Backend – Sistema de Recuperación de Información

Este directorio contiene el **backend del Sistema de Recuperación de Información (RI)** desarrollado como parte de la práctica final de la asignatura.

El backend está implementado en **Python** utilizando **FastAPI**, y expone una serie de endpoints que representan las distintas fases del proceso de Recuperación de Información: análisis léxico, tokenización, eliminación de stopwords, lematización/stemming, ponderación de términos, indexación y búsqueda.

Tecnologías utilizadas

- Python 3.13
- FastAPI
- Uvicorn
- NLTK
- Pydantic
- Entorno virtual (`venv`)

El desarrollo y las pruebas se han realizado sobre **Arch Linux**, aunque el backend es portable a cualquier sistema compatible con Python 3.

Estructura del backend

```
backend/
├── app/
│   ├── main.py             # Punto de entrada de FastAPI
│   ├── api/
│   │   └── routes.py       # Definición de endpoints
│   ├── core/
│   │   └── config.py       # Configuración global
│   ├── schemas/
│   │   ├── requests.py     # Modelos de entrada (Pydantic)
│   │   └── responses.py    # Modelos de salida (Pydantic)
│   ├── services/
│   │   ├── preprocess.py   # Pipeline de preprocesado
│   │   ├── indexer.py      # Construcción del índice
│   │   └── searcher.py     # Motor de búsqueda
│   └── storage/
│       ├── jsonl.py        # Lectura de corpus en formato JSONL
│       └── paths.py        # Gestión de rutas de datos
├── scripts/
│   ├── install.sh          # Instalación del entorno y dependencias
│   ├── dev.sh              # Arranque del servidor en modo desarrollo
│   └── download_nltk.py    # Descarga de recursos NLTK
├── requirements.txt
└── README.md
```

Requisitos del sistema

- Python **3.10 o superior** (recomendado 3.13)
- `python-venv`
- `pip`
- Bash (para ejecutar los scripts)

Instalación

Desde el directorio `backend/`, ejecutar:

```
chmod +x scripts/*.sh
./scripts/install.sh
```

Este script realiza las siguientes acciones:

1. Crea un entorno virtual en `.venv`
2. Instala las dependencias del proyecto
3. Descarga los recursos necesarios de NLTK

Ejecución del backend

Para arrancar el servidor en modo desarrollo:

```
./scripts/dev.sh
```

El backend quedará accesible en:

```
http://localhost:8000
```

La documentación automática de la API (Swagger UI) está disponible en:

```
http://localhost:8000/docs
```

Endpoints disponibles

El backend expone los siguientes endpoints principales:

Preprocesado de texto

- `POST /lexical_analysis`
- `POST /tokenize`
- `POST /remove_stopwords`
- `POST /lemmatize`
- `POST /weight_terms`
- `POST /select_terms`

Indexación

- `POST /index`
  Construye el índice a partir de un corpus en formato JSONL.

Búsqueda

- `GET /search?query=texto`
  Devuelve los documentos más relevantes para una consulta.

Formato del corpus

El endpoint `/index` espera un corpus en formato **JSONL**, donde cada línea representa un documento independiente.

Ejemplo de documento:

```
{
  "doc_id": "1",
  "title": "Ejemplo de documento",
  "text": "Contenido del documento...",
  "url": "https://ejemplo.com"
}
```

Notas importantes

- El backend debe ejecutarse siempre dentro del entorno virtual `.venv`.
- El endpoint `/search` requiere que el índice haya sido construido previamente mediante el endpoint `/index`.
- El sistema implementa un motor de búsqueda **baseline basado en TF-IDF**, suficiente para cumplir los objetivos de la práctica.
