },  
    "categories" : {  
      "type" : "keyword"  
    },  
    "content" : {  
      "type" : "text",  
      "analyzer" : "folding_analyzer"  
    },  
    "title" : {  
      "type" : "text"  
    }  
  }  
}
```

Esquema Neo4j

```
{  
  "labels": ["Article"],  
  "relationshipsType": ["Link"],  
  "article_id": 23862,  
  "categories": [  
    "Class-based programming languages",  
    "Computational notebook",  
  ],  
  "title": "Python (programming language)"  
}
```

Unique Constraints: article_id y title

Importación

Se realiza enviando un payload a **/api/import** de las siguientes características:

- **Artículo inicial:** Título del artículo centro por el que se empieza a importar
- **Radio:** Distancia máxima de artículos a importar desde el centro
- **Categorías:** Lista de categorias de artículos a importar (any of)
- **Lenguaje:** El lenguaje de wikipedia a utilizar (default 'en')

No importamos nodos de id 0, o links autorreferenciales

Importación

```
{  
  "center_page": "Python (programming language)",  
  "radius": 3,  
  "categories": [  
    "Object-oriented programming languages",  
    "Class-based programming languages"  
  ],  
  "lang": "en"  
}
```

Búsquedas

Se realizan enviando un payload a `/api/search` de las siguientes características:

- **Estructura general**
 - **Tipo de retorno:** Total count, lista de ids, lista de títulos, nodos sin contenido, nodos con contenido
 - **Filtros Elasticsearch:** Lista de filtros para elastic
 - **Filtros Neo4j:** Lista de filtros para Neo4j
 - **Filtros generales:** Lista de filtros que no pertenecen a ninguna base en particular
 - **Sort:** Configuración de ordenamiento
 - **Limit y Offset:** Configuración de paginación

Búsquedas

Se realizan enviando un payload a `/api/search` de las siguientes características:

```
{
  "return_type": "NODE",
  "elastic_filter": [...],
  "neo_filter": [...],
  "general_filters": [...],
  "sort": {
    "sort_by": "TITLE",
    "type": "ASC"
  },
  "limit": 10,
  "offset": 30
}
```


Tipos de Retorno

- **Cantidad total de nodos (COUNT):** Entero que representa el total de nodos que verifican
- **Lista de ids (ID):** Lista de los ids de los nodos que verifican
- **Lista de títulos (TITLE):** Lista de los títulos de los nodos verifican
- **Nodos (NODE):** Lista de nodos sin contenido. Valor default.
- **Nodos con contenido (NODE_WITH_CONTENT):** Lista de nodos con contenido

Modelado de Nodo

- Id
- Título
- Categorías
- Links (relaciones)
- Contenido (*opcional*)

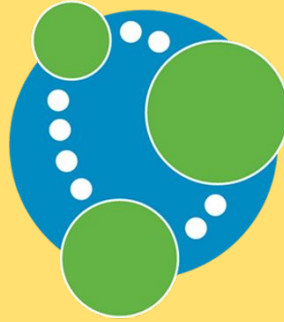
```
{
  "id": 215824,
  "title": "Gambas",
  "categories": ["BASIC compilers", "Computer program"],
  "links": [
    {
      "article_id": 23939,
      "title": "Perl"
    },
    {
      "article_id": 15881,
      "title": "Java (programming language)"
    }
  ],
  "content": null
},
```

Tipos de filtros



ElasticSearch

Filtros de texto



Neo4J

Filtros sobre las
relaciones entre artículos



Generales

Filtros de artículos que no
son específicos a alguna
de las DB

Filtros de Elasticsearch

- **Búsqueda de texto**

- **Campo:** Título o contenido (TITLE, CONTENT)
- **Fuzzy:** Realizar una búsqueda fuzzy del patrón ingresado
- **Operación:** Si debe contener alguno o todos los patrones (AND, OR)
- **Patrones:** Lista de cadenas a buscar

```
{  
  "field": "TITLE",  
  "fuzzy": true,  
  "bool_op": "AND",  
  "matches": [  
    "programming",  
    "development"  
  ]  
}
```

Fuzzy solo es soportado con palabras, no frases

Filtros de Neo4J



- **Filtro de distancia**

Selecciona los artículos en base a la longitud del camino más corto desde/hasta un artículo de origen

- **Filtro de links**

Selecciona los artículos en base a la cantidad de links entrantes/salientes en cada uno

Filtros de Neo4J

- **Filtro de distancia**

- **Nodo de origen:** Título del artículo central
- **Distancia:** Cantidad de saltos hasta el nodo de origen
- **Estrategia:** Filtrar artículos hasta la distancia especificada o exactamente a ese valor (UP_TO_DIST, AT_DIST)
- **Dirección:** Buscar caminos desde el nodo de origen o hacia él

```
{  
  "source_node": "Python",  
  "dist": 1,  
  "strategy": "UP_TO_DIST",  
  "direction": "OUTGOING"  
}
```

Filtros de Neo4J

- **Filtro de links**

- **Cantidad mínima de links**
- **Cantidad máxima de links**
- **Categorías:** Solo cuenta links a/desde artículos que pertenezcan a ciertas categorías
- **Dirección:** Cuenta links entrantes o salientes (INGOING, OUTGOING)

```
{
  "min_count": 20,
  "max_count": 60,
  "categories": [
    "Scripting languages",
    "cross platform"
  ],
  "direction": "INGOING"
}
```

Filtros generales

- Ids

```
{  
  "ids": [25768, 23862]  
}
```

- Títulos

```
{  
  "titles": ["Python", "Ruby"]  
}
```

- Categorías

```
{  
  "categories": ["Scripting", "Cross-platform"]  
}
```

Los artículos tendrán **alguno** de los especificados

Ordenamiento

- **Ordenamiento por:**
 - **Título** (TITLE)
 - **Id** (ID)
 - **Grado del nodo** (LINK_COUNT)
- **Tipo:**
 - **Ascendente** (ASC)
 - **Descendente** (DESC)

```
"sort": {  
  "sort_by": "TITLE",  
  "type": "ASC"  
}
```

Paginación

- **Limit**
- **Offset**

```
"limit": 10,  
"offset": 30
```

Ejemplos de Respuesta

COUNT

```
{
  "result": {
    "count": 5
  }
}
```

ID

```
{
  "result": [
    3417,
    777,
    51260,
    25484,
    18952765
  ]
}
```

TITLE

```
{
  "result": [
    "Biennial plant",
    "Annual plant",
    "Bulb",
    "Rosemary",
    "Succulent plant"
  ]
}
```

NODE

```
{
  "result": [
    {
      "id": 3417,
      "title": "Biennial plant",
      "categories": [
        "Biennial plants",
        "Garden plants"
      ],
      "links": [
        {
          "article_id": 212563,
          "title": "Lunaria"
        },
        {
          "article_id": 777,
          "title": "Annual plant"
        }
      ],
      "content": null
    }
  ]
}
```


Ejemplos

"Quiero todos los **títulos** de los artículos de la categoría '**Plantas**' que estén a **menos de 5 saltos** de la página de wikipedia de '**Jazmín**' y que se sean de la **misma familia biológica**"

Resultado: ["Jasmine", "Jasminum nudiflorum"]

```
{
  "return_type": "TITLE",
  "neo_filter": [
    {
      "source_node": "Jasmine",
      "dist": 5,
      "strategy": "UP_TO_DIST",
      "direction": "OUTGOING"
    }
  ],
  "general_filters": [
    { "categories": ["Jasminum"]}
  ],
  "sort": {
    "sort_by": "TITLE",
    "type": "ASC"
  }
}
```

Ejemplos

"Quiero los primeros 5 **títulos** de los artículos de la categoría '**Perfumes**' que estén a **menos de 5 saltos** de la página de wikipedia de '**Jazmín**' y que contengan la palabra '**Aroma**' ordenado por su **cantidad de apariciones**"

Resultado: ["Kolonya", "Perfume"]

```
{
  "return_type": "TITLE",
  "elastic_filter": [
    {
      "field": "CONTENT",
      "fuzzy": false,
      "bool_op": "AND",
      "matches": [
        "aroma"
      ]
    }
  ],
  "neo_filter": [
    {
      "source_node": "Jasmine",
      "dist": 5,
      "strategy": "UP_TO_DIST",
      "direction": "OUTGOING"
    }
  ],
  "general_filters": [
    { "categories": ["Perfumes"]}
  ],
  "sort": {
    "sort_by": "TITLE",
    "type": "ASC"
  },
  "limit": 5,
  "offset": 0
}
```

The background is a solid light pink color. It is decorated with various abstract elements: a large yellow shape in the top left, a dark blue shape in the top right, a yellow circle in the middle right, a blue shape in the bottom center, and a yellow shape in the bottom right. There are also white wavy lines, a white dashed pattern, and a blue wavy pattern. The word "DEMO" is centered in a bold, dark blue font.

DEMO